[1,2]Khalid Jameel Munshid

An Improved Knowledge Graph Representation for Ransomware Attacks Using LSTM-Based Named Entity Relation Technique and Twitter Data

Journal of Electrical Systems

Abstract: - This research aims to examine the challenges associated with ransomware knowledge and develop a knowledge graph of ransomware attacks utilizing Twitter data. Ransomware is a constantly evolving global threat. The three main steps involved in the development of a knowledge graph from informal text are data collection and preprocessing, features extraction, and relation extraction. A ransomware ontology previously proposed has been used in this work for the extraction of the ransomware entities from unstructured data; this is aimed at making it fit the attacks reported on Twitter. The next process is the identification of the existing relationships in the dataset for the knowledge graph construction; the output of the process, which is the developed knowledge graph is evaluated for accuracy using a tracing technique to demonstrate its efficacy.

Keywords: LSTM; NER; Twitter; Ransomware; Knowledge Graph; Ontology.

I. INTRODUCTION

People and organizations are constantly exposed to ransomware attacks globally on an alarming scale [1], and these forms of attack mostly involve victims' data encryption and the subsequent demand for payment to release the decryption key. Therefore, such attacks are most times devastating as they cause disruptions and significant financial losses to the victims [2, 3]. One major issue is that despite the growing incidents of these attacks, there are things that are yet to be known about them [4, 5] as most of the existing research has mostly focused on their technical aspects by developing mostly the mechanisms for their detection and prevention [6], while neglecting the social and behavioural aspects of such attacks, such as how the social media facilitate the dissemination of information about these attacks.

An ontology for ransomware can assist in creating models that detect and monitor cyberattacks from beginning to end [7]. Ontologies establish a framework for a particular domain by outlining key classes and their associated properties. By setting the boundaries of a class through class constraints, instances inherit these boundaries. This facilitates the aggregation, portrayal, and dissemination of threat data on a large scale that would otherwise be challenging to replicate, reuse, and evaluate. Ontology can be used by both humans and software agents to understand the structure of data across various sources. [7]; it offers a common language for researchers within a field and include machine-readable definitions of core concepts and their connections.

Ransomware-as-a-Service allows anyone to create ransomware easily, leading to the current malware ontology's inability to accommodate the vast variety of ransomware types [6]. Twitter can be valuable for gathering vulnerability data since ethical hackers often report new malware findings. However, analyzing tweets presents challenges, such as the need to identify and categorize posts related to specific events [7]. Data extraction from Twitter may face issues due to short contexts, informal language, non-standard abbreviations, capitalization, and the use of hashtags, complicating the information extraction process from social networks. Therefore, there is a need for a systematic approach that arranges and represents data in such a way that makes it easy to identify potential risks as early as possible.

This paper strives to bridge this knowledge gap by introducing an innovative method for constructing graph-based knowledge on ransomware attacks utilizing deep learning that works on ransomware-related data collected from Twitter. The steps of the proposed approach involves sourcing information related to ransomware attacks, analysis of the collected information, and the building of a graph-based depiction of the observed relationships between the entities involved in these attacks. Valuable insights regarding the dynamics of ransomware attacks can be obtained

[1]Al-rafidain University College, computer science, Iraq

[2]Islamic Azad University‹Science And Research Branch,Iran

alabdekhaled@gmail.com,

from this graph representation to guide the development of better mechanisms for minimizing their effects [8]. This work relied on the concepts of graph theory and deep learning to device a mechanism for improving the development of deep graph-based representations of ransomware attacks [4, 9]. A deeper understanding of the hidden relationships between different entities involved in ransomware attacks, including the victims, attackers, and intermediaries, can be better understood using the developed graph. The use of both structured and unstructured data in this work also enabled the construction of a more comprehensive representation of ransomware attacks that better portrays their dynamics.

II. PREVIOUS STUDIES

Cybersecurity is one of the major application areas of graph-based representations where they are mostly used for malware and ransomware attacks analysis [7]. Experts rely on these representations to model the complex relationships between the different components of these attacks [10], while the analysis of the structure of these graphs allows understanding of the dynamics of these attacks [11]. Graph-based representations are mostly beneficial because they can portray the complex relationships between different entities involved in complex attacks; they can be used by scholars to model the propagation of malware through networks of targeted systems [11] [7]. Many scholars have tried to use knowledge representation techniques in many fields within cybersecurity; for instance, the study by some scholars [11-13] have reported the development of ontologies for cyberattacks' modelling on cyber supply chain systems, where these ontologies serve as a structured approach for depicting the relationships between the involved entities involved in the attacks. The nature of these cyber supply chain attacks can be understood via analysis of these ontologies. The field of knowledge management in cybersecurity is another application area of knowledge representation [10, 14, 15] where it is used to structure and organize cyber threat information [16], as well as to support decision-making and risk management processes.

Among the limitations of the existing methods of building graph-based representations of malware and ransomware attacks is their reliance on static data sources (such as logs or reports) that are not sufficient to capture the full nature of these attacks; these approaches also focus mostly on the attacks' technical aspects without considering their social and behavioural aspects. Research on the creation of TINKER, a well-designed knowledge graph for the extraction of information from unstructured threat data was reported by [17]. In this method, RDF triples was used to represent entities and their relationships; 83 threat reports published between 2006 and 2021, encompassing 3,000 triples, were used to develop the graph. The three key classes that formed the TINKER's ontology are Malware, Vulnerability, and Indicator, which are typical of malware attacks. The development and training of NER and RE models using a combination of ML and text mining algorithms for streamlining the process of annotating threat reports, was the aim of the study.

Another study by [18] on malware KGs sourced data from After-Action Reports (AARs) due to their insightful analysis of cyber incidents. Piplai created a tailored entity recognizer known as "Malware Entity Extractor" (MEE) for the purpose of extracting entities. The MEE model underwent training to anticipate cybersecurity entities within AARs, utilizing annotated cybersecurity text segments as a basis. Subsequently, the Relation Extractor (RelExt) identifies the relationships between the extracted entities by MEE [19]. Figure 1 illustrates Piplai's System Architecture. A conducted by [20] classified ransomware entities in informal text using machine learning. It proposed a method that identifies ransomware entities in casual text. Figure 2 shows the the schematic of developed system.

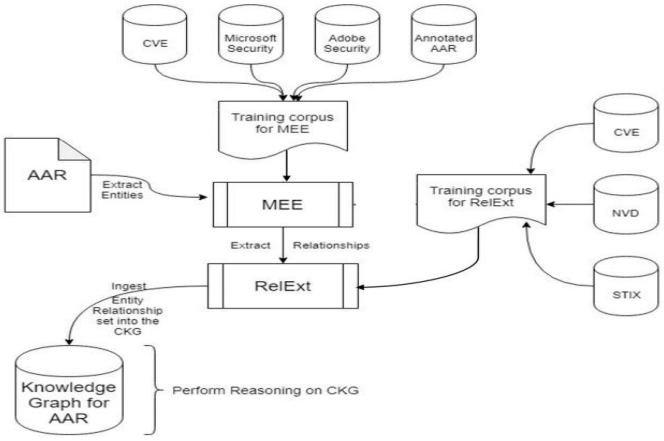

Figure 1: Ransomware entities classification model [20]

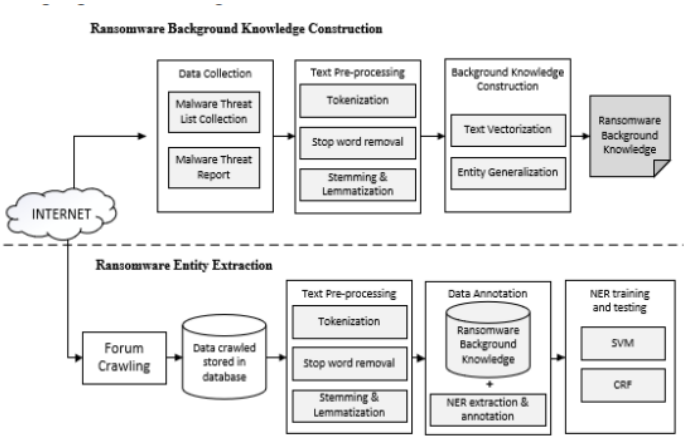

Figure 2: Ariffini scheme Retrieved from [13]

This work was conceived as a novel approach to the the building of graph-based knowledge on ransomware attacks using Twitter data that leverages the rich and dynamic data available on social media platforms such as Twitter to construct a more comprehensive representation of ransomware attacks that incorporates both technical and social aspects. The proposed method in this work allows a better understanding of the inter-relationships between the entities involved in ransomware attacks; it will also add to the knowledge of the dynamics of these attacks. The proposed analysis of ransomware attacks using these techniques and Twitter data can improve knowledge on this significant aspect of cybersecurity.

III. METHODS

The different steps of the applied methodology towards the development of the graph-based representation of ransomware attacks using Twitter data was described in this section. The involved steps include (i) data collection, (ii) pre-processing, (iii) graph construction, and (iv) graph analysis. Each of these steps are detailed and the tools used for their implementation are well discussed. This approach allows the construction of a complete representation of ransomware attacks that can aid understanding of their dynamics and nature. The performance of search engines, such as Yahoo, Google, and Baidu are mostly determined by the applied knowledge graphs which are semantic networks with directed graph structures. These graphs contain nodes and edges that represent entities and the relationships between them, respectively. This structure provides the semantic connections within a domain, as well as information networks mostly encountered in academic databases like the Web of Science and Wikidata. Also, efforts such as NELL, DBPedia, and OpenIE extracts structured data from unstructured web content to develop the knowledge bases. The concept of a knowledge graph for malware is intriguing, given the extensive variety of malware types. Research in this area is largely reliant on the Undercoffer Knowledge Graph for Intrusion Detection, adapted for malware. The framework is depicted in Figure 3.

Figure 3: An Ontology for Intrusion Detection [21]

3.1 A Contextual Named Entity Relationship Technique for Ransomware Knowledge Graph Representation

Named Entity Recognition (NER) is a specialized area within Natural Language Processing (NLP) that detects and categorizes named entities in unstructured text into standard groups. These groups include names of individuals, locations, organizations, temporal expressions, percentages, codes, quantities, and monetary amounts. For our case, the CyNER model, which is an open-source Python module developed for cybersecurity named entity recognition, is a perfect depiction of a NER model as it combines the transformer-based models for extracting entities in cybersecurity, NER models for generic entity types, and heuristics for identifying various indicators of compromise. The categorization of events as classes and extraction of malware attack information from a threat intelligence corpus has been reported in previous research, such as MALOnt2.0 [22] and MALOnt. These approaches allows the combination of predictions from different approaches to meet specific needs. The training of the NER model using the Long Short-Term Memory (LSTM) networks was described in this section; LSTM network was considered for this purpose because it can model long-term dependencies between named entities and their context, making it suitable for NER tasks; it can also be trained for the identification and classification of named entities within new text data as it can make predictions relying on the input feature representations. The proposed approach is made up of 4 steps which are (i) data collection, (ii) pre-processing, (iii) feature extraction, and (iv) model training; the first step is the collection of a large labelled dataset that consist of annotated text data with named entity labels. The next step is the pre-processing of the data and extraction of features to reflect the named entities. The LSTM network uses these extracted features as input as it has been trained to rely on such features to predict the named entity labels. The internal parameters of the network are adjusted during training to keep the prediction error low.

3.2 Data Collection, preprocessing, and cleaning

Data collection is an important step towards building a successful graph-based knowledge representation; this is because they data quality impacts the model's performance in many ways. This first step of the proposed method involved collection of a large corpus of Twitter data on ransomware attacks using the Twitter API and scraping tweets based on the Python libraries such as Tweepy and Twint. The search involved the use of keywords and hashtags related to ransomware attacks and the collected data represents all types of tweets that relay information regarding ransomware attacks; the information must also contain cases of named entities from the predefined categories that can be recognized by the model after training. The processing and cleaning of the collected data is done in the second step to get the data ready and fit for the proposed graph-based knowledge representation. The steps involved in this phase are tokenization, stop-word removal, and stemming or lemmatization; redundant information was also removed from the tweets at this stage. The steps are detailed below:

- Noise removal: Performed to eliminate irrelevant entries in the textual data, such as HTML tags, symbols, and emojis. Removing these elements reduces ambiguity and improves model training.

- Tokenization: Splits text into smaller units, like words or phrases, making it easier to analyze and process the textual data during stemming.

- Stop word removal to eliminate the common words in a language that don't add significant meaning to a sentence. They are often removed during text preprocessing to improve analysis.

- Stemming and lemmatization both extract root words from different forms. Stemming removes suffixes, consolidating words with similar meanings, while lemmatization identifies the common root despite differences in form. Lemmatization helps locate all instances of a word in various forms, making sentences easier to analyze.

After these, WordNet, a large lexical database for English, was used to pre-process the cleaned data; this process was done to ensure the standardization of the vocabulary to improve the data quality. At the end of this step, a cleaned dataset of tweets on ransomware for the construction of the proposed graph-based knowledge representation is achieved (see Figure 4 for the detail of this phase).

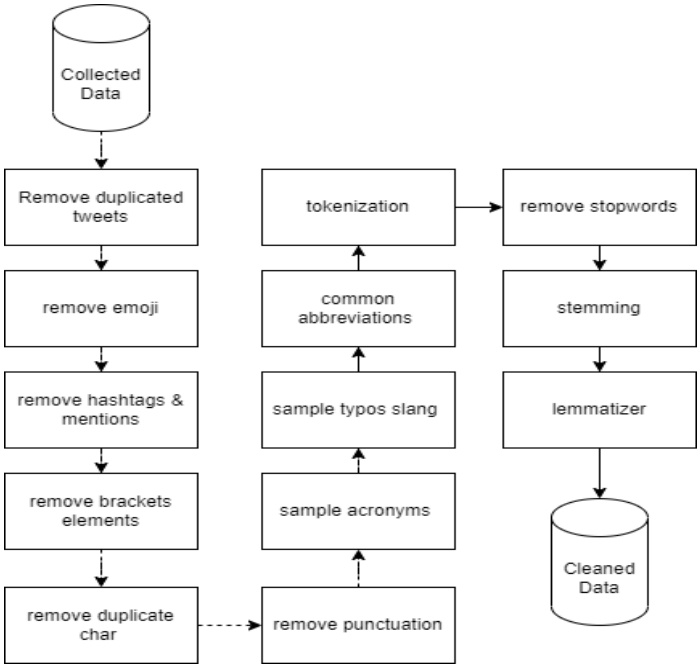

Figure *4*. **Path coefficients and significance (Assaggaf et al., 2023)**

3.3 Entity Extraction

Entity extraction involves identifying and classifying key pieces of information such as hashtags, mentions, and keywords related to ransomware threats. These entities are then used to create nodes and edges in the knowledge graph, which represents the relationships between the different pieces of information. By using this approach, we can create a comprehensive and structured representation of ransomware threats based on real-time data from Twitter that can be easily queried and analyzed. During the preprocessing stage, we extracted ransomware entities from the data using an ontology as described in [20]. After tagging and tokenizing the data, training of the LSTM was conducted. The entities in the ontology are shown in Table 1 below.

Table 1: Entity Table

Entity	Description	Example
Ransomware Name	Name of the ransomware	Intermittent
Ransomware family	Name of the ransomware family in which the sample belongs to	LockFile
Target	The company or program has been infected by a ransomware attack.	MS Windows, Mac OS, GitHub

3.4 Long Short-Term Memory Training for the Contextual Named Entity Relationship Technique

The data was then fed into the LSTM network in a format suitable for training. The LSTM network is then trained on this data using supervised learning techniques, taking into account the context of each tweet. During training, the model learns to recognize patterns in the data and associate them with the corresponding entities. Once the model has been trained, it was used to extract entities from new ransomware-related tweets.

Training an LSTM model involved several steps and the selection of various parameters and layers, including an input layer, hidden layer(s), and an output layer. The pre-processed data is received in the input layer and forwarded to the hidden layer. Each hidden layer consists of multiple LSTM cells, which have three gates - input, forget, and output gates and the role of these gates is to coordinate information flow through the cell. During training, the adjustable parameters are the number of hidden layers and the number of LSTM cells per layer. Figure 5 shows the LSTM gate structure.

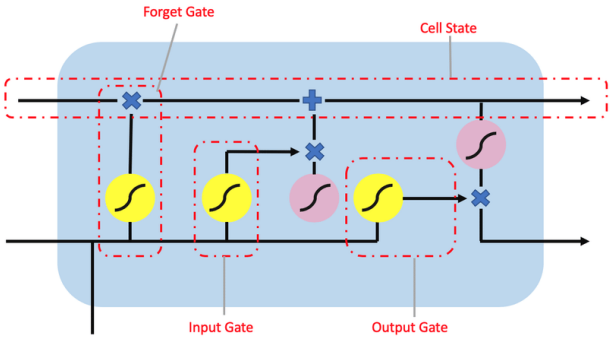

Figure 5. LSTM gates architecture (Qian et al., 2022)

During training, the weights of the connections between the LSTM cells are adjusted using backpropagation through time (BPTT), a variant of the backpropagation algorithm that is used for training RNNs. The learning rate, which controls the step size during weight updates, was another hyperparameter that was adjusted during training.

Several hyperparameters were adjusted experimentally for the LSTM model as follows.

i- Embedding layer:

- Embedding size: 150, as Twitter data is generally shorter and less complex than other text sources.

ii- LSTM layer:

- Number of LSTM layers: 2 layers.
- Number of hidden units per LSTM layer: 80, as Twitter data is typically less complex.
- Dropout rate: 0.6, to help prevent overfitting.
- Bidirectional LSTM: Bidirectional LSTM was used to capture the both past and future context in the text.
- Activation function: Relu function

iii- Dense output layer:

- Number of output units: Equal to the number of unique named entity and relationship classes in the dataset.
- Activation function: Softmax, to generate probabilities for each class.

iv- Model training:

- Loss function: Categorical cross-entropy, as it is suitable for multi-class classification tasks.
- Optimizer: Adam, with a learning rate of 0.001.

- Batch size: 32, to accommodate the shorter length and possible limitations in the dataset size.
- Number of epochs: 30, with using early stopping with a patience of 5 epochs to prevent overfitting.

v- Regularization:

- Regularization: L2 regularization was applied on the weights of the LSTM and dense layers, using a small weight decay factor (1e-5) to prevent overfitting.

Once the model has been trained, it becomes ready to make predictions on new data. The output layer produces a probability distribution over the possible entities for each input sequence.

3.5 Design the Knowledge Graph

Extracted entities from the contextual LSTM were used by RRelExt, dedicated to extract relationships between ransomware entities. As there are only a few ransomware entities, we can determine and link these entities without using machine learning to extract basic relationships. To build an NER model, the open-source spaCy toolkit was used for advanced NLP in Python. It's designed for production use and helps develop systems to analyze large amounts of text. The text annotation and training process includes five steps: annotating data, converting it to a spaCy bin object, creating a configuration file, training the model via command line, and loading and testing the stored model. Figure 6 shows an example of SpaCy toolkit output.

Figure 6. Schema of the knowledge graph

Initially, the online spaCy NER annotation tool was adopted to annotate the text. The figure below illustrates an example of the annotated data. We then proceeded with the development of the knowledge graph project by utilizing the Python programming language and leveraging the capabilities of the KGEMs and PyKEEN libraries. These are well-known libraries owing to their effectiveness in knowledge graph creation and have demonstrated high accuracy in their use [23]. The schema of the knowledge graph and system design are shown in Figures 7 and 8, respectively.

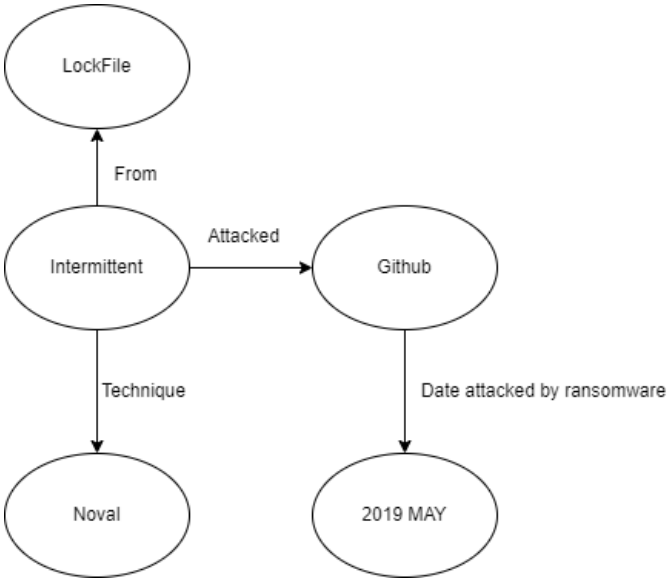

Figure 7. Schema of the knowledge graph

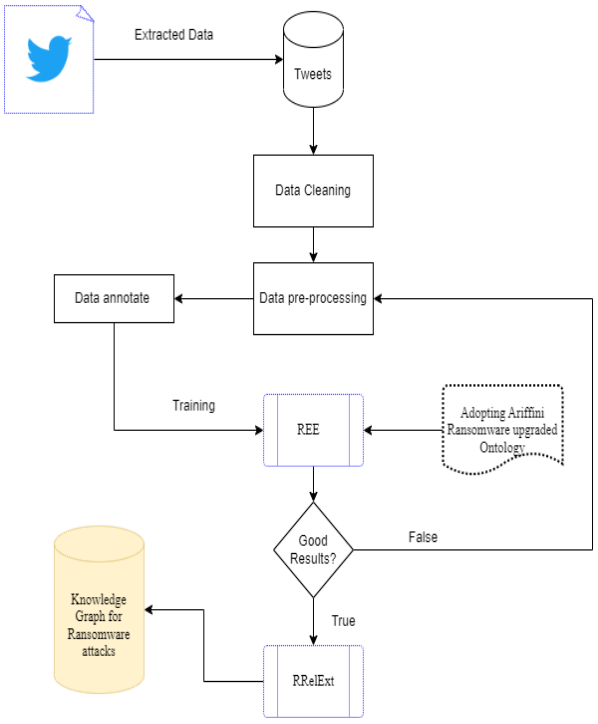

Figure 8. System design

To build the Knowledge Graph for ransomware, it is necessary to develop the Relation Entity Extractor function. This requires defining entity pairs and their relationships, ensuring that at least two entities are present within a single tweet. Table 2 showcases the entity pairs and their corresponding relationships.

Table 2: Description of Entity relations

Ransomware Name	Infect	Target
Target	InfectedOn	Date
Ransomware Name	BelongTo	Ransomware Family
Ransomware Name	Uses	Technique
Target	Damages	1TB of data

At first, we stored all entities and their corresponding tweet dates in a single CSV file. The date entity is included in all nodes but is only utilized when linked to a target entity. The proposed knowledge graph was developed using Neo4j, and online database that simplifies the storage and management of sensitive information by providing access control for systems, applications, and individuals. The data structure of the graph enhances personal data visualization and analysis of data pattern. The schema of the proposed KG in Neo4j is depicted in Figure 9.

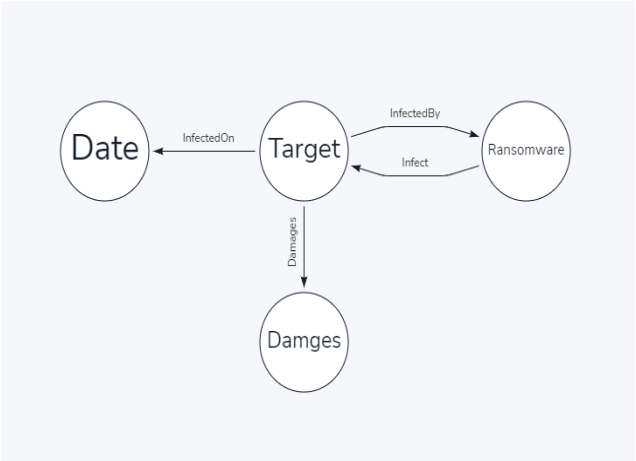

Figure 9: neoj4 KG schema

3.6 Evaluation

The methodology for evaluating the performance of a Knowledge Graph involves assessing accuracy using steps from Gao, et al. [24]. The process begins with entity identification and relation validation. Then, three steps are followed: 1) Sample Collector selects a small batch of samples from the Knowledge Graph, 2) Sample Pool includes all samples drawn and requests manual annotations for new models, and 3) Estimation component calculates an unbiased estimate of Knowledge Graph accuracy and margin of error based on human annotations and the sampling design. 4) Quality checks to ensure the user-specified margin of error and confidence interval are reached. If satisfied, the process ends; otherwise, it returns to Step 1.

3.7 Data Collection

The aim at this stage is to realize objective (a) via the collection of ransomware attacks related data using the Twitter API; this can be achieved using Tweepy, a Python library that allows easy access Twitter's public APIs; it helps users to manage various low-level tasks such as rate limiting, authentication, HTTP requests, serialization, etc. Tweepy handles these details for the user, reducing the likelihood of errors. However, it is essential to be aware that Twitter enforces a rate limit on API requests, allowing a maximum of 900 requests every 15 minutes. Any additional requests will result in an error.

The evaluation was performed using precision, recall, and F1-score for each class, which are the primary evaluation metrics for NER training results. These metrics rely on true positives (TP), false positives (FP), and false negatives (FN). TP are class members correctly classified, while FP are those wrongly classified into a specific class. FN are cases not labeled as part of any class but actually belong to one. TN represent instances where events are not classified when the given conditions are absent. Using these indicators, we calculateed Precision, Recall, Accuracy, and F1-Score as follows:

Accuracy = TP + TN / (TP + TN + FP + FN)

Precision = TP/(TP + FP)

Recall = TP/(TP + FN)

F1=2(Precision)(Recall)/(Precision + Recall)

IV. RESULTS AND DISCUSSIONS

To achieve our research goal, we utilized the Twitter API to collect data. Despite facing numerous challenges such as tweet quality and patterns, we successfully gathered up to 1 million tweets. To address these issues, we implemented early filtering using search queries and optimized our dataset by selecting the 85,000 most relevant tweets for our topic. Figure 10 displays the distribution of the accounts from which we collected the tweets.

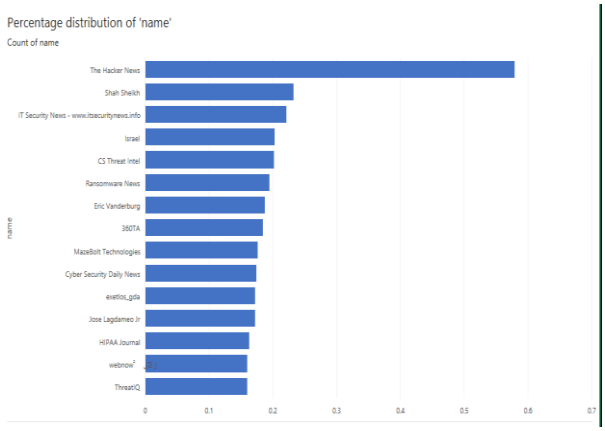

Figure 10: Representation of accounts used as data sources

The collected data then underwent preprocessing, aimed at removing background noise and conducting initial processing. Text preprocessing prepares data for model building and is the starting point for all NLP projects. It readies raw text for use as input in models by eliminating textual noise such as emojis, punctuation, and

capitalization errors. These elements need to be removed since they are ineffective when generated by machines. Figures 11 and 12 show a text sample before and after preprocessing, demonstrating a significant difference. Uncleaned text can cause issues during model training as it may contain incomprehensible characters or be written in incorrect formats.

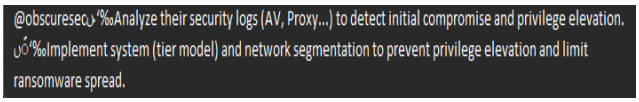

Figure 11: Non-processed tweet

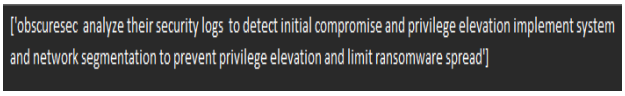

Figure 12: Processed tweet

To assess the NER model, we manually evaluated 1000 tweets that were processed by the model, from which we extracted 178 entities. Table 3 displays the entity prediction performance using the proposed contextual LSTM-based NER technique in terms of accuracy, precision, recall, and F1 score. We tested the model with three different dropout score to investigate the best parameter to avoid the overfitting.

Table 3: Confusion matrix table

Metric	**Dropout (0.5)**	**Dropout (0.5)**	**Dropout (0.5)**
Accuracy	93.42	92.22	90.03
Precision	91.24	90.31	88.41
Recall	92.33	91.14	89
F1 Score	94.03	93.15	91.23

Table 3 reveals that the best performance was 93.42 for accuracy, 91.24 for precision, 92.33 for recall, and 94.03 for F1 score. This highest performance was achieved when we set the dropout rate to 0.5. Likewise, the table shows that the lowest performance as 90.03, 88.41, 89, and 91.23 for accuracy, precision, recall, and F1 score, respectively. This indicates the best dropout value was 0.5.

Figure 13 compares the performance of the proposed contextual LSTM-based NER with other prediction techniques used by related works. The x-axis represents the performance metrics while the y-axis represents the performance value expressed in percentage (%). We can observe that the proposed model outperformed the previous studies for all metrics. This confirms the efficacy of the use of LSTM to discover the hidden contextual relations between the entities in the Twitter text. This is attributed to the ability of the LSTM to capture the temporal and spatial relations within the data and the inference power it gains by the inclusion of the context.

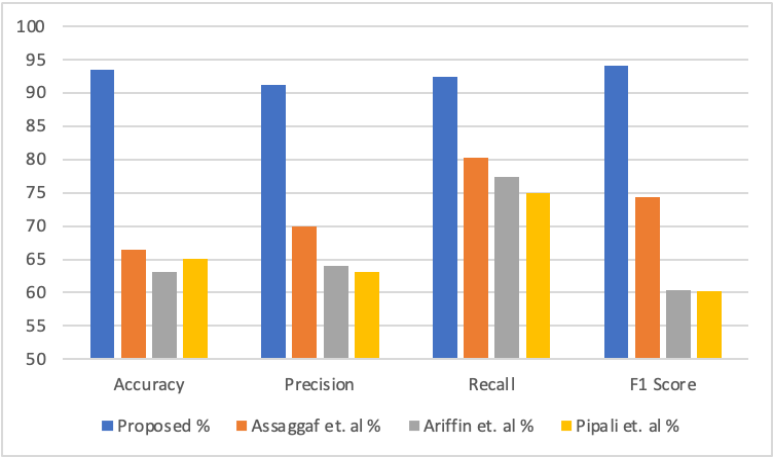

Figure 13: The performance comparison between proposed and related works

Neo4j was used to construct a Knowledge Graph in this work; initially, the most relevant cases to display were determined, which required some manual filtering and the consolidation of all entities into a single CSV file. We also linked related entities to specific events for simultaneous connection. After importing 615 rows of data and replacing empty entities with unknowns, we created nodes and their relationships. Neo4j automatically eliminated duplicate nodes, resulting in a final count of 536 nodes and 1,437 connections. Figure 14 illustrates the graph depicting node infect relations. In many instances, the extent of damage remains undisclosed as affected companies often choose not to reveal this information publicly.

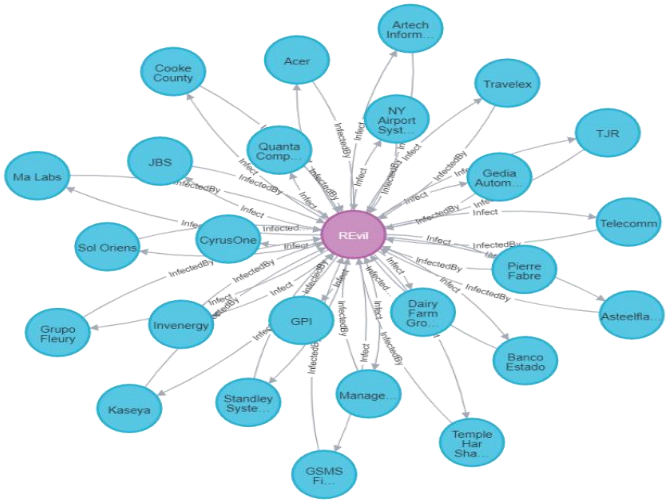

Figure 14: Node's infect relations.

V. CONCLUSION

This study demonstrated the application of a contextual deep learning-based NER technique for enhancing the knowledge-graph representation of ransomware infections. Through the achievement of four key objectives, we have made significant strides in extracting valuable insights from a vast dataset of one million tweets. The development of our deep learning-based NER model, which achieved an acceptable score of 74%, has proven its usability and effectiveness in extracting relevant entities from pre-processed and cleaned informal text. This accomplishment paves the way for more accurate and comprehensive knowledge-graph representations, ultimately benefiting cybersecurity researchers, practitioners, and organizations in understanding and mitigating ransomware attack risks. The third objective is the building of the final KG using Neo4j, showcasing ransomware attacks reported on Twitter. By manually mapping entity relations and effectively managing the data, our approach resulted in 536 nodes and 1,437 relations. This knowledge graph serves as a valuable resource for understanding ransomware-related events and their connections. Lastly, we proposed an evaluation schema for assessing the knowledge graph, which can be utilized by experts in the field for further analysis and refinement. Although we are not experts in this domain, our research provides a foundation for future work, promoting continued advancements in knowledge-graph construction and evaluation. In summary, our research underlines the importance and potential of utilizing contextual deep learning-based NER techniques for improving knowledge-graph representations of ransomware infections. This work contributes to ongoing efforts in understanding and combatting the ever-evolving threat of ransomware attacks, equipping researchers and practitioners with state-of-the-art tools and techniques for effective analysis and mitigation.

REFERENCES

[1] Y. A. Ahmed *et al.*, "A Weighted Minimum Redundancy Maximum Relevance Technique for Ransomware Early Detection in Industrial IoT," *Sustainability*, vol. 14, no. 3, p. 1231, 2022.

[2] Peng, Ciyuan, et al. "Knowledge graphs: Opportunities and challenges." *Artificial Intelligence Review* 56.11 (2023): 13071-13102.

[3] Al-Majdi, Kadhum, et al. "MLCM: An efficient image encryption technique for IoT application based on multi-layer chaotic maps." *International Journal of Nonlinear Analysis and Applications* 13.2 (2022): 1591-1615.

[4] B. A. S. Al-Rimy *et al.*, "Redundancy coefficient gradual up-weighting-based mutual information feature selection technique for crypto-ransomware early detection," *Future Generation Computer Systems*, vol. 115, pp. 641-658, 2021.

[5] Y. A. Ahmed, B. Koçer, S. Huda, B. A. S. Al-rimy, and M. M. Hassan, "A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection," *Journal of Network and Computer Applications*, vol. 167, p. 102753, 2020.

[6] Taha, Mustafa Sabah, et al. "A steganography embedding method based on P single/P double and Huffman coding." 2021 3rd International Cyber Resilience Conference (CRC). IEEE, 2021.

[7] A. M. A. Assaggaf, B. A. Al-Rimy, N. L. Ismail, and A. Al-Nahari, "Development of Graph-Based Knowledge on Ransomware Attacks Using Twitter Data," in *Data Science and Emerging Technologies: Proceedings of DaSET 2022*: Springer, 2023, pp. 168-183.

[8] Y. A. Ahmed, B. Kocer, and B. A. S. Al-rimy, "Automated analysis approach for the detection of high survivable ransomware," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 5, pp. 2236-2257, 2020.

[9] Ophoff, Jacques, and Mcguigan Lakay. "Mitigating the ransomware threat: a protection motivation theory approach." Information Security: 17th International Conference, ISSA 2018, Pretoria, South Africa, August 15–16, 2018, Revised Selected Papers 17. Springer International Publishing, 2019.

[10] 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.

[11] I. R. Chowdhury and D. Bhowmik, "Capturing Malware Behaviour with Ontology-based Knowledge Graphs," in *2022 IEEE Conference on Dependable and Secure Computing (DSC)*, 2022: IEEE, pp. 1-7.

[12] M. Keshavarzi and H. R. Ghaffary, "An ontology-driven framework for knowledge representation of digital extortion attacks," *Computers in Human Behavior*, vol. 139, p. 107520, 2023.

[13] O. Adekanmbi, H. Wimmer, and A. Shalan, "Semantic Web Ontology for Botnet Classification," in *Semantic Intelligence: Select Proceedings of ISIC 2022*: Springer, 2023, pp. 43-54.

[14] K. Liu, F. Wang, Z. Ding, S. Liang, Z. Yu, and Y. Zhou, "A review of knowledge graph application scenarios in cyber security," *arXiv preprint arXiv:2204.04769*, 2022.

[15] 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.

[16] J. Yin, M. Tang, J. Cao, M. You, H. Wang, and M. Alazab, "Knowledge-driven cybersecurity intelligence: software vulnerability co-exploitation behaviour discovery," *IEEE Transactions on Industrial Informatics*, 2022.

[17] S. Dutta, N. Rastogi, D. Yee, C. Gu, and Q. Ma, "Malware knowledge graph generation," *arXiv preprint arXiv:2102.05583*, 2021.

[18] 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. 211691-211703, 2020.

[19] A. Pingle, A. Piplai, S. Mittal, A. Joshi, J. Holt, and R. Zak, "Relext: Relation extraction using deep learning approaches for cybersecurity knowledge graph improvement," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2019, pp. 879-886.

[20] N. Ariffini, A. Zainal, M. A. Maarof, and M. N. Kassim, "Ransomware Entities Classification with Supervised Learning for Informal Text," in *2019 International Conference on Cybersecurity (ICoCSec)*, 2019: IEEE, pp. 86-90.

[21] 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*, 2003: Springer, pp. 113-135.

[22] R. Christian, S. Dutta, Y. Park, and N. Rastogi, "An Ontology-driven Dynamic Knowledge Graph for Android Malware," 2021.

[23] M. Ali *et al.*, "PyKEEN 1.0: a python library for training and evaluating knowledge graph embeddings," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 3723-3728, 2021.

[24] J. Gao, X. Li, Y. E. Xu, B. Sisman, X. L. Dong, and J. Yang, "Efficient knowledge graph accuracy evaluation," *arXiv preprint arXiv:1907.09657*, 2019.