ForensiQ: A Knowledge Graph Question Answering System for IoT Forensics

Ruipeng Zhang and Mengjun Xie(✉)

University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA
ruipeng-zhang@mocs.utc.edu, mengjun-xie@utc.edu

Abstract. The increasing number of attacks against the Internet of Things (IoT) has made IoT forensics critically important for reporting and mitigating cyber incidents and crimes. However, the heterogeneity of IoT environments and the complexity and volume of IoT data present significant challenges to forensic practitioners. The advent of question answering (QA) systems and large language models (LLM) offers a potential solution to accessing sophisticated IoT forensic knowledge and data. In light of this, we propose ForensiQ, a framework based on knowledge graph question answering (KGQA), to help investigators navigate complex IoT forensic artifacts and cybersecurity knowledge. Our framework integrates knowledge graphs (KG) into the IoT forensic workflow to better organize and analyze forensic artifacts. We also have developed a novel KGQA model that serves as a natural-language user interface to the IoT forensic KG. Our evaluation results show that, compared to existing KGQA models, ForensiQ demonstrates higher accuracy in answering natural language questions when applied to our experimental IoT forensic KG.

Keywords: Internet of Things · Digital Forensics · Knowledge Graph · Ontology Design · Question Answering

1 Introduction

The rapid adoption of Internet of Things (IoT) not only has resulted in exciting transformation in many sectors, e.g., industry 4.0, smart cities, and smart health, but also has introduced significant challenges in terms of IoT cybersecurity. Recent years have witnessed a quickly growing interest in IoT forensics, as the involvement of IoT in cyber criminal activities becomes increasingly popular. However, the heterogeneous nature of IoT devices and the enormous volume of data they generate make it nearly impossible for IoT forensic investigators, especially those in their early career phases, to possess extensive expertise and up-to-date knowledge in the forensic techniques required for IoT forensics. A recent survey [33] emphasizes the pressing challenges faced by cybersecurity

This work was supported in part by the National Science Foundation (award no. 1663105) and National Security Agency (award no. H98230-20-1-0408).

S. Goel and P. R. Nunes de Souza (Eds.): ICDF2C 2023, LNICST 571, pp. 300–314, 2024.
https://doi.org/10.1007/978-3-031-56583-0_20

practitioners, including the need for technical training, software, and education in IoT digital forensics. There is a strong and critical demand for a more effective framework to support IoT forensic practitioners especially those inexperienced investigators in investigating and promptly responding to IoT-related crimes and security incidents.

Knowledge graph question answering (KGQA) offers a new perspective and approach to facilitate IoT forensic investigation process, assist forensic investigators, and lower the overheads associated with IoT forensics (e.g., learning and searching). Knowledge graphs (KGs) provide a structured representation of real-world objects and their relationships, forming the graphs (often sparse) that can be processed and analyzed using graph algorithms. KGQA systems can leverage Natural Language Processing (NLP) to interpret user intents and reason over KGs. Moreover, KGQA models are capable of answering complex, domain-specific questions. Recent studies show that they can outperform large language models (LLMs) such as GPT-3 and ChatGPT [23,31].

In this paper, we propose ForensiQ, an IoT forensics framework based on KGQA, to address those challenges faced by forensic investigators in IoT forensic investigations. ForensiQ is aimed to simplify and facilitate the access to and analysis of forensic artifacts and cybersecurity knowledge. In ForensiQ, complex forensic artifacts collected from crime scenes are first transformed into structured KGs and then enriched with cybersecurity knowledge. To answer case specific natural language questions, ForensiQ employs a combination of the LLM and graph neural network (GNN) based KGQA model, utilizing the KG as the data source. Leveraging ForensiQ, a variety of overheads such as learning and searching associated with IoT forensics can be significantly reduced, and the analysis of forensic artifacts can be expedited.

Our main contributions are summarized as follows:

1. We have designed a knowledge graph ontology specifically for IoT forensic data analysis. This ontology serves as a standardized vocabulary for organizing case related data and guiding the construction of the IoT forensic knowledge graph.
2. We have curated a comprehensive dataset for IoT forensic KGQA by collecting data from multiple sources. This dataset serves as a valuable resource for evaluating IoT forensic KGQA systems and enables future research in the field of IoT KGQA.
3. We have developed a novel KGQA model for IoT forensics that combines a large language model and a graph neural network. Our experiments demonstrate the effectiveness of the new model in accurately answering natural language questions about the IoT forensic knowledge graph.

The rest of this paper is organized as follows: We provide a brief background and related work on IoT forensics and KGQA in Sect. 2. We then detail our proposed framework in Sect. 3 and present the experimental setup and results in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Background and Related Work

Digital forensics in IoT can be broadly divided into three categories based on the scope of operation: device forensics, network forensics, and cloud forensics [15]. In device forensics, the investigator acquires the target IoT device from the crime scene and collects evidence directly from the device. This approach focuses on data extracted from the device, such as multimedia files (image, audio, video), local databases, and log files. For example, Alabdulsalam *et al.* examined the Apple Watch Series 2 and manually extracted messages, pictures, and emails [1], while Li *et al.* conducted a digital forensic operation simulation on an Amazon Echo using the Alexa Pi and dumped the device's firmware to an image file [21].

As IoT devices are usually connected to a network, network forensics is crucial for identifying cyber attacks in an IoT environment and collecting evidence for subsequent analysis and incident response. This aspect of IoT forensics often employs networking tools such as packet sniffers and analyzers to monitor abnormal activities at the network level. For instance, Rizal *et al.* proposed a network forensics model to detect flooding attacks on IoT devices, using WireShark to capture and examine network traffic [25]. Koroniotis *et al.* developed the Particle Deep Framework (PDF) for IoT network forensics, which integrates a deep neural network based on particle swarm optimization algorithms to detect and trace abnormal events in IoT networks [18].

Many IoT devices transmit personal data to cloud providers for functionality and analysis. The involvement of cloud-based processing and storage introduces new challenges to IoT forensics. Cloud forensics is a new approach to tackling digital forensics for cloud-enabled IoT environments. One major issue with cloud forensics is trust, as cloud providers may collude with malicious parties to conceal illegal activities. To address this issue, the Open Cloud Forensics (OCF) model has been proposed to help cloud architects build a forensics-aware cloud infrastructure that supports trustworthy cloud forensic investigations [35]. Another primary challenge for cloud forensics is legal regulation. Data territoriality, cloud content ownership, user authentication, and data preservation are among the main issues that must be considered in cloud forensic investigation [16].

The popularity of deep learning powered natural language processing has sparked extensive research in the field of natural language QA. KGQA, in particular, utilizes structured multi-relational data from a knowledge graph to deliver accurate and reliable results for QA tasks. A KG or knowledge base (KB) is a structured graph representation of facts. It is formally defined as

$$\mathcal{G} = \{(s, p, o) | s, o \in \mathcal{E}, p \in \mathcal{R}\}, \tag{1}$$

where each (s, p, o) is a triple or fact, with s as the subject entity, p as the predicate, and o as the object entity. The entity set is denoted as $\mathcal{E}$ and the relation set as $\mathcal{R}$. In KGQA, the objective is to predict a set of answers $\mathcal{A}_q$ for a given natural language question q, based on a knowledge graph $\mathcal{G}$. For simple questions, the answers can be directly retrieved from the entity set $\mathcal{E}$ ($\mathcal{A}_q \subseteq \mathcal{E}$).

However, for more complex questions, especially those involving numerical or aggregation operations, the answer may need to be derived from the information contained within the knowledge graph.

A KG ontology serves as a framework for understanding domain knowledge and acts as a blueprint for KGs. It promotes the sharing and reuse of domain knowledge by importing concepts from other ontologies or extending its coverage to subdomains. Standard ontologies have been established for IoT, including the Semantic Sensor Network (SSN) Ontology [11], the Smart Applications REFerence (SAREF) ontology [6], and the oneM2M base ontology [24]. In the field of digital forensics, Dosis *et al.* proposed an innovative approach to representing and integrating digital evidence from various sources using ontologies [8]. Ellison *et al.* developed an ontology for reactive digital forensics techniques, organizing them based on their purposes and providing a formalized framework [10]. Regarding digital forensics analysis, Sikos *et al.* outlined four types of knowledge that can be utilized: technical knowledge, investigation process knowledge, cybersecurity knowledge, and case-specific knowledge [30].

Research on integration of IoT and KGQA has increased significantly due to KGQA's capability of providing an intuitive interface for complex domain knowledge. Li *et al.* developed a KGQA system specifically for smart care of elderly individuals with chronic diseases, improving access to relevant knowledge for primary care staff [19]. In the smart grid sector, Yun *et al.* designed a fault operation and maintenance KG, along with a QA system for diagnosing faults in electric information collection systems [34]. Tan *et al.* utilized KG to model concepts in the electric power customer service business and introduced a QA application architecture to enhance customer service intelligence [32]. Chen *et al.* demonstrated AgriKG, a KG-based agricultural information system that extracts agricultural knowledge from unstructured text and enables QA through subgraph matching [4].

3 Proposed Approach

A high level overview of the ForensiQ framework is shown in Fig. 1. This framework integrates KG ontology design, KG construction, and KGQA into the six phases of the IoT forensic investigation process, which are evidence identification, device acquisition, data extraction, data analysis, evidence examination, and reporting [2]. The investigation begins by identifying forensic artifacts crucial for establishing the crime. Once the relevant evidence is identified, the investigator can utilize ontologies to define the core concepts and attributes related to the specific artifacts under examination. Forensic artifacts, such as log files and network traces, are transformed into structured KGs. This transformation can be achieved through either rule-based or heuristic parsers. The resulting KGs are then intricately linked with common cybersecurity KGs, forming a comprehensive forensic KG through the KG fusion process. Finally, to answer the investigator's natural language questions regarding the case, a KGQA model is trained using the forensic KG and a set of generated questions.

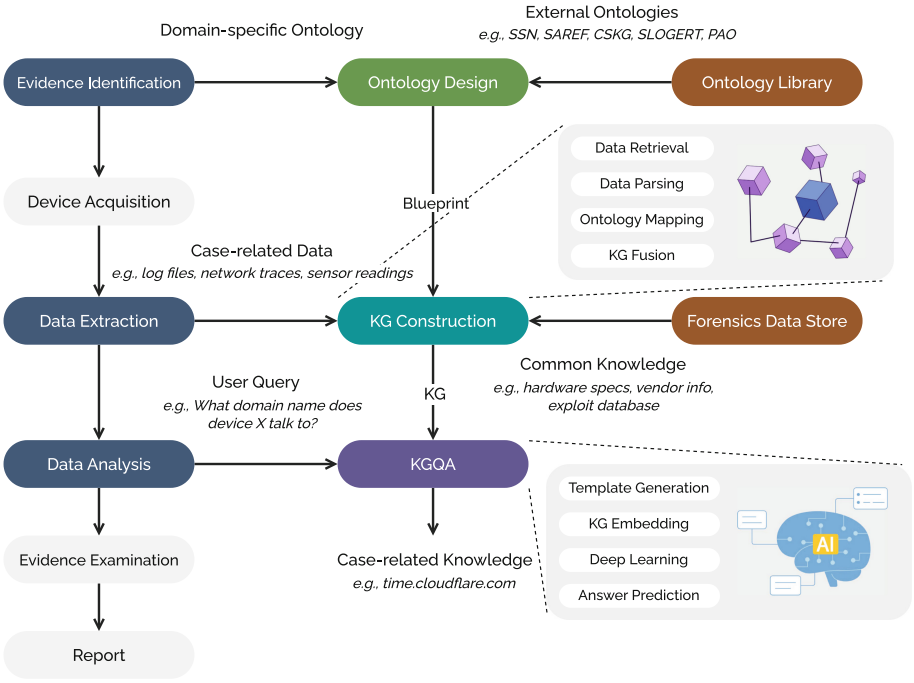

Fig. 1. The ForensiQ IoT forensics framework

3.1 Ontology Design

The proposed IoT forensics ontology is visually presented in Fig. 2. Building upon the SAREF ontology, which defines the properties, functions, and measurements of IoT devices, our ontology extends it further. It incorporates additional concepts to describe ownership, locality, and connectivity of IoT devices. To capture a comprehensive representation of IoT devices, we introduce concepts such as organization, hardware and software information, location, and networking capabilities. These concepts establish connections with well-defined external ontologies that model crucial artifacts generated within IoT environments. Notably, we incorporate the Semantic LOG ExtRaction Templating (SLOGERT) [9] for representing log events and templates, and the Packet Analysis Ontology (PAO) [29] for handling network captures. Furthermore, the inclusion of the SEPSES Cybersecurity KG [17] enriches the ontology by providing valuable resources for identifying vulnerabilities and weaknesses in IoT hardware and software. This assists investigators in identifying potentially compromised IoT devices.

3.2 KG Construction

To create an IoT forensic KG for investigation, we begin by collecting case-related artifacts from the crime scene using forensic data extraction tools. Meanwhile,

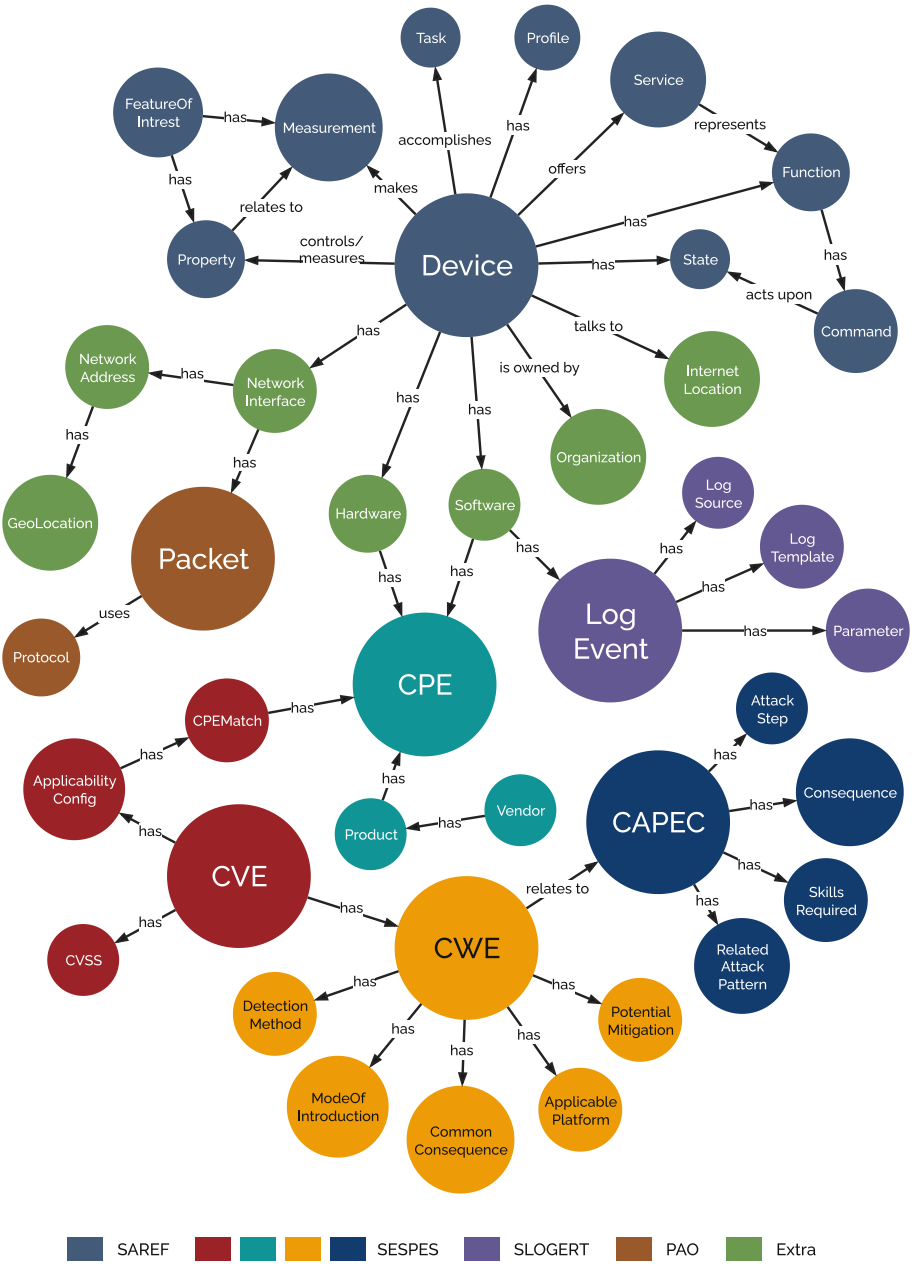

Fig. 2. High-level overview of the proposed ontology

we gather auxiliary cybersecurity knowledge from official sources, including vulnerability databases and vendor information, which can be valuable for the case. Then, we employ data parsing tools to process the unstructured data and build

the KG. For example, an online log parser called Drain [12] is used to convert plain-text IoT log files into structured log events. When dealing with binary-formatted network captures generated by IoT devices, we rely on Scapy [26], which extracts various details such as protocol, host, and other packet information from the network captures. This extraction process enables us to retrieve crucial data from the network traffic generated by IoT devices.

The next step in constructing the KG involves converting data from various sources into individual KGs using ontology mapping. During this process, we extract triples from structured data generated in the data extraction process. Afterwards, the entities from the individual KGs are merged together in the KG fusion step to form a comprehensive KG. For instance, log events are linked to IoT devices based on the software responsible for generating the logs, and network packets are associated with the device's network interfaces that transmit or receive them. Moreover, IoT devices are connected to relevant cybersecurity knowledge in the SEPSES cybersecurity KG through the hardware and software of these devices. This integration allows for a comprehensive understanding of the relationships between IoT devices and cybersecurity standards.

3.3 KGQA Model

The architecture of ForensiQ's KGQA model is illustrated in Fig. 3. Initially, the model uses a large language mode (LLM) to infer topic entities $\mathcal{E}_q \subseteq \mathcal{E}$ and relations $\mathcal{R}_q \subseteq \mathcal{R}$ from the input natural language question q. This process, known as entity and predicate linking, helps identify relevant entities and relations within the question. Subsequently, the subgraph extractor retrieves a subgraph $\mathcal{G}_q$ from the IoT forensic knowledge graph $\mathcal{G}$, which contains the predicted topic entities and relations. To determine the answer to question q, the answer predictor calculates the probability of each entity in the subgraph being the answer $\mathcal{A}_q$. The entity with the highest probability is then selected and returned as the answer.

Entity and Relation Predictors. A question about the IoT forensic KG typically contains one or more topic entities $\mathcal{E}_q \subseteq \mathcal{E}$ and relations $\mathcal{R}_q \subseteq \mathcal{R}$. To identify the most relevant entities and relations for a given question q, the entity predictor and relation predictor calculate semantic similarity between q and every entity $e \in \mathcal{E}$. Pretrained LLMs like BERT [7] are trained on extensive text data and are effective for many natural language tasks. Hence, we employ BERT-based models in predictors to measure the proximity of entities and relations to the question. Denoting the entity predictor as $f_e(\cdot)$, the relation predictor as $f_r(\cdot)$, and the LLM as $f_{\text{LLM}}(\cdot)$, the semantic similarity between q and an entity e or relation r can be computed as follows:

$$f_e(q, e) = f_{\text{LLM}}([q; \text{SEP}; e]), \tag{2}$$

$$f_r(q, r) = f_{\text{LLM}}([q; \text{SEP}; r]). \tag{3}$$

Here, $[a; b]$ represents string concatenation of a and b, and SEP denotes BERT's sentence separation token. The entities and relations with the highest similarities are selected as $\mathcal{E}_q$ and $\mathcal{R}_q$.

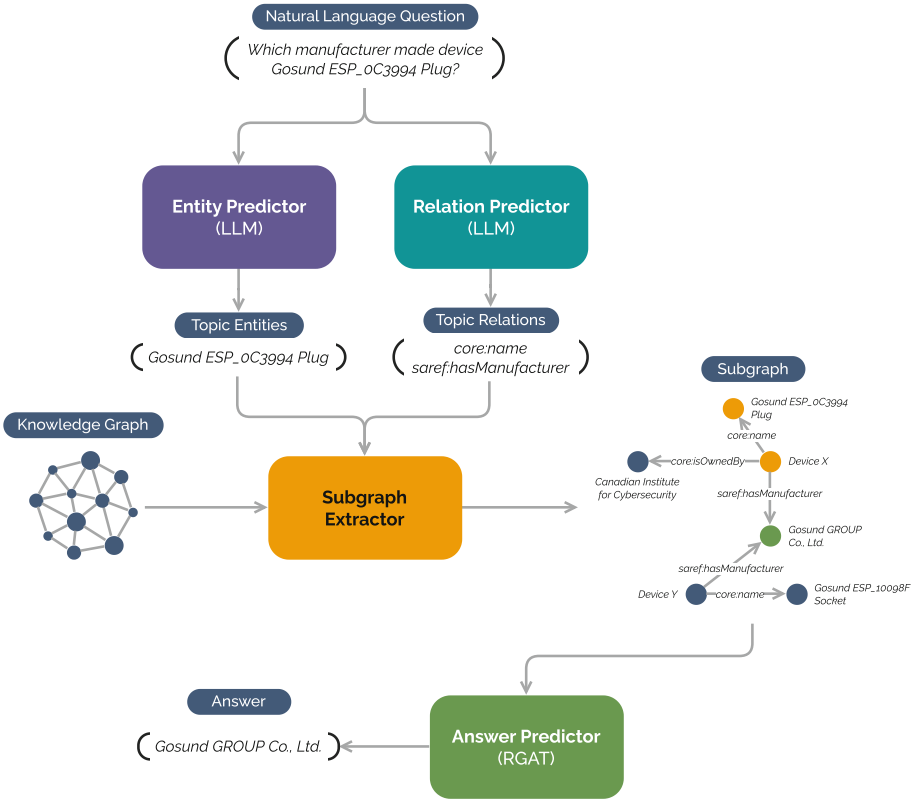

Fig. 3. Overview of the KGQA model architecture

The entity and relation predictors are trained using margin ranking loss to effectively distinguish true topic entities (positive samples) from others (negative samples). Negative sampling is employed during training, which randomly selects only a small number of negative samples for each question. This approach enhances training efficiency and creates a more balanced dataset, considering that the number of positive samples in the original KG is several magnitudes smaller than that of negative samples. Given the similarity f_e of a positive entity and $f_{e'}$ of a negative entity, the loss function for the entity predictor is defined as follows:

$$\mathcal{L}_e = \sum_{e,e' \in \mathcal{E}} \max(0, m - (f_{e'} - f_e)), \tag{4}$$

where m represents the desired maximum distance between f_e and $f_{e'}$. Similarly, the loss function for the relation predictor is defined as:

$$\mathcal{L}_r = \sum_{r,r' \in \mathcal{R}} \max(0, m - (f_{r'} - f_r)). \tag{5}$$

Subgraph Extractor. Utilizing the entity and relation predictors' predictions, the subgraph extractor retrieves the subgraph $\hat{\mathcal{G}}$ from the original KG, representing the question and its answer. This extraction process significantly narrows down the search space by excluding irrelevant KG entities. To construct $\hat{\mathcal{G}}$ containing $\mathcal{E}_q$ and $\mathcal{R}_q$, the extractor first retrieves the k-hop closed neighborhood of all nodes in $\mathcal{E}_q$. Next, edges not present in $\mathcal{R}_q$ are removed, along with any orphan nodes. In addition, inverse edges are incorporated into $\hat{\mathcal{G}}$ to account for reverse relations in q. To enhance the robustness of the answer predictor, a fixed number of random edges are introduced into $\hat{\mathcal{G}}$ as noise.

Answer Predictor. The answer predictor employs a variant of GNN called Relational Graph Attention Network (RGAT) [3]. The predictor comprises a learnable entity embedding module $f_{\text{EMB}}(\cdot)$, L RGAT convolution layers $f_{\text{RGAT}}(\cdot)$, and a linear output layer $f_{\text{OUT}}(\cdot)$. It takes all entities $\mathcal{E}_{\hat{\mathcal{G}}}$ and the edge features of all edges in $\hat{\mathcal{G}}$ as input and generates an array of scores $\boldsymbol{S}_q = \{s_1, s_2, \ldots, s_N\}$, where $N = |\mathcal{E}_{\hat{\mathcal{G}}}|$. Each score represents the probability of an entity in $\hat{\mathcal{G}}$ being the answer entity.

Denote the edge features from the i^{th} entity to the j^{th} entity as $\boldsymbol{e}_{i,j}^{(r)}$, and the hidden state of all entities in $\mathcal{E}_{\hat{\mathcal{G}}}$ at the l^{th} RGAT layer as $\boldsymbol{H}^{(l)}$. The answer predictor can be defined as follows:

$$\boldsymbol{E} = \{\boldsymbol{e}_{i,j}^{(r)}\} \in \mathbb{R}^{C \times B}, r \in \mathcal{R}, (i, r, j) \in \hat{\mathcal{G}}, \tag{6}$$

$$\boldsymbol{H}^{(0)} = f_{\text{EMB}}(\mathcal{E}_{\hat{\mathcal{G}}}) \in \mathbb{R}^{N \times F}, \tag{7}$$

$$\boldsymbol{H}^{(l)} = f_{\text{RGAT}}(\boldsymbol{H}^{(l-1)}, \boldsymbol{E}) \in \mathbb{R}^{N \times F'}, 1 \leq l \leq L, \tag{8}$$

$$\boldsymbol{S}_q = \text{Sigmoid}(f_{\text{OUT}}(\boldsymbol{H}^{(L)})) \in \mathbb{R}^N. \tag{9}$$

Here, $C = |\hat{\mathcal{G}}|$ represents the number of edges in $\hat{\mathcal{G}}$, B is the dimension of the edge features, F is the dimension of the entity features, and F' is the dimension of the entity's hidden state. During inference, the scores $\boldsymbol{S}_q$ are ranked from highest to lowest, and answers with the highest scores are returned since there may be multiple correct answers to a question. The model is trained using binary cross-entropy loss:

$$\mathcal{L}_a = \sum_{i=1}^{N} (y_i \log s_i + (1 - y_i) \log(1 - s_i)), \tag{10}$$

where y_i is either 1 or 0, indicating if the i^{th} entity is the answer entity or not.

4 Experimental Details

4.1 Dataset

To construct the IoT forensic KG for the KGQA evaluation, we curated 43 devices from the CIC IoT dataset [5] and generated four synthetic devices. These

devices were enriched with annotations including manufacturer, model, software, hardware, and geolocation information. Manufacturer and model details were obtained using FingerBank[1], a hardware fingerprinting service. The SEPSES Cybersecurity KGs were generated using data from the NIST NVD and MITRE, up until January 2023. The PAO KG was created using network captures from the CIC dataset on January 3, 2022. Additionally, we incorporated log files from Loghub [13] since the CIC IoT dataset lacks IoT device log files.

The KG obtained consists of over 6 million unique subject entities and approximately 45 million triples. Over 80% of the triples originate from the SEPSES Cybersecurity KGs, while PAO triples make up around 14.2% of the total. Triples from other sources such as SAREF and SLOGERT represent only a small portion. To reduce the dataset size and improve relevance to forensic investigation, we removed less relevant triples. Additionally, we balanced the number of triples across all the KGs by randomly sampling from SEPSES, PAO, and SLOGERT KGs. As a result, the reduced KG contains only 0.4% of the triples from the original full KG.

The QA dataset was created using a template-based approach. We manually crafted over 100 question templates and their corresponding SparQL templates based on the ontology design. By inserting randomly selected subject entities into the question templates and utilizing the SparQL templates, we generated over 10,000 synthetic questions. The dataset was then split into training, validation, and testing datasets with an 8:1:1 ratio. Regarding question complexity, the majority of the dataset (69.7%) consists of simple one-hop questions, while two-hop questions account for 23.6% of the total. The number of hops in a question indicates the minimum steps required to reach the answer entities from the subject entity in the KG, with more hops indicating greater complexity.

4.2 Baselines

Fine-tuned LLM. We fine-tuned the RoBERTa [22], a BERT-based model, on our dataset to evaluate its effectiveness in answering IoT forensic questions. A fully-connected linear layer was added as a multi-class classifier for answer entities.

KEQA [14]. KEQA utilizes pretrained KG embeddings (KGE) and employs head entity and predicate embeddings learning, head entity token detection, and joint search to locate the answer entity within the KG.

EmbedKGQA [27]. EmbedKGQA learns a question embedding that captures the relations between the head entity and the answer entity mentioned in the question. It predicts the answer by scoring and ranking entities in the KG using KGE and a scoring function.

TransferNet [28]. TransferNet employs a multi-step approach for KGQA, allowing the model to "jump" from the topic entity to the answer entity. At

[1] https://www.fingerbank.org.

each step, the model attends to edges differently by considering the input question.

SSKGQA [20]. SSKGQA predicts the semantic structure of a question and retrieves query graphs based on the predicted structure. It ranks candidate query paths and answer entities using a graph ranking model.

It is important to note that EmbedKGQA, TransferNet, and SSKGQA assume the topic or head entity has already been identified prior to inference. In contrast, the fine-tuned LLM, KEQA, and our model do not make this assumption.

4.3 KGQA Results

We evaluate the accuracy of KGQA models using Hits@k, defined as follows:

$$\text{Hits@k} = \frac{1}{|\mathcal{Q}'|} \sum_{q \in \mathcal{Q}'} \left(\mathbb{1}\left(\exists \text{rank}(a|q) \leq k, a \in \mathcal{A}_q\right) \right). \tag{11}$$

Here, $\mathcal{Q}'$ represents the test question set, $\text{rank}(a|q)$ is the rank of the correct answer a to question q among all answer predictions generated by the model, and $\mathcal{A}_q$ is the set of correct answers for question q. The function $\mathbb{1}(cond)$ equals 1 when $cond$ is true, and 0 otherwise.

Table 1. KGQA Performance of Evaluated Models on the IoT Forensic QA Dataset

Model	Hits@1				Hits@3				Hits@10			
	1-hop	2-hop	3-hop+	Overall	1-hop	2-hop	3-hop+	Overall	1-hop	2-hop	3-hop+	Overall
RoBERTa	36.02%	50.19%	14.06%	38.17%	42.56%	62.65%	25.00%	46.44%	48.40%	73.15%	48.44%	54.52%
KEQA	49.10%	0.39%	1.56%	34.13%	70.93%	0.39%	1.56%	49.23%	**96.24%**	7.78%	1.56%	68.56%
Ours	**87.76%**	**77.82%**	**59.38%**	**83.56%**	**89.57%**	**82.88%**	**60.94%**	**86.15%**	89.71%	**87.94%**	**62.50%**	**87.60%**
EmbedKGQA	88.73%	72.37%	**60.94%**	82.98%	89.85%	82.10%	**65.63%**	86.44%	90.26%	86.38%	**68.75%**	87.98%
TransferNet	90.68%	72.76%	39.06%	83.08%	91.10%	73.93%	42.19%	83.85%	91.10%	73.93%	42.19%	83.85%
SSKGQA	**91.10%**	65.76%	48.44%	82.21%	**93.32%**	71.21%	48.44%	85.10%	**97.22%**	80.54%	53.13%	**90.38%**
Ours (w/o EP)	89.43%	**82.88%**	**60.94%**	**86.06%**	89.71%	**85.60%**	60.94%	**86.92%**	89.71%	**89.49%**	60.94%	87.88%

The performance of the KGQA models on questions of varying complexities is summarized in Table 1. Our proposed model achieves an overall Hits@1 of 83.56% on the testing QA dataset. For simple 1-hop and 2-hop questions, our model achieves Hits@1 ranging from 77% to 87%, while for more complex questions, it achieves around 60% Hits@1. Compared to the models such as RoBERTa and KEQA that do not require topic entities as input, our model shows a significant improvement in Hits@k (ranging from 19% to 45%) for questions of all complexities.

To ensure a fair comparison with the models that require topic entities as input, we conducted a benchmark by replacing the entity predictor (EP) with ground truth topic entities as input to the subgraph extractor. The corresponding Hits@k results are presented in the lower section of Table 1. The results

demonstrate that our model outperforms previous works in making accurate predictions on the IoT forensic QA dataset, especially for 2-hop questions, as indicated by the Hits@1 and Hits@3 scores. Furthermore, our model achieves an overall Hits@1 result (83.56%) that is comparable to state-of-the-art KGQA models (83.08%) when utilizing the entity predictor.

Table 2. Hits@1 of Evaluated Models on Each Category

Model	SAREF	SEPSES	PAO	SLOGERT	Extra
RoBERTa	22.22%	38.94%	33.89%	6.38%	71.43%
KEQA	15.56%	43.43%	42.78%	0.00%	8.40%
Ours	**97.78%**	**81.20%**	**90.56%**	**87.23%**	**76.47%**
EmbedKGQA	75.56%	81.20%	97.22%	69.15%	84.03%
TransferNet	57.78%	82.36%	**100.00%**	**100.00%**	57.14%
SSKGQA	88.89%	82.20%	**100.00%**	74.47%	58.82%
Ours (w/o EP)	**100.00%**	**83.69%**	90.56%	87.23%	**84.87%**

Table 2 provides the Hits@1 results categorized by the answer categories of the KGQA models. Our approach outperforms the other methods in answering questions related to SAREF, SEPSES, and Extra categories, which predominantly consist of 2-hop and 3-hop+ questions. However, our method struggles with 1-hop questions in the PAO and SLOGERT categories. In these cases, TransferNet and SSKGQA achieve perfect Hits@1 scores of 100%.

We also conducted an ablation study to examine the impact of the entity predictor (EP) and relation predictor (RP) on the accuracy of our model. Table 3 presents the Hits@1 results for our model when EP and/or RP are replaced by ground truth topic entities and relations. Compared to using only the answer predictor with ground truth inputs, incorporating EP and RP results in a 4% decrease in accuracy. EP has a slightly larger impact, causing a 2.50% decrease, compared to RP with a 2.34% decrease, across questions of all complexities. However, for complex questions with three or more hops, RP has a more pronounced effect on the overall prediction accuracy compared to EP.

Table 3. Hits@1 of Proposed Model without Entity Predictor (EP) and/or Relation Predictor (RP)

Model	1-hop	2-hop	3-hop+	Overall
Ours	87.76%	77.82%	59.38%	83.56%
Ours (w/o EP)	89.43%	82.88%	60.94%	86.06% (+2.50%)
Ours (w/o RP)	87.76%	80.93%	68.75%	84.90% (+2.34%)
Ours (w/o EP & RP)	89.43%	85.99%	71.88%	87.50% (+3.94%)

5 Conclusions

We have presented ForensiQ, a framework powered by KGQA for IoT forensics, in this paper. ForensiQ transforms unstructured forensic data into comprehensible KGs. Moreover, it allows users to perform natural language queries for intricate forensic artifacts. To achieve this, we have designed an ontology specifically for IoT forensics by integrating well established ontologies in the field. Utilizing this ontology, we have constructed an experimental KG dataset for IoT forensics. Furthermore, we have developed a KGQA model based on LLM and GNN techniques, which can effectively respond to natural language questions related to the KG. The evaluation results demonstrate the superior performance of our KGQA model compared to several existing models when it comes to answering complex questions.

References

1. Alabdulsalam, S., Schaefer, K., Kechadi, T., Le-Khac, N.-A.: Internet of things forensics – challenges and a case study. In: DigitalForensics 2018. IAICT, vol. 532, pp. 35–48. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99277-8_3
2. Atlam, H.F., Hemdan, E.E.D., Alenezi, A., Alassafi, M.O., Wills, G.B.: Internet of things forensics: a review. Internet Things **11**, 100220 (2020)
3. Busbridge, D., Sherburn, D., Cavallo, P., Hammerla, N.Y.: Relational graph attention networks. arXiv preprint arXiv:1904.05811 (2019)
4. Chen, Y., Kuang, J., Cheng, D., Zheng, J., Gao, M., Zhou, A.: AgriKG: an agricultural knowledge graph and its applications. In: Database Systems for Advanced Applications, pp. 533–537 (2019)
5. Dadkhah, S., Mahdikhani, H., Danso, P.K., Zohourian, A., Truong, K.A., Ghorbani, A.A.: Towards the development of a realistic multidimensional IoT profiling dataset. In: 2022 19th Annual International Conference on Privacy, Security & Trust (PST), pp. 1–11 (2022)
6. Daniele, L., den Hartog, F., Roes, J.: Created in close interaction with the industry: the smart appliances REFerence (SAREF) ontology. In: Formal Ontologies Meet Industry, pp. 100–112 (2015)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. 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. 4171–4186 (2019)
8. Dosis, S., Homem, I., Popov, O.: Semantic representation and integration of digital evidence. Procedia Comput. Sci. **22**, 1266–1275 (2013). https://doi.org/10.1016/j.procs.2013.09.214
9. Ekelhart, A., Ekaputra, F.J., Kiesling, E.: The SLOGERT framework for automated log knowledge graph construction. In: The Semantic Web, pp. 631–646 (2021)
10. Ellison, D., Ikuesan, R.A., Venter, H.S.: Ontology for reactive techniques in digital forensics. In: 2019 IEEE Conference on Application, Information and Network Security (AINS), pp. 83–88 (2019)

11. Haller, A., Janowicz, K., Cox, S., Phuoc, D.L., Taylor, K., LefrançSois, M.: Semantic sensor network ontology. W3c recommendation, W3C (2017)
12. He, P., Zhu, J., Zheng, Z., Lyu, M.R.: Drain: an online log parsing approach with fixed depth tree. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 33–40 (2017)
13. He, S., Zhu, J., He, P., Lyu, M.R.: Loghub: a large collection of system log datasets towards automated log analytics. arXiv preprint arXiv:2008.06448 (2020)
14. Huang, X., Zhang, J., Li, D., Li, P.: Knowledge graph embedding based question answering. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 105–113 (2019)
15. Janarthanan, T., Bagheri, M., Zargari, S.: IoT forensics: an overview of the current issues and challenges. In: Montasari, R., Jahankhani, H., Hill, R., Parkinson, S. (eds.) Digital Forensic Investigation of Internet of Things (IoT) Devices. ASTSA, pp. 223–254. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-60425-7_10
16. Karagiannis, C., Vergidis, K.: Digital evidence and cloud forensics: contemporary legal challenges and the power of disposal. Information **12**(5), 181 (2021)
17. Kiesling, E., Ekelhart, A., Kurniawan, K., Ekaputra, F.: The SEPSES knowledge graph: an integrated resource for cybersecurity. In: The Semantic Web – ISWC 2019, pp. 198–214 (2019)
18. Koroniotis, N., Moustafa, N., Sitnikova, E.: A new network forensic framework based on deep learning for internet of things networks: a particle deep framework. Futur. Gener. Comput. Syst. **110**, 91–106 (2020)
19. Li, A., Wei, Q., Han, C., Xing, X.: Research on the construction of smart care question answering system based on knowledge graph. Procedia Comput. Sci. **214**, 1595–1602 (2022)
20. Li, M., Ji, S.: Semantic structure based query graph prediction for question answering over knowledge graph. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 1569–1579 (2022)
21. Li, S., Choo, K.K.R., Sun, Q., Buchanan, W.J., Cao, J.: IoT forensics: amazon echo as a use case. IEEE Internet Things J. **6**(4), 6487–6497 (2019)
22. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
23. Omar, R., Mangukiya, O., Kalnis, P., Mansour, E.: ChatGPT versus traditional question answering for knowledge graphs: current status and future directions towards knowledge graph Chatbots. arXiv preprint arXiv:2302.06466 (2023)
24. oneM2M: oneM2M Technical Specification TS-0012-V3.7.3. onem2m technical specification, oneM2M (2021)
25. Rizal, R., Riadi, I., Prayudi, Y.: Network forensics for detecting flooding attack on internet of things (IoT) device. Int. J. Cyber-Secur. Digit. Forensics **7**(4), 382–390 (2018)
26. Rohith, R., et al.: SCAPY- a powerful interactive packet manipulation program. In: 2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS), pp. 1–5 (2018)
27. Saxena, A., Tripathi, A., Talukdar, P.: 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, pp. 4498–4507 (2020)
28. Shi, J., Cao, S., Hou, L., Li, J., Zhang, H.: TransferNet: an effective and transparent framework for multi-hop question answering over relation graph. In: Proceedings

of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 4149–4158 (11 2021)
29. Sikos, L.F.: Knowledge representation to support partially automated honeypot analysis based on wireshark packet capture Files. In: Intelligent Decision Technologies 2019, pp. 345–351 (2020)
30. Sikos, L.F.: AI in digital forensics: ontology engineering for cybercrime investigations. Wiley Interdisc. Rev. Forensic Sci. **3**(3), e1394 (2021)
31. Tan, Y., et al.: Evaluation of ChatGPT as a question answering system for answering complex questions. arXiv preprint arXiv:2303.07992 (2023)
32. Tan, Y., et al.: Research on knowledge driven intelligent question answering system for electric power customer service. Procedia Comput. Sci. **187**, 347–352 (2021)
33. Wu, T., Breitinger, F., Baggili, I.: IoT ignorance is digital forensics research bliss: a survey to understand IoT forensics definitions, challenges and future research directions. In: Proceedings of the 14th International Conference on Availability, Reliability and Security (2019). https://doi.org/10.1145/3339252.3340504
34. Yun, F., Feng, Z., Baofeng, L., Yongfeng, C.: Research on intelligent fault diagnosis of power acquisition based on knowledge graph. In: 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), pp. 1737–1740 (2019)
35. Zawoad, S., Hasan, R., Skjellum, A.: OCF: an open cloud forensics model for reliable digital forensics. In: 2015 IEEE 8th International Conference on Cloud Computing, pp. 437–444. IEEE (2015)