2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Realtime Disaster Detection Through GNN Models Using Disaster Knowledge Graphs

Seonhyeong Kim
Computer Science and Engineering
Kyungpook National University
Daegu, Korea
kimsh951027@knu.ac.kr

Irshad Khan
Computer Science and Engineering
Kyungpook National University
Daegu, Korea
irshad.cs@knu.ac.kr

Young-Woo Kwon
Computer Science and Engineering
Kyungpook National University
Daegu, Korea
ywkwon@knu.ac.kr

***Abstract*—In the context of the increasing scale and complexity of disasters caused by rapid climate change, a comprehensive understanding of disaster big data is essential for effective detection and response. The disaster knowledge graph proposed in this paper fills this gap by capturing the connections between various disaster-related data sources and their potential for growth across heterogeneous datasets. We generate time-series disaster graphs every minute using SNS data (e.g., Twitter) and public data, specifically focusing on disasters. Then, we create disaster knowledge graphs to represent the relationships between various data sources and try to predict their potential developments. We label and annotate knowledge graphs and then detect sudden changes in time-series disaster knowledge graphs for disaster detection. To that end, we assess the effectiveness of three state-of-the-art GNN models for graph-based event classification using Graph Convolutional Network (GCN), Graph Attention Network (GAT), and SageConv. In addition, we evaluate a simple clustering model, K-means, for comparison. Our experiments show promising results with approximately 87% precision in detecting disaster events using structural data and connectivity patterns within disaster graphs. Finally, we measure the result of disaster detection time with an unseen dataset, showing positive results that about 70% detect a disaster in less than 3 minutes. To comprehensively analyze real-time social media data and understand the patterns of disaster to enhance disaster management and response strategies, our approach combines the strength of GNNs with a designed disaster knowledge graph.**

***Index Terms*—knowledge graphs, graph neural networks, disaster detection**

I. Introduction

With the recent increase in the number of disasters worldwide, there is a significant loss of life and property [1]. To minimize such damages, collecting, managing, and utilizing disaster data has become an important issue [2]. For disaster management, it is necessary to process structured and unstructured big data created in the private and public domains into meaningful information to detect and respond to disaster fast [3]. Since disaster data is generated on numerous platforms, it is essential to manage it in an integrated system so that various organizations can access and use it properly.

Social network services(SNS) became a powerful communication tool during disasters [4]. We can monitor the emergency site, plans of evacuation, and communications from individuals seeking assistance. During a disaster, we must receive precise and trustworthy information [5]. As a result, there is a growing trend in research that focuses on analyzing the structure and significance of social networks in disaster response. However, not all information obtained during emergencies is relevant or beneficial. It is essential to extract meaningful information and utilize it effectively as a communication tool in emergencies. Furthermore, it plays a significant role in connecting community members in the aftermath of a disaster to develop a mitigation plan.

Therefore, effectively representing the relationship between the data generated during emergencies is essential. Knowledge graphs are widely used in research, particularly in efficiently organizing knowledge from big data to represent concepts and infer new valuable knowledge from them [6]. Knowledge graphs are mainly used for knowledge representation, and there are many applications using knowledge graphs in various domains, such as link prediction, recommendation systems, and natural language processing [7]. These graphs provide a simple yet powerful framework by leveraging the rich network of relationships captured within the graph structure, which can be the input to machine learning models [8]. This paper introduces a novel approach for constructing a disaster knowledge graph that effectively represents the relationship between a disaster's big data. The proposed method involves generating real-time graphs using the knowledge graph structure. In addition, we present a disaster occurrence detection model, which utilizes a graph-based neural network to analyze the knowledge graph and identify disaster events.

The main contributions of this work are summarized as follows:

- We present a time-series disaster knowledge graph that effectively represents the relationships among various types of disaster big data. In our prior work [9], we defined a basic architecture of a disaster knowledge graph using nodes, edges, and their labels. Here, we improve

ASONAM '23, November 6-9, 2023, Kusadasi, Turkey

ACM ISBN 979-8-4007-0409-3/23/11…$15.00
https://doi.org/10.1145/3625007.3627514

and redefine the structure of the disaster knowledge graph according to the schema concept of a previously defined knowledge graph.

- We assess multi-layer graph neural network models to classify time-series disaster graphs to detect anomalies and find similar disasters through pattern analysis. We employ three prominent GNN architectures: GCN, GAT, and SageConv.
- We design a comprehensive disaster big data management system. The system facilitates the collection, organization, and utilization of disaster-related data, enabling a better understanding of disaster situations and supporting effective response and recovery efforts.

These contributions collectively aimed to enhance the analysis and utilization of disaster data, enabling more accurate anomaly detection and informed decision-making in disaster management. To evaluate the system, we use earthquake events to test the effectiveness of our strategy by comparing the detection capacities of several graph neural networks. The experiments showcase promising outcomes in identifying disaster events by leveraging the structural information and connectivity patterns within the disaster knowledge graphs. Additionally, the detection times of these events are evaluated using an unseen dataset, and the results are encouraging, indicating the potential for timely and accurate detection of disasters using the proposed approach.

The rest of this paper is structured as follows. Section II shows the related works, and section III describes the proposed disaster big data management system. Section IV represents the structure of the disaster knowledge graph and evaluates the generated graphs with GNN models in Section V and VI. Finally, Section VII presents the summary of this work.

II. Related Work

There is a study on an ontology-based framework for heterogeneous data management with urban flood disasters [10]. The concept of data is extracted based on the ontology framework and analyzes the most significant impact on disasters. Another study uses a knowledge graph to analyze various types of typhoon information by correlating them [11]. The constructed knowledge graph is visualized based on location, time, wind speed, wind volume, and air pressure to enable an overall understanding of typhoons that occurred at the same time or in the exact location. These studies showed the connection between concepts; however, they are limited by the lack of real-time data, which could pose difficulties in comprehending and responding to disaster situations. Therefore, automatic data collection is required to overcome these limitations and facilitate a better understanding of disaster scenes.

The utilization of social media for managing and analyzing disasters is a growing area of study [12]. Recognizing the situation as soon as possible is critical to minimize the damage [13]. Consequently, social media platforms, which provide immediate access and efficient communication during disasters, are widely used to assess the situation. They play a vital role in such events' communication, response, and recovery efforts. However, there are challenges in generalizing and accurately labeling input data to train algorithms due to various data. Moreover, the propagation of false information is a significant concern. Therefore, a comprehensive system that can analyze various information is needed to better understand complex disaster situations.

III. Integrated Management of Disaster Bigdata

This section addresses the comprehensive management of disaster big data. It outlines the system flow and explains the analysis methods using graphs for different data types.

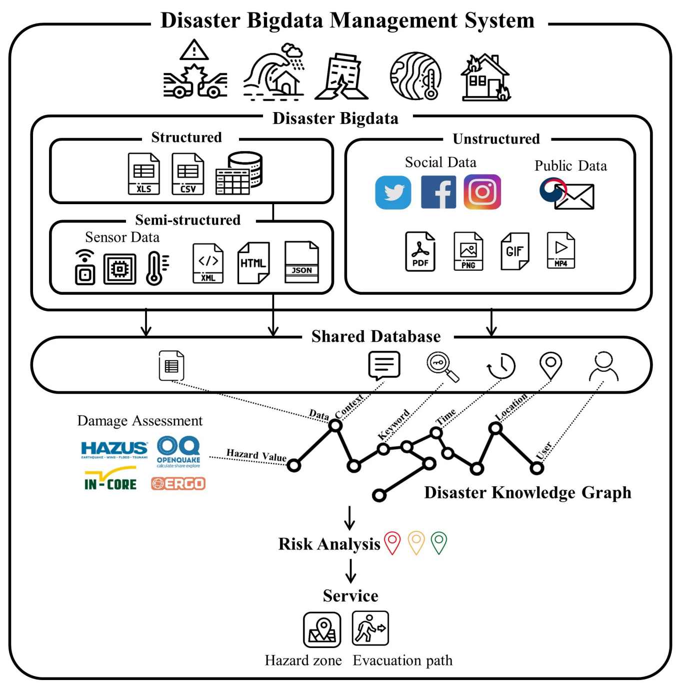


Fig. 1. Disaster bigdata management system

A. Disaster Bigdata Management System

Figure 1 shows the layered architecture of a disaster big data management system. Numerous types of disasters, including natural disasters, social disasters, and accidents, can occur independently or in correlation. These disasters generate diverse data types that are distributed and have different formats. The disaster data encompass structured and unstructured data, necessitating an integrated management system to effectively analyze the various data types and better understand the disaster situation.

This study used a graph concept to manage data in an integrated manner. The graph structure represents objects as nodes and their relationships as edges. It enables efficient correlation identification between different data types and facilitates quick data retrieval. The collected data can then be utilized to analyze the extent of damage caused by the disaster. Calculate the risk through the damage assessment platform,

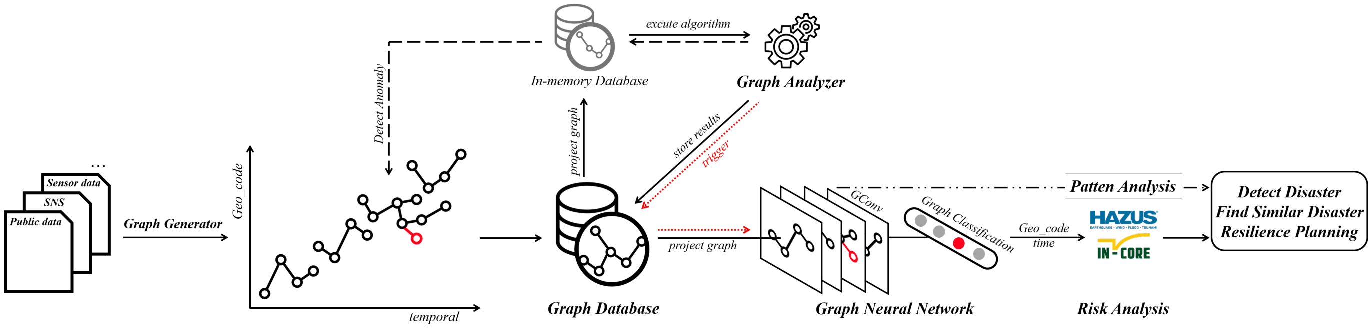


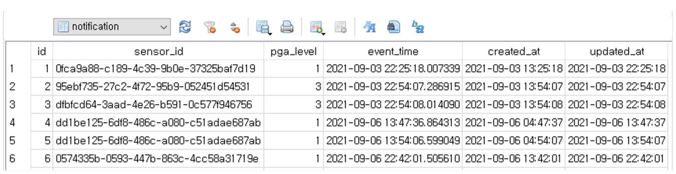
Fig. 2. A process for disaster big data analysis

and based on the analyzed results, we can provide services to dangerous disaster sites to help recovery.

The process of data analysis to detect a disaster is illustrated in Figure 2. We represent the relationship between data and generate a disaster knowledge graph in a time series. We calculate risk values with graph properties and train Graph Neural Network models to detect and analyze the patterns of a disaster. Finally, we find similar disasters and define the final risk for resilience planning.

B. Disaster Bigdata

Structured and Semi-structured Data: Disaster-related structured data are stored in XLS, CSV, or relational databases (RDBMS). It has a structure in which values are stored in tables according to the schema structure, and there are relations between tables through relationships. In addition, there are semi-structured data in files such as XML, HTML, and JSON or data collected from accelerometers, pressure, temperature, and fire sensors. Figure 3 shows the structured data stored in RDBMS.


notification

	id	sensor_id	pga_level	event_time	created_at	updated_at
1	1	0fca9a88-c189-4c39-9b0e-37325baf7d19	1	2021-09-03 22:25:18.007339	2021-09-03 13:25:18	2021-09-03 22:25:18
2	2	95ebf735-27c2-4f72-95b9-052451d54531	3	2021-09-03 22:54:07.286915	2021-09-03 13:54:07	2021-09-03 22:54:07
3	3	dfbfcd64-3aad-4e26-b591-0c577f946756	3	2021-09-03 22:54:08.014090	2021-09-03 13:54:08	2021-09-03 22:54:08
4	4	dd1be125-6df8-486c-a080-c51adae687ab	1	2021-09-06 13:47:36.864313	2021-09-06 04:47:37	2021-09-06 13:47:37
5	5	dd1be125-6df8-486c-a080-c51adae687ab	1	2021-09-06 13:54:06.599049	2021-09-06 04:54:07	2021-09-06 13:54:07
6	6	0574335b-0593-447b-863c-4cc58a31719e	1	2021-09-06 22:42:01.505610	2021-09-06 13:42:01	2021-09-06 22:42:01

Fig. 3. Structured data about sensor information

Unstructured Data: Disaster-related unstructured data encompass various formats, including social media, news articles, and public data. This data also includes text, images, or videos. As a result, additional analysis is required to derive meaningful insights from such data. Table I and figure 4 show the example of Twitter and emergency alert messages, which are unstructured data.

Unlike structured data, unstructured data cannot be stored in a relational database with a fixed schema. Instead, they are stored in a non-relational database, especially a NoSQL

TABLE I
EXAMPLE OF TWEETS

date	user	tweet	hashtags	place	retweets
1:42 AM Jul 6, 2023	@user	@Fishery collapse, lake eutrophication may well happen soon ...	#collapse #eutroph-ication	@location	77

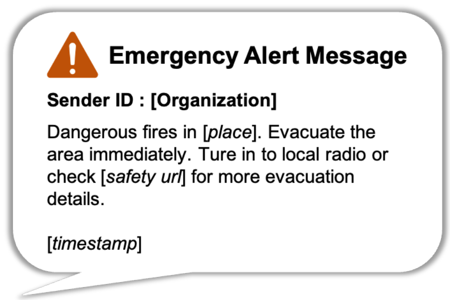


Fig. 4. Example of emergency alert message

database, which offers flexibility and ease of storing and processing unstructured data.

C. Data Storage

We use a graph database to store structured and unstructured disaster data in one storage and perform integrated management. A graph database expresses major data elements as nodes in the graph and has relationships that explain the connections between nodes. Each node and edge can have labels and their properties. In addition, we can set the weight to edge according to the importance. The relationship between the collected disaster data can be expressed in a graph to visualize the relationship between the data. Moreover, it is possible to check how disasters change and spread through graphs created based on the location or time of data generation.

D. Graph Data Analysis

When a disaster occurs, structured, semi-structured, and unstructured data is generated, and the relationship between data is expressed in a graph. In addition, the graph expresses and visualizes the relationship between data based on the location or time of data creation. A desired sub-graph can be projected to an in-memory database to perform pattern analysis using various algorithms, and the analyzed results can be stored in the database again. We construct several graph-based neural network models using the analyzed graph to quickly and precisely identify disaster situations. We can find similar historical disasters through pattern analysis, which helps with resilience planning.

IV. Disaster Knowledge Graph

A knowledge graph can represent a relationship between heterogeneous data. Since various types of disaster data are generated in a disaster situation, we design a knowledge graph structure to express their relationships. Nodes and edges are defined according to the schema concept of KBpedia [14], including the concept of a previously defined knowledge graph. Figure 5 shows the structure of the proposed disaster knowledge graph. A disaster domain has natural disasters, social disasters, and accidents as a sub-class. Various data, such as social and public data, are generated when a disaster occurs. Additionally, disasters cause multiple damages, including physical injury and social harm. These disaster knowledge graphs were inspired from structures used in DBpedia, Wikidata, and Wikipedia.

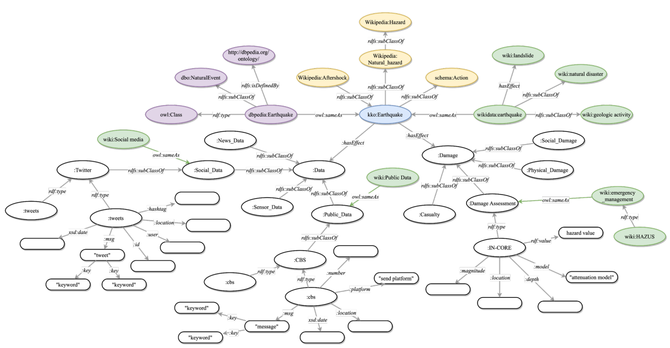


Fig. 5. The structure of a disaster knowledge graph

A. Knowledge Graph Structure

Node, Edge, and Label: The entities are represented as nodes in the disaster knowledge graph with class, instance, and property types. Table III describes the defined node, type, and labels. Disaster information includes sensor, social, public, and news data. Also, damages caused by disasters include physical damage, social damage, casualties, and damage assessment information. In addition, nodes with the same label are derived from one node and can have multiple labels to distinguish between them. For example, important keywords extracted from data are stored as nodes with key labels. Therefore, since each platform can have a key value, it has a double label to distinguish between key nodes.

And edge represents the relationship between entities. As we defined in our prior work [9], nodes and edges have their labels. According to RDF, RDFS, OWL, and XML schema definitions, which are web ontology language standards released to build web ontology, we define edges:‘:subClassOf’, ‘:isDefinedBy’, ‘:type’, ‘:value’, ‘:sameAs’, ‘:date’, and ‘hasEffect’. The defined edge is shown in table III. Each edge can have an attribute value and set weights according to its importance.

Node, Edge, and Weight: Each node and edge can have an attribute value, and we set weights according to their importance. In the case of the ’Twitter’ sub-graph, we store collected tweets in each instance node. Date, tweet, ID, user, location, and hashtag information collected from the tweets are stored on each node, and important keywords are extracted from the tweet. The weight of the tweet edge is determined according to the keyword importance. When generating a Twitter node, the default weight value of the edge is 0.5. The edge weight value of the node, including terms related to a disaster, emotional expression, or exclamation, is set as 1. Additionally, the node with a meaningless tweet is set as 0.

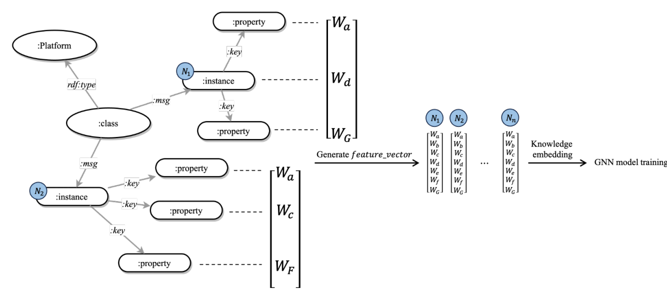
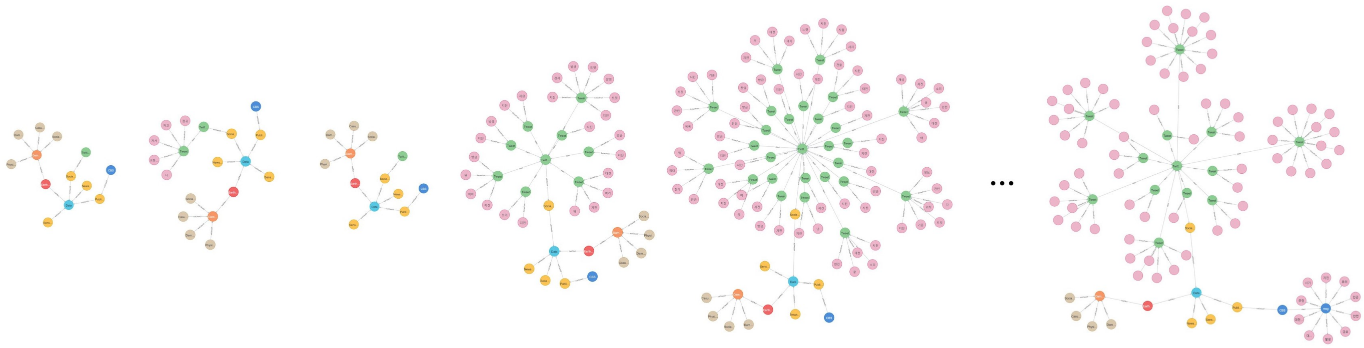


Fig. 6. Knowledge embedding

Knowledge Embedding: The knowledge of nodes and edges can be represented as vectors that are created by calculating the edge value based on the keyword importance and use the node degree centrality value. A vector produced based on the number of overlapping keywords is also added

TABLE II
Node and Label

Label	Node	Type
Information	:Data	Class
	:Damage	Class
Dataset	:Social_Data	Class
	:News_Data	Class
	:Public_Data	Class
	:Sensor_Data	Class
Damages	:Casualty	Class
	:Damage_Assessment	Class
	:Physical_Damage	Class
	:Social_Damage	Class
SNS	:Twitter	Instance
Public	:CBS	Instance
Twitter:Key	:keyword	Property
CBS:Key	:keyword	Property



(a) (b) (c) (d) (e) (f)

Fig. 7. Time-series disaster knowledge graph shows a 1-minute graph, a disaster occurs in (d) 09:44:00 ˜ 09:45:00) and issues an emergency alert message in (f) 10:06:00 ˜ 10:07:00)

TABLE III
EDGE AND LABEL

LABEL	EDGE
RDFS	:SUBCLASSOF :ISDEFINEDBY
RDF	:TYPE :VALUE
OWL	:SAMEAS
XSD	:DATE
	:HASEFFECT

to the knowledge embedding by some important predefined keywords. The knowledge embedding is depicted in figure 6.

B. Graph Database

To monitor disaster situations in real-time and detect their occurrences, a disaster knowledge graph is generated and stored every minute. We first start with a basic knowledge graph. Then, data collected from Twitter, a social media, is stored in the 'Twitter' sub-graph, and emergency alert messages issued by governments are stored in the 'Cbs' sub-graph. We crawl the text messages through the open API provided by the government, and keywords are extracted using a Korean morpheme analyzer (KoNLpy). In addition, we calculate the degree of centrality of the 'Twitter' node to detect a sudden increase in social media data. We calculate the degree centrality using the Graph Data Science library provided by Neo4j. This paper uses an earthquake as an example disaster to construct a time-series disaster knowledge graph. Figure 7 shows the graph generated for a disaster on April 8, 2023, around 9:44 a.m. As shown in Figure 7, the number of nodes in the 'Twitter' sub-graph increased noticeably from the time of occurrence in Figure (d), and the 'Emergency Alert Messages' sub-graph was generated about 20 minutes after the occurrence of the disaster as shown in figure (f).

V. MULTI-LAYER GRAPH NEURAL NETWORK FOR DISASTER OCCURRENCE DETECTION

We employ three prominent GNN architectures, GCN, GAT, and SageConv, to classify event-based graphs into earthquake and non-event categories. These models enable the extraction of meaningful representations by leveraging graph structure and incorporating neighborhood interactions. These models are effective regarding node interactions in a graph, as they have different mechanisms, such as attention, sampling, and graph convolutions, to capture both local and global information, which are essential to graph structure and node representation [15]. Additionally, these models can handle variable graph structures. Here is a brief description of each model.

A. GCN

GCN is a robust architecture that uses neighboring nodes to spread information as it operates on graph data [16]. It leverages a graph structure to aggregate features from neighboring nodes, enabling effective representation learning. The update rule for a single layer in a GCN is defined as follows:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \quad (1)$$

where $H^{(l)}$ represents the hidden feature matrix at layer l, $\hat{A}$ is the normalized adjacency matrix of the graph, $W^{(l)}$ is the weight matrix of the l-th layer, and σ is the activation function (ReLU). Multiple layers can be stacked to form a deep GCN. The degree centrality and attribute values of each graph node and edge can be expressed as a vector through vector embedding.

Figure 8 and Table IV show the structure of a GCN model, which typically ends with one or more fully connected layers and maps the learned node features to a set of output classes or regression values.

B. SageConv

SageConv (GraphSage) is a variant of GNN that learns node representations through sampling and aggregation oper-

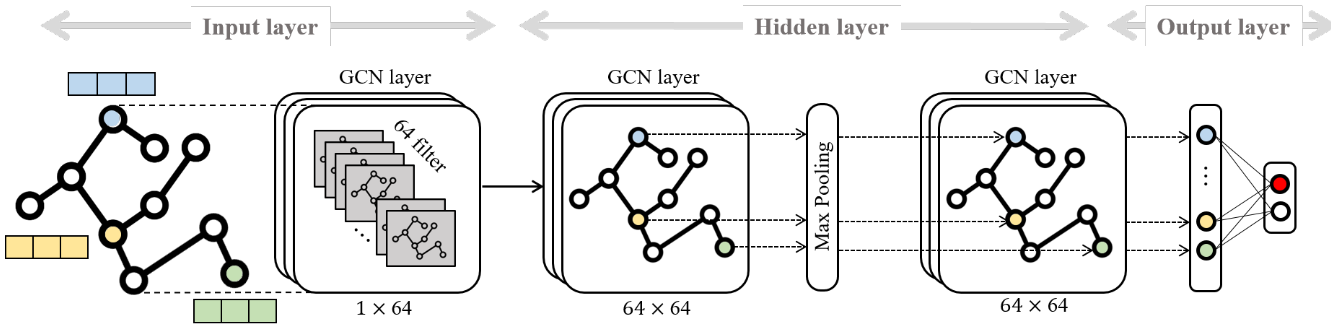


Fig. 8. GCN-model architecture

TABLE IV
GCN MODEL SPECIFICATION

Layer	Details
Input	$n_{\text{features}} \to 64$
GraphConv	$64 \to 64$
MaxPooling	-
GraphConv	$64 \to 64$
Output	$64 \to 2$

ations [17]. It leverages aggregating information from local and global neighborhoods, enabling effective representation learning for graph classification. SageConv defines the update rule for a single layer as follows:

$$h_i^{(l+1)} = \sigma(\text{AGGREGATE}(h_i^{(l)}, \{h_j^{(l)} | j \in \mathcal{N}(i)\})) \quad (2)$$

here $h_i^{(l)}$ represents the hidden feature vector of the node i at layer l, $\mathcal{N}(i)$ is the set of neighbors of node i, AGGREGATE is an aggregation function (e.g., mean or max pooling) that combines the feature vectors of the central node and its neighbors, and σ is the activation function.

SageConv typically performs multiple iterations of the above update rule to refine node representations. Figure 9 represents the aggregate approach of the model.

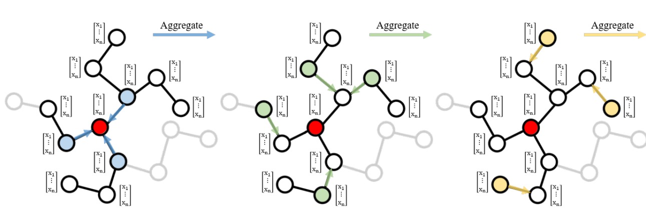


Fig. 9. Aggregate approach of SageConv

C. GAT

GAT incorporates attention mechanisms into the graph convolution operation, allowing each node to weigh the importance of its neighbors during information propagation dynamically [18]. Through this, GAT can capture local and global dependencies in the graph, enhancing its expressive power for graph classification tasks.

In a GAT, the following is the definition of the update rule for a single layer:

$$h_i^{(l+1)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l)} \quad (3)$$

here $h_i^{(l)}$ represents the hidden feature vector of node i at layer l, $\mathcal{N}(i)$ is the set of neighbors of the node i $\alpha_{ij}^{(l)}$ is the attention coefficient between nodes i and j, computed as a softmax function of their feature similarities. While $W^{(l)}$ is the weight matrix of the l-th layer.

GAT employs multi-head attention to capture different patterns and relationships. Figure 10 shows the architecture of a GAT model.

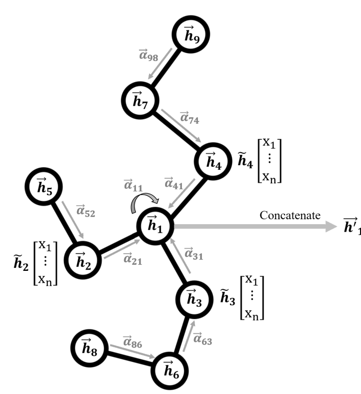


Fig. 10. GAT model architecture

VI. EVALUATION

For the evaluation, we select an earthquake as one of the disasters. Even though we do not limit the types of disasters, we begin with earthquakes that happened in South Korea for this study to show the effectiveness of our approach. We

Performance: A performance measure provides an essential quantitative evaluation of a classification model's accuracy and effectiveness [19]. Utilizing a confusion matrix that encompasses various classification results, including true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), enables a comprehensive assessment of four classification models using our disaster knowledge graphs.

Further performance metrics that offer additional insights into the efficiency of the classification model include precision, recall, and F1-score [20]. Precision measures the model's ability to correctly identify positive instances out of all those classified as positive. High precision indicates a low rate of false positives. Recall, also known as sensitivity or true positive rate, quantifies the model's ability to identify all positive instances correctly. High recall signifies a low rate of false negatives. F1-score considers both precision and recall, comprehensively evaluating the model's accuracy and robustness. A higher F1 score indicates a better performance.

Dataset: Based on the disaster knowledge graph discussed above, we created a customized dataset tailored specifically for earthquake event detection. The dataset incorporates various data sources, including social media and emergency alert messages sent from the government to capture information on earthquake events. Social data like Twitter contain textual content, user interactions, and temporal patterns. These features provide valuable insights into the occurrence and characteristics of earthquake-related discussions on social media.

We selected 50 earthquakes for the experimental setup to construct disaster knowledge graphs. We used 40 earthquakes for model training; the dataset specification is described in Table V. The earthquake events included in the graph were derived from a ten-year period from 2013 to 2023. Specifically, earthquakes with a magnitude(M.) of 3.0 and above; whereas intensity 1.0 is our minimum threshold for selecting earthquakes, ensuring their perceptibility and potential impact on human activities.

We generated 120 graphs for each earthquake at 1-minute interval for an hour before and after each disaster to train four models. A labeling mechanism was employed to ensure proper labeling of the generated graphs. Each graph is assigned a label based on its temporal relationship to the occurrence of an earthquake. Specifically, disaster knowledge graphs created before the earthquake are labeled as 'non_eq' to indicate the absence of an earthquake event. Graphs corresponding to when the earthquake occurred are labeled 'eq' to signify the actual earthquake event. Furthermore, graphs representing the period up to 10 minutes after the earthquake are labeled as 'after_eq' to capture the immediate aftermath of the seismic activity.

Results: We generated a graph dataset and stored it in the Neo4j graph database. We then exported the data in a GraphML format for model training and testing. We selected 40 disasters of earthquake for the model training. For each earthquake event, 120 subgraphs were created based on minute intervals. To evaluate the performance of our models, we conducted tests using 75% of the data for training, 20% for testing, and reserved 5% for validation. To classify the graphs based on their structure and specification, we used GNN models and applied the K-means clustering model [21] as a baseline method for the comparison. Table VI summarizes the models' test performance. Precision is the ratio of positive predictions that contain real disasters. The highest precision was attained by GCN, 87.22%, demonstrating its ability to classify earthquake graphs accurately. The precision of 78.02% achieved by GraphSAGE demonstrates its mediocre performance in differentiating between event and non-event graphs. GAT and K-means' precision of 82.99% and 82.34% demonstrates its effectiveness in correctly identifying earthquake event graphs.

TABLE V
DATASET SPECIFICATION(M. IS A MAGNITUDE OF EARTHQUAKE)

Intensity	M.	Graphs	Nodes	Edges	'eq'	'non_eq'
> 1	>=3	4800	524841	524840	1025	3775

TABLE VI
SUMMARY OF THE MODELS TEST PERFORMANCE

Model	TP	FP	TN	FN	Acc.	Pre.	Recall	F1-score
GCN	70	8	747	135	85.10	87.22	66.54	75.49
GSG	80	33	722	125	83.54	78.02	67.33	70.23
GAT	67	15	740	138	84.06	82.99	65.35	68.66
K-means	69	17	738	136	84.06	82.34	65.70	69.02

The model's ability to accurately classify positive instances (earthquake event graphs) while reducing false positives. Regarding reducing false positives, GCN performed better than other models. There is always a precision and recall tradeoff, as demonstrated in the recall column of the performance table, where GSG recall is higher than GCN, which is 67.33% and 66.54%, respectively. Overall, the comparative analysis highlights the superior performance of the GCN model in terms of accuracy, precision, recall, and F1-score. The GraphSAGE model shows moderate performance across the metrics, while the GAT model demonstrates competitive accuracy with GCN but relatively lower precision, recall, and F1-score.

Moreover, we utilized a dataset comprising recent disastrous events not included in the models' training phase. Among the 50 disasters selected as a dataset, 10 disasters not used for model training were selected as unseen data. This ensured that the models were tested on unseen data, allowing us to assess their generalization capabilities. For this purpose, we used 10 disasters and monitored the models' performance at specific intervals (1 minute) to gauge their ability to detect disasters over time. Table VII shows the disaster detection results with 10 unseen disasters. The 1-min column indicates the percentage of disasters detected by each model within the first minute of monitoring. The Non-detected column displays the percentage of disasters missed by each model. Among the models evaluated, the GSG model exhibited the highest recall rate, achieving a recall of 67.33%. This indicates that within the first minute, the GSG model successfully detected 40% of the disasters in the unseen dataset, and within three minutes, it managed to detect 90% of the disasters. On the other hand, the GCN model showed relatively lower performance, with a recall rate of 30% within the first minute and 50% within two minutes. The GAT model exhibited a similar recall rate of 30% within one minute but had a lower performance in subsequent minutes, detecting only 20% of disasters. The K-means model performed moderately well, with a 30% recall rate within one minute and 50% within two minutes, but it struggled to detect disasters after the four-minute mark.

TABLE VII
DISASTER DETECTION RESULTS WITH UNSEEN DATASET

Model	1-min	2-min	3-min	Over 4-min	Non-detected
GCN	30%	50%	0%	10%	10%
GSG	40%	40%	10%	10%	0%
GAT	30%	20%	0%	40%	10%
K-means	30%	50%	0%	20%	0%

VII. Conclusion and Future Work

In this paper, we presented a novel approach for managing and analyzing data generated from multiple sources to recognize and respond to disasters through a knowledge graph. The relationships between various aspects of disaster big data were captured by designing a disaster knowledge graph structure, enabling comprehensive monitoring and analysis of nationwide or worldwide disasters. To evaluate the effectiveness of the GNNs using disaster knowledge graphs in predicting the incidence of disasters, we examined three different GNNs.

For future work, there are several promising directions to explore. First, further analysis can be conducted on the generated graphs' node and edge attribute values. This analysis will provide deeper insights into the specific characteristics and patterns associated with different types of disasters, potentially enhancing the accuracy and effectiveness of disaster detection and response. Additionally, the precise labeling of the generated graphs can be pursued to provide more detailed and accurate information about the occurrences of disasters. By assigning specific labels to different periods before, during, and after an event, disasters' temporal dynamics and impact can be better understood and utilized in decision-making processes. The proposed work establishes a solid foundation for further research and development in disaster management and analysis using graph-based approaches. By continuously advancing analysis techniques leveraging graph neural networks, we can make significant strides in enhancing disaster preparedness, response, and mitigation efforts.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Ministry of Education (No. 2021R1I1A3043889) and Ministry of Science and ICT (No.2021R1A5A1021944) and also partially supported by the Digital Innovation Hub project supervised by the Daegu Digital Innovation Promotion Agency (DIP) grant funded by the Korea government (MSIT and Daegu Metropolitan City) in 2023 (No.DBS1-03).

References

[1] M. Zhang and J. Wang, "Trend analysis of global disaster education research based on scientific knowledge graphs," *Sustainability*, vol. 14, no. 3, p. 1492, 2022.

[2] S. Akter and S. F. Wamba, "Big data and disaster management: a systematic review and agenda for future research," *Annals of Operations Research*, vol. 283, pp. 939–959, 2019.

[3] C. Fan, C. Zhang, A. Yahja, and A. Mostafavi, "Disaster city digital twin: A vision for integrating artificial and human intelligence for disaster management," *International Journal of Information Management*, vol. 56, p. 102049, 2021.

[4] N. Dragović, . Vasiljević, U. Stankov, and M. Vujičić, "Go social for your own safety! review of social networks use on natural disasters–case studies from worldwide," *Open Geosciences*, vol. 11, no. 1, pp. 352–366, 2019.

[5] J. Kim and M. Hastak, "Social network analysis: Characteristics of online social networks after a disaster," *International journal of information management*, vol. 38, no. 1, pp. 86–96, 2018.

[6] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, p. 112948, 2020.

[7] X. Chen, H. Xie, Z. Li, and G. Cheng, "Topic analysis and development in knowledge graph research: A bibliometric review on three decades," *Neurocomputing*, vol. 461, pp. 497–515, 2021.

[8] S. Bhatt, A. Sheth, V. Shalin, and J. Zhao, "Knowledge graph semantic enhancement of input data for improving ai," *IEEE Internet Computing*, vol. 24, no. 2, pp. 66–72, 2020.

[9] S. Kim and Y.-W. Kwon, "Construction of disaster knowledge graphs to enhance disaster resilience," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 6721–6723.

[10] Z. Wu, Y. Shen, H. Wang, and M. Wu, "An ontology-based framework for heterogeneous data management and its application for urban flood disasters," *Earth Science Informatics*, vol. 13, pp. 377–390, 2020.

[11] P. Liu, Y. Huang, P. Wang, Q. Zhao, J. Nie, Y. Tang, L. Sun, H. Wang, X. Wu, and W. Li, "Construction of typhoon disaster knowledge graph based on graph database neo4j," in *2020 Chinese Control And Decision Conference (CCDC)*. IEEE, 2020, pp. 3612–3616.

[12] M. Martínez-Rojas, M. del Carmen Pardo-Ferreira, and J. C. Rubio-Romero, "Twitter as a tool for the management and analysis of emergency situations: A systematic literature review," *International Journal of Information Management*, vol. 43, pp. 196–208, 2018.

[13] V. Pekar, J. Binner, H. Najafi, C. Hale, and V. Schmidt, "Early detection of heterogeneous disaster events using social media," *Journal of the Association for Information Science and Technology*, vol. 71, no. 1, pp. 43–54, 2020.

[14] KBpedia, Accessed Nov. 18, 2022. [Online]. Available: https://kbpedia.org/

[15] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

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

[17] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

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

[19] S. Aronoff *et al.*, "Classification accuracy: a user approach," *Photogrammetric Engineering and Remote Sensing*, vol. 48, no. 8, pp. 1299–1307, 1982.

[20] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[21] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.