Human-centric Computing and Information Sciences

March 2022 | Volume 12

www.hcisjournal.com

KIPS Korea Information Processing Society

KIPS CSWRG Korea Information Processing Society Computer Software Research Group

HCIS | Human-centric Computing and Information Sciences

RESEARCH (Research Manuscript) Open Access

Human-centric Computing and Information Sciences (2022) 12:12
DOI: https://doi.org/10.22967/HCIS.2022.12.012
Received: December 17, 2021; Accepted: January 25, 2022; Published: March 15, 2022

Terrorist Organization Identification Using Link Prediction over Heterogeneous GNN

Micaela Lucia Bangerter, Giuseppe Fenza, Mariacristina Gallo, Vincenzo Loia*, Alessandra Petrone, and Alberto Volpe

Abstract

Terrorist attacks are threats to undermine state security and citizens' confidence. In the last year, the Member States of European Union reported a total of 57 completed, failed, and foiled terrorist attacks, with 21 people killed. Counter-terrorism activities, through intelligence analysis experts, attempts to face, tackle, and prevent new attacks. In this sense, Artificial Intelligence demonstrates promising support for data analysis and patterns identification in security concerns. This work treats the application of a deep learning approach for the association between attacks and perpetrator groups, which is often unknown. Identifying the most involved actors helps extract inherited features and better study attacks to implement suitable countermeasures and predict feature events. Starting from the well-known resource related to anti-terrorism operations, Global Terrorism Database (GTD), we build a knowledge graph (KG) representing entities and relationships involved in terrorist attacks. Subsequently, we adopted the KG to train a graph neural network (GNN) to identify terrorist organizations from events using the inductive link prediction technique. The experimentation, conducted by adopting the HinSAGE framework, demonstrates promising performance in terms of accuracy with a discrete improvement to state-of-the-art.

Keywords

Counter-Terrorism, Graph Neural Network, Link Prediction, HinSAGE, Knowledge Graph

1. Introduction

After the attacks on the United States by Al-Qaeda on 11 September 2001, the number of deaths from terrorist attacks, perpetrated mainly by religious extremists, has increased fivefold. Last year, the Member States of European Union (EU) reported 57 completed, failed and foiled terrorist attacks. The UK reported 62 terrorist incidents and Switzerland reported two potential jihadist terrorist attacks. In 2020, a total of 21 people were killed in terrorist attacks in the EU. Three people died in the UK and one in Switzerland. With the exception of the targeted murder of a teacher in France, the fatal victims appear to have been randomly selected as representatives of populations identified as enemies for ideological reasons [1].

***Corresponding Author:** Vincenzo Loia (loia@unisa.it)
Department of Management & Innovation Systems, University of Salerno, Salerno, Italy

Counter-terrorism could be defined as a series of activities, including techniques, tactics, or strategies implemented by either governments, police, military, or companies to prevent terrorism. Examples of activities are Web crawling for violent or propagandist websites identification, managing shared (inter-)national terrorism lists, strengthening cooperation among states, etc.

Artificial intelligence (AI) has globally attracted much interest as a tool that can process large amounts of data and discover patterns and correlations invisible to the human eye, increasing effectiveness and efficiency in analyzing complex information. As a general-purpose technology, such benefits can also be used in the field of counter-terrorism with multiple uses like facial recognition, biometric application of AI-powered object recognition, etc. [2, 3]. Other types of applications adopt techniques such as natural language processing (NLP) to identify individuals at risk of radicalization in online communities. It could facilitate appropriate investigations and interventions into an increasingly relevant online phenomenon that is difficult to detect using traditional law enforcement methods. The experimentation with AI technologies has begun in several law enforcement agencies around the world.

In this scenario, this work exploits knowledge contained in one of the most comprehensive attacks repository available, namely, the Global Terrorism Database (GTD), to associate the perpetrator name to a new attack. GTD is an open-source database that provides information on international terrorist attacks worldwide since 1970 and actually includes more than 200,000 events. A wide range of information is available for each event, including the date and location of the incident, the weapons used, nature of the target, number of killed and injured people, and, when identifiable, the responsible group.

The proposed methodology starts with generating a knowledge graph (KG) containing nodes and relationships that describe the characteristics of each terrorist attack. This KG is then used to train a learning model that, through a graph neural network (GNN) and a link prediction layer, predicts the existence of a relationship between the terrorist event and a perpetrator group.

Two experiments were carried out: firstly, we selected a sub-graph of the overall KG containing only groups that made more than five attacks; secondly, the entire KG was adopted. The second test was used to verify if performance decreases when adding more information to the KG. Results show good performance in both experimentations with a substantial Accuracy improvement with respect to existing literature.

The main contributions of this research work are:

- A graph-based representation of the knowledge available in a database widely used in the area of counter-terrorism, i.e., the GTD.
- KG embedding through a GNN architecture for enabling terrorist organization identification as link prediction problem.
- Improvement of the performances wrt the baseline approaches promising potential future developments of GNN applied to counter-terrorism problems as outlined in the conclusions.

The remaining part of the manuscript is structured as follows. Section 2 discusses the state-of-the-art about counter-terrorism solutions adopting GTD and the recent applications of deep neural networks (DNNs) in the form of GNNs. Section 3 introduces theoretical aspects like GNN and the GraphSAGE and HinSAGE frameworks. Section 4 exposes the proposed methodology, Section 5 details experimentation and result, and Section 6 provides the conclusion and future work.

2. Related Work

This section explores the existing literature focusing on counter-terrorism solutions, especially ones adopting the GTD, and recent applications of DNNs in the form of GNNs

2.1 Counter-Terrorism through GTD

GTD is often subjected to predictions analysis through machine learning models. Agarwal et al. [4] worked on analyzing the GTD dataset and made predictions on different aspects. They implemented

various data mining techniques and machine learning algorithms such as support vector machine (SVM), random forest, and logistic regression to predict factors such as the involved group, the success of terrorist attacks, and the effect of external elements. Uddin et al. [5] concluded that DNN is the best suitable model for predicting the behavior of terrorist activities and, in particular, to work with features about suicide, success, weapon, region, and attack type by experimenting with different machine learning (ML) and deep learning (DL) models.

Regarding the objective of this thesis work (i.e., prediction of the perpetrator group), Talreja et al. [6] trained several ML algorithms, including decision tree, random forest, and SVM, on a subset of the GTD for the Indian attacks. The goal was to predict the perpetrator of a terrorist attack, giving input attributes that described the event, including the types of attack, target, weapons, location, and year. ALfatih et al. [7] made a similar analysis for predicting the responsible group of terrorism attacks using only tree-based models, specifically, random forest, decision tree, and gradient-boosting method.

To the best of our knowledge, there are no existing research works regarding the adoption of GNN to predict attributes related to terrorist attacks and, in particular, the perpetrator group. Moreover, other structured terrorism data sources (e.g., RAND Database of Worldwide Terrorism Incidents [RDWTI]) are lesser complete than GTD and are no longer adopted by recent research works [8, 9].

2.2 Graph Neural Networks

GNNs are applied to many data mining tasks: ranking, clustering, classification, similarity measures, recommender systems, and link prediction.

Huang et al. [10] used GNN for text classification, a fundamental NLP problem, with many applications like SPAM detection and news filtering. Their proposal was a GNN based method with a text-level graph for each input text.

Clustering based on networked data (e.g., community detection) has been substantially investigated because it can lead to a better understanding of both hidden structures of the network and the individual role that each object plays in each cluster. For example, Sun et al. [11] exposed a clustering process on a heterogeneous bibliographic network, which splits the original network into several layers (a set of sub-network clusters).

The link prediction technique is widely used for recommender systems and many other real-world applications. A typical example regards friends' recommendation in social networks. Saracco et al. [12] propose a statistically-based solution that links node pairs based on their similarity. In [13], the authors propose a link prediction method aiming to protect users' private information.

In addition to social networks, Link Prediction was likewise applied to networks connecting citation. For example, Liu et al. [14] utilized keywords to predict the citation relationship between papers. In addition, Zhou et al. [15] proposed an improved random walk restart (RWR) algorithm for citation network prediction.

Regarding the actual pandemic context, the contact tracing applications had been used to identify individuals near to those infected with SARS-Cov2 (severe acute respiratory syndrome-coronavirus-2). However, people were uncertain about installing the application or updating their health condition status on these applications because of privacy issues. Ganesan et al. [16] presented ideas of explainable link prediction over GNN that could improve trust in these applications and encourage adoption by the research community.

Samizadeh and Minaei-Bidgoli [17] have used node embeddings in a heterogeneous network to obtain the information of each node and then use a binary classifier like logistic regression to predict drug and target interactions. The tool helps scientists scale down huge experimental space, reduce costs, enable faster drug development, and predict the side effects and potential function of new drugs in the pharmaceutical industry.

This work exploits a heterogeneous KG to construct and train a GNN aiming to predict the authors of a terrorist attack. In this sense, in [18], a similar approach is adopted for the name–face association problem.

3. Theoretical Background

This section details some theoretical aspects relevant to understanding the proposed methodology; in particular, it introduces GNNs; then details the adopted framework (i.e., HinSAGE) as an extension of the GraphSAGE framework based on the GNN.

3.1 Graph Neural Network

GNN is a class of DNNs that operates on the graph structure and is designed to infer inherited data. The basic foundation regards the analogy between neural networks and graphs; encoding leverages the graph structure represented by links between nodes.

Being $G = (V, E)$ a graph in which V are sets of nodes and E the edges between nodes. Being $ne[v]$ the neighbors of $v \in V$ (i.e., the nodes linked to v by an edge), and $co[v]$ the set of edges having v as a vertex.

The GNN aims to learn a state embedding $h_v \in R^S$ encapsulating information of the neighborhood of each node. The state embedding h_v is a vector that enables the production of the output o_v, being the node class or label.

This work adopts the HinSAGE framework, an extension for heterogeneous graphs of the GraphSAGE[19] network, described following.

3.2 GraphSAGE

GraphSAGE is a framework for inductive representation learning on large graphs. It generates low-dimensional vector representations for nodes and is especially useful for graphs with rich node attribute information.

Starting from an objective node $v \in V$, the fixed set of neighborhoods u^k is sampled as follows [20]:

$$\begin{gathered} u^0 = \{v\}, \\ u^k = \cup_{v \in u^{k-1}} S(A_v, N^k), k = 1,2,\ldots,K \end{gathered} \tag{1}$$

where A_v is the set of neighbors of v, and N^k is the sample size at k-th depth. $S(A_v, N^k)$ is set through the uniform distribution $\cup\ (1, deg(v))$. The initial node embeddings, h_u^0 for a sampled set u consists of node input features x_v of dimension M:

$$h_v^0 = x_v, \qquad \forall\, v \in v \cup u^1 \cup \ldots \cup u^K \tag{2}$$

The *mean_concat* averages the embeddings, $h_{v \in N(u)}^{k-1}$, of the neighboring nodes, $\mathrm{N}(u)$, of a set of sampled node u. Then, that aggregated neighbor embedding is concatenated with the embedding h_u^{k-1} of a node u to represent a new embedding h_u^k into the node:

$$\begin{gathered} for\ k = 1, 2, \ldots, K\ do \\ for\ u \in v \cup u^1 \cup \ldots \cup u^{K-k}\ do \\ h_u^k = \sigma\left\{\left(W_v^k \sum_{v \in N(u)} \frac{h_v^{k-1}}{|v|}\right) || \left(W_u^k h_u^{k-1}\right)\right\} \end{gathered} \tag{3}$$

where W_v^k and W_u^k are weight matrices shared among nodes in the network layer k. Their size is $M' \times M$ at the first layer and $M' \times M'$ at the other layers. M' is the dimension of hidden feature, and $\sigma(\cdot)$ is a non-linear function (e.g., a rectified linear unit), defined as $max(0, x)$. The operator $||$ corresponds to two

vectors concatenation. Then, the new embedding, h_u^k, is normalized. Finally, when the processing of the K-layer finishes, a final embedding vector, h_u^K, is generated and adopted by the classifying layer to make predictions. The GraphSAGE model objective, during the training, regards the minimization of classification cross-entropy loss defined as follows:

$$L(\hat{y}, y) = -\sum_{v \in V} \sum_{i=1}^{C} y_i \, log \, log \, \hat{y}_i \, , \forall y \in Y. \tag{4}$$

3.3 HinSAGE

Heterogeneous SAGE (HinSAGE) extends the GraphSAGE algorithm to generate embeddings from heterogeneous graphs. So, it is able to discriminate between multiple nodes and edges when aggregating and concatenating embeddings of neighbors: a relevant aspect when working with heterogeneous graphs. HinSAGE adopts GraphSAGE's embeddings algorithm with an updated mean aggregator function that leverages multiple neighborhood weight matrices for every unique ordered tuple and a unique matrix to describe the single node by type. The mean aggregator of features from the neighbors of node *v* via edges of type *r* is formulated as follows [20]:

$$h_{N_r(v)}^k = \frac{1}{|N_r(v)| \sum_{u \in N_r(v)} D_p[h_u^{k-1}]}. \tag{5}$$

where h_v^k is output for node *v* at layer *k*; *r* is the edge type between nodes *v* and *u*; $N_r(v)$ is the neighbor nodes set *v* via edge type *r*; D_p is the drop out with probability *p* applied to its argument vector; and $h_u^{k-1}$is the embedding coming from the previous layer *k* of a specific edge type *r*.

When concatenation is set to false, its step (or the forward propagation) through layer *k* is defined as:

$$h_v^k = \sigma\left(W_{t_v,self}^k D_p[h_v^{k-1}] + W_{r,neigh}^k h_{N_r(v)}^k + b^k\right) \tag{6}$$

where $W_{t_v,self}^k$ is the weight matrix for the self-edge of node type t_v; and $W_{r,neigh}^k$ is the weight matrix for edges of type r.

4. Methodology

This paper proposes a methodology that, starting from historical data, constructs the Knowledge Graph of terrorist events by transforming raw data into a graph containing multiple types of nodes and edges (i.e., a heterogeneous graph). Then, through a Graph Neural Network, it trains a model that predicts the existence of a relationship between event and group nodes to identify the perpetrator group of a terrorist attack. The overall methodology is illustrated in Fig. 1 and consists of the following steps:

- KG creation through data source pre-processing, features selecting, and mapping rules defining;
- GNN training: A GNN is set up and trained through terrorist events in the constructed KG;
- Group prediction: The GNN is tested in terms of perpetrator group identification capacity.

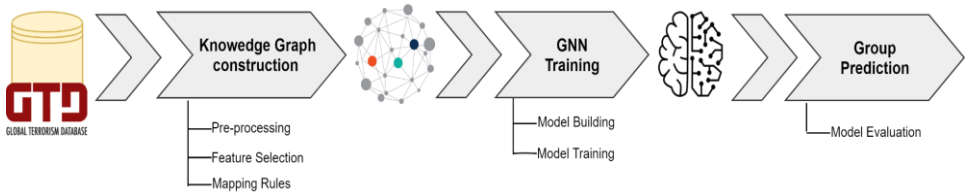

Fig. 1. Methodology overview.

4.1 Knowledge Graph Creation

In this phase, we first identify the best-representing features in GTD regarding the association between events and author groups. Then, pre-processing and mapping sub-phases are accomplished to adapt row data to graph-like representation.

4.1.1 Feature selection

The feature selection phase aims to obtain a score for each GTD feature to extract the K most representatives. We performed the scoring through the chi-squared statistic, a popular feature selection method suitable for categorical data.

Firstly, starting from all the attributes present in the dataset, we filtered the ones corresponding to characteristics or composition of the perpetrator group, as is the fact we want to predict, and we do not have information about it. Then, another filter regarded sub-specifications of attributes because of the quantity of missing data. A total of 29 categorical features are used to identify the most relevant ones of GTD. Fig. 2 shows a bar chart with the resulting 25 best-representative features and their importance scores. As described in the subsequent phase, among those features, we select the first 20 as the most involved in groups-events relationships description.

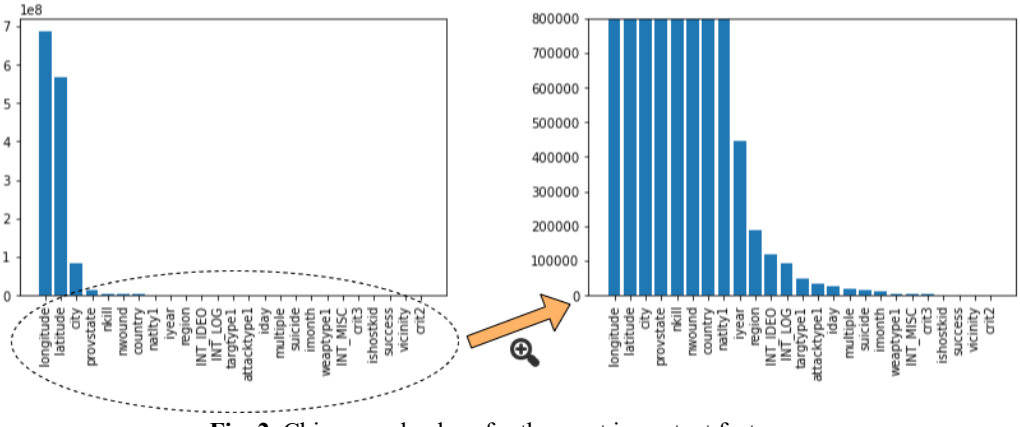

Fig. 2. Chi-squared values for the most important features.

4.1.2 Feature mapping

A pre-processing phase deals with transforming raw data in a suitable format for subsequent adoptions. Firstly, quantities of killed and injured people were split into three levels, as follows:

- Low: For killed in the range (1–3) and injured in (0–5).
- Medium: For killed in the range (3–10) and injured in (5–30).
- High: For killed people greater than 10 and injured greater than 30.

Features used to create the KG are shown in Table 1, with a brief description and how they contribute to making the KG.

Mapping between row data and graph elements works as follows. Firstly, any object, person, place, or event in the dataset is mapped as a node; edges define relationships between nodes. Second, selected features of the input dataset were transformed into a KG structure, describing them as property of nodes or new nodes.

Defining a new node for features such as region, month, weapon type, etc., gives us a chance to construct a context around an event that better inherits all its characteristics. In particular, the adopted combination of nodes and properties shown in Table 1 was empirically determined.

Table 1. Contribution of GTD attributes to the creation of the KG

Attribute	Description	Contribution to KG
gname	Perpetrator group name	Group node
iyear	Year of the event	Property + "Time feature" for Event
imonth/iday	Month/day of the event (1–12)/(1–31)	Month node + "Time feature" for Event
Region/country/state/city	Region/country/state/city where the event took place	Specific node
Latitude/ longitude	Specific latitude/longitude where the event took place	Property of Event
attackType	Type of attack	Attack type node
weaponType	Type of weapon used in the attack	Weapon node
target	Target of the attack	Target node
nkill/nwound	Quantity of killed/injured people	Kills node
natlty	Target nationality	Nationality node
suicide	The attack is of type suicide	Property of Event
multiple	The attack is part of multiple attacks	Property of Event
INT_LOG	The attack is international or domestic	Property of Event
INT_MISC	The attack is miscellaneous international or domestic	Property of the node Event
INT_IDEO	The attack is ideologically international or domestic	Property of Event

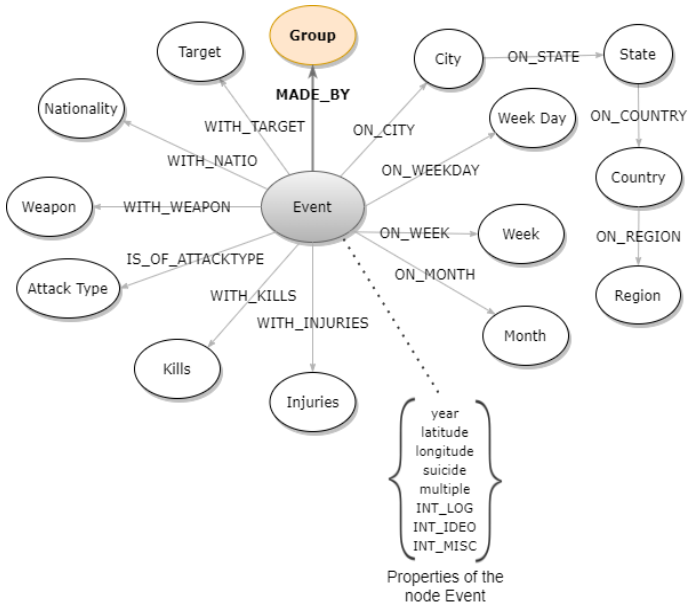

Fig. 3. Knowledge graph structure for a terrorist event.

As expressed in the table, some generated nodes do not directly depend on dataset attributes; specific mapping rules are established. In particular, rules were defined based on the nature of each selected feature, as detailed following.

1) Time features: The date of the attack, composed of the year, month, and day is mapped as follows:
   - A Month node;
   - A Week Day node with values in the range (1–7) that specifies which day of the week the attack was accomplished.
   - A Week node with values in the range (0–52) that specifies the week of the year of the attack.

2) Location features: The geographical location of the terrorist attack is represented by the following attributes (from the most specific to the most general): latitude, longitude, city, state, country, and region. The following rules were defined to generate a hierarchy of the attack position inside our KG:

- The latitude and longitude location of the attack will be properties of the event node.
- Information about the state, country and region are extracted and mapped to a new node.

Once all nodes are created, relationships among them generate edges of the KG. In particular, the relation between Event and Group nodes is defined as "MADE_BY" and is the edge we want to predict. Fig. 3 shows the resulting KG structure.

4.2 GNN Training

In this phase, the constructed KG is adopted to create and train the GNN aiming to identify the perpetrator group of a given attack. In particular, the following steps are accomplished:

1) Training and test datasets are defined by splitting edges (in the KG) between nodes of type Event and Group (that associate the perpetrator group to an attack).
2) Building the machine learning model based on the HinSAGE framework described previously, whose a link prediction layer is added for predicting the relationship between the event and group node. Given the node and a group, if the relationship exists, the output is 1; 0, otherwise.
3) Training the machine learning model through the training set.

4.3 Group Prediction

The second phase of the proposed methodology adopts the trained Machine Learning model to identify the unknown perpetrator groups of terrorist attacks in the test dataset. Practically, the GNN extracts the corresponding embedding of the target node and each group; then, the link prediction layer establishes whether the relationship exists by associating a probability to case 1 (i.e., the association exists) and case 0 (i.e., the association does not exist). Finally, by empirically establishing a threshold (e.g., 0.6), it is possible to select, based on the probability, if the relationship between group and event exists.

An example of framework application on real data is shown in Fig. 4.

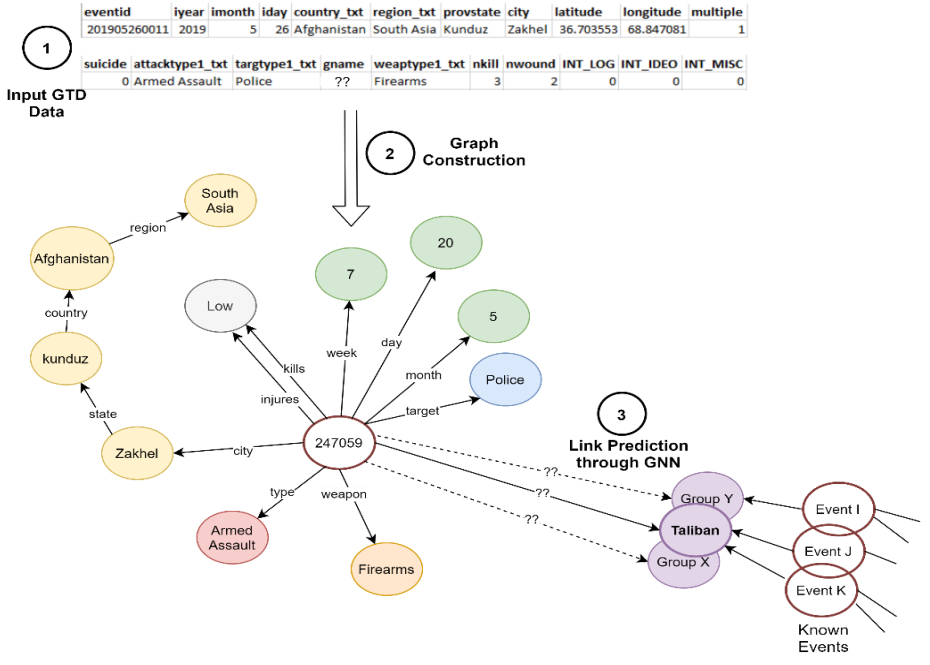

Fig. 4. Practical example on real data.

5. Experimentation

The real implementation of the proposed methodology, written in Python language, exploits Neo4j to create the KG, and StellarGraph library to build, train and test the GNN model. The following sections give some implementation details.

5.1 Knowledge Graph Creation

Neo4j was the tool adopted to create the KG. By employing the Cypher language, queries are executed to generate nodes, their properties, and relationships. The KG representation in Neo4j allows visualization of relationships and query results: a more natural way of interpreting the available information that enables understanding of relationships among events.

5.2 GNN Training

The GNN has been implemented through the StellarGraph Python library. In particular, it is based on the HinSAGE framework and consists of two HinSAGE layers (i.e., K=2), which means neighbor information is aggregated from a two-hop neighborhood. For the aggregation function, we use the mean function, which finds the element-wise mean of the edge embeddings in the sampled neighborhood. The last layer is the link prediction that takes a pair of node embeddings produced by HinSAGE, applies a binary operator to them to make the corresponding link embedding ("ip" for the inner product), and passes it through a dense layer.

The KG, mapped into a StellarGraph object, is split into train and test sets. It ensures all nodes are kept in both training and test sets, but the edges are randomly sampled. Then, the train set is adopted for training purposes.

5.3 GNN Training

An evaluation of model performance is conducted through the test dataset containing a sampled set of edges between Group and Event nodes. In this sense, we empirically set the threshold to 0.6. It is the cut-off to discriminate between network outputs: 0, when the output probability is lesser than 0.6, means no link presence; 1, when the output probability is higher or equal to 0.6, implies the presence of a link. In other words, if the prediction output of the model is greater than the threshold, we assume the existence of the relationship "MADE_BY" and generate a relationship between the perpetrator group and the terrorist event.

5.4 Performance Measures

Considering the link prediction problem as a binary classification issue, we use the following measurement to assess performance.

The area under the ROC curve (AUC) measures the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The ROC curve is a plot of true positive rate (TPR, or sensitivity) versus the false positive rate (FPR, or (1–Specificity)) at different probability cut-offs. Sensitivity is on Y-axis, and (1–Specificity) is on X-axis.

Cut-off represents the minimum threshold to identify the predicted probability as "event" (i.e., outcome equals 1, or, equivalently, the existence of a relationship between the Group and Event nodes, in our model). To generate the ROC curve, we calculate Sensitivity and (1–Specificity) at all possible cut-offs and then plot them. Finally, the AUC is evaluated as follows:

$$\frac{(FPR_{i+1} - FPR_i) * (TPR_i + TPR_{i+1})}{2}$$

The closer the AUC is to 1, the better the performance of the model.

Other adopted measures are accuracy, precision, recall, and F-measure, defined as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - Measure = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

where TP is true positive, FP is false positive, TN is true negative, and FN is false negative.

5.5 Results

Conducted experimentation consisted of two different tests. The first focused on Group nodes having more than five "MADE_BY" relationships (i.e., who made more than five attacks); the second considered the entire KG.

Table 2 details the results of both tests. As the tables show, the generated learning model has a good discriminating capacity for perpetrator group identification. Moreover, as demonstrated by the second test, adding more information to the KG (i.e., adopting the total KG) does not negatively impact on overall performances of the GNN.

Table 2. Performances summary

Measure	First test (i.e., on the filtered KG)	Second test (i.e., on the full KG)
AUC	0.93	0.96
Accuracy	0.89	0.89
F-measure	0.82	0.89
Recall	0.81	0.91
Precision	0.83	0.88

Table 3. Comparison with existing approaches

Classification technique	Accuracy (%)
Random forest [7]	87.8
Decision tree [7]	84.3
Gradient boosting [7]	84.1
C4.5 Decision tree [6]	60.0
Random forest [6]	58.5
SVM [6]	73.2
GNN HinSAGE (filtered KG)	89.5
GNN HinSAGE (full KG)	89.4

5.6 Comparison with Existing Approaches

Performance results of both tests demonstrate a substantial improvement concerning state of the art. In particular, as shown in Table 3, our performances overcome more consolidated ML techniques, including

decision trees, random forests, SVM, and gradient-boosting [6, 7]. Such promising results could be attributable to two characteristics of the proposed framework. On the one hand, the high prediction accuracy recently registered by GNNs in many research areas. On the other hand, choices regarding the representation of attack context through the graph allow the model to learn from neighbors and better identify patterns in data.

6. Conclusion and Future Work

Nowadays, counter-terrorism is a critical issue for police, policymakers, and defense departments. Many research and enterprise efforts are applied by mainly exploiting available data and the new technologies in AI and decision support systems. The goal is to give new tools to fight against terrorist attacks. In this direction, this work aims to identify the perpetrator group of a terrorist attack based on its characteristics; so, extracting and exploiting inherited information to help decision-makers.

Experimentation results demonstrate the suitability of GNNs for this type of task because of their capacity to represent the reference KG and extract inherited knowledge about the event and its neighborhood. In particular, the produced learning model demonstrates to predict, with a good performance, the existence or not of the link “MADE_BY” that relates the terrorist event with the perpetrator group. First, experimentation was done with a subset of events considering only perpetrators made more than five attacks. Then, the entire dataset with all events was used to train the model. Although the results show that using the whole dataset as input to the GNN did not worsen concerning the original approach, both learning models, with almost similar accuracy, improved the existing literature performance. In addition, representing data in a graph-based model gives a convenient visual tool for analyzing the relationships among different attacks.

In the future, it would be interesting to identify the improvement of the actual learning model by integrating data coming from unstructured data sources into the reference KG for augmenting entity representation. The objective is to provide valuable insights also to address the attack predictions. For example, defining a learning model that, by taking a new event and its characteristics as input, gives the experts instant knowledge about the nature of attacks and guides them about a likely imminent attack.

Acknowledgement

Not applicable.

Author’s Contributions

Conceptualization, MLB, GF, MG, VL, AP, AV. Investigation and methodology, MLB, GF, MG, VL, AP, AV. Resources, MLB, GF, MG, VL, AP, AV. Writing of the original draft, MLB, GF, MG, VL, AP, AV. Writing of the review and editing, MLB, GF, MG, VL, AP, AV. Data curation, MLB, GF, MG, VL, AP, AV. All the authors have read and approved the final manuscript.

Funding

None.

Competing Interests

The authors declare that they have no competing interests.

References

[1] Europol, *European Union Terrorism Situation and Trend Report 2021*. Luxembourg, European Union, 2021.

[2] S. Sudhakaran and O. Lanz, “Learning to detect violent videos using convolutional long short-term memory,” in *Proceedings of 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy 2017, pp. 1-6.

[3] B. Ionescu, M. Ghenescu, F. Rastoceanu, R. Roman, and M. Buric, “Artificial intelligence fights crime and terrorism at a new level,” *IEEE MultiMedia*, vol. 27, no. 2, pp. 55-61, 2020.

[4] P. Agarwal, M. Sharma, and S. Chandra, “Comparison of machine learning approaches in the prediction of terrorist attacks,” in *Proceedings of 2019 12th International Conference on Contemporary Computing (IC3)*, Noida, India, 2019, pp. 1-7.

[5] M. I. Uddin, N. Zada, F. Aziz, Y. Saeed, A. Zeb, S. A. Ali Shah, M. A. Al-Khasawneh, and M. Mahmoud, “Prediction of future terrorist activities using deep neural networks,” *Complexity*, vol. 2020, article no. 1373087, 2020. https://doi.org/10.1155/2020/1373087

[6] D. Talreja, J. Nagaraj, N. J. Varsha, and K. Mahesh, “Terrorism analytics: Learning to predict the perpetrator,” in *Proceedings of 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India, 2017, pp. 1723-1726.

[7] M. ALfatih, C. Li, and N. E. Saadalla, “Prediction of groups responsible for terrorism attack using tree based models,” in *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*, Wuhan, China, 2019, pp. 320-324.

[8] R. Mason, B. McInnis, and S. Dalal, “Machine learning for the automatic identification of terrorist incidents in worldwide news media,” in *Proceedings of 2012 IEEE International Conference on Intelligence and Security Informatics*, Washington, DC, 2012, pp. 84-89.

[9] D. Sinka, “Assessing the attractiveness of nuclear power plants as terrorist attack targets,” 2013 [Online]. Available: https://arhiv.djs.si/proc/nene2013/pdf/NENE2013_1501.pdf.

[10] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, “Text level graph neural network for text classification,” 2019 [Online]. Available: https://arxiv.org/abs/1910.02356.

[11] Y. Sun, Y. Yu, and J. Han, “Ranking-based clustering of heterogeneous information networks with star network schema,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 2019, pp. 797-806.

[12] F. Saracco, M. J. Straka, R. Di Clemente, A. Gabrielli, G. Caldarelli, and T. Squartini, “Inferring monopartite projections of bipartite networks: an entropy-based approach,” *New Journal of Physics*, vol. 19, no. 5, article no. 053022, 2017. https://doi.org/10.1088/1367-2630/aa6b38

[13] H. Kou, H. Liu, Y. Duan, W. Gong, Y. Xu, X. Xu, and L. Qi, “Building trust/distrust relationships on signed social service network through privacy-aware link prediction process,” *Applied Soft Computing*, vol. 100, article no. 106942, 2021. https://doi.org/10.1016/j.asoc.2020.106942

[14] H. Liu, H. Kou, C. Yan, and L. Qi, “Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph,” *Complexity*, vol. 2020, article no. 2085638, 2020. https://doi.org/10.1155/2020/2085638

[15] X. Zhou, W. Liang, K. I. K. Wang, R. Huang, and Q. Jin, “Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data,” *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 1, pp. 246-257, 2021.

[16] B. Ganesan, H. Patel, and S. Mehta, “Explainable link prediction for privacy-preserving contact tracing,” 2020 [Online]. Available: https://arxiv.org/abs/2012.05516.

[17] M. Samizadeh and B. Minaei-Bidgoli, “Drug-target interaction prediction by metapath2vec node embedding in heterogeneous network of interactions,” *International Journal on Artificial Intelligence Tools*, vol. 29, no. 1, article no. 2050001, 2020. https://doi.org/10.1142/S0218213020500013

[18] G. Fenza, M. Gallo, V. Loia, and A. Volpe, “Cognitive name-face association through context-aware graph neural network,” *Neural Computing and Applications*, 2021. https://doi.org/10.1007/s00521-021-06617-z

[19] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 1025-1035, 2017.

[20] J. Oh, K. Cho, and J. Bruna, “Advancing graphsage with a data-driven node sampling,” 2019 [Online]. Available: https://arxiv.org/abs/1904.12935.