Network Anomaly Detection Using a Graph Neural Network

Patrice Kisanga[1], Isaac Woungang[1], Issa Traore[2], and Glaucio H. S. Carvalho[3]
[1]Department of Computer Science, Ryerson University, Toronto, ON., Canada.
[2]Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada.
[3]Department of Computer Science and Engineering, Brock University, St. Catharines, ON., Canada.
Email: pkisanga@ryerson.ca, iwoungan@ryerson.ca, itraore@ece.uvic.ca, gdecarvalho@brocku.ca

2023 International Conference on Computing, Networking and Communications (ICNC) | 978-1-6654-5719-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICNC57223.2023.10074111

Abstract—**Contrary to the many traditional network security approaches that focus on volume-based threats, the Activity and Event Network (AEN) is a new approach built on a graph model, which addresses both volumetric attacks and long-term threats that traditional security tools cannot deal with. The AEN graph structural foundation can serve as a basis to construct a graph to be used in Graph Neural Network (GNN) for anomaly and threat detection purposes. In this paper, an AEN-based supervised Graph Convolutional Network (GCN) model is proposed, then evaluated using two labelled datasets, namely, the distributed denial of service (DDoS) and the TOR-nonTOR datasets, yielding an accuracy score of 76% with the DDoS dataset and 88% with the TOR-nonTOR dataset, respectively.**

Index Terms—**Anomaly detection, intrusion prevention system, intrusion detection systems, Activity and Event Network (AEN), Graph neural network (GNN), datasets, Graph convolutional network (GCN)**

I. Introduction

Over the years, several network tools such as antivirus software, intrusion prevention systems (IPSs) and intrusion detection systems (IDSs), have been developed to detect anomalies in a network environment. However, the mechanisms sustaining these approaches have focused on everyday volumetric anomalies, but have failed in addressing silent long-term security breaches. These stealthy threats can remain undetectable for a long period of time, causing tremendous leak of information, including private ones. To address this challenge, the AEN model has been proposed [1] as a new graph framework approach for detecting not only traditional network attacks, but also long-term attacks. AEN can be used to perform a continuous real-time analysis of a network while capturing the silent network attacks that can be undetected using regular or conventional detection mechanisms [1].

Using the structural foundation acquired from the AEN graph, along with the AEN elements and constructs, an emphasis has been put on graph neural network (GNN) as a mean to detect the anomalies in the network. GNN is a type of deep learning-based method that works on graphs, modeling the graph nodes and edges using relationships [2]. A GNN can be described as a neural network applied on graphs, meant to accomplish various tasks after it is trained on a given dataset. In this paper, the graphs are represented with both structured and unstructured data, making GNN an important tool to model various real-time data such as data from network systems, text recognition systems, to name a few.

978-1-7281-7315-3/20/$31.00 © 2020 IEEE

This paper focuses on the design of a novel GNN model inherited from an AEN graph construct, which is meant to detect security anomalies in a network environment. The proposed approach consists of studying the AEN model to acquire a structural foundation on graphs, then apply this knowledge to construct a GNN model. To achieve this task, an analysis of the AEN graph and its elements is performed using the following steps:

1) Extract the snapshots to obtain a snapshot of the graph at a specific point in time.
2) Cluster the snapshots by dividing the graph into smaller groups (i.e. clusters).
3) Match the constructed clusters based on the number of common nodes or edges between the different clusters.
4) Measure the difference between two clusters (referred to as Graph Edit Distance (GED)).

To accomplish the task of accurately distinguishing between normal and abnormal traffic in a network system, a GCN is proposed as GNN message passing method unlike other methods that are available in the literature such as graph attention and message passing networks mechanisms [1].

The rest of the paper is structured as follows. Section 2 discusses some related work. Section 3 briefly summarizes the AEN graph elements and steps involved in its analysis. Section 4 describes the proposed GCN model. Section 5 presents a preliminary experimental evaluation of the proposed detection scheme. Section 6 concludes the paper.

II. Related Work

In the recent years, GNNs [2] have emerged an in interesting extension to neural networks because of their ability to represent both structured and unstructured data in graphical form. Few studies of GNN models for anomaly detection purpose in network environment have been reported in [3]- [12].

In [3], Shchur et al. proposed four GNN model based on a semi-supervised node classification method (i.e. a form of graph mining technique), namely GCN, Mixture Model Network (MoNet), GraphSage and Graph Attention Network (GAT). Using the same parameter selections to experiment all four architectures, the following accuracy and average rank scores were obtained, with 1 being the best performance and 10 being the worst one: GCN (99.4 and 2.3), MoNet (99.0 and 2.7), GraphSage (98.3 and 2.7), and GAT(95.9 and 3.6), showing that GCN yielded the best result.

In [4], Pujol-Perich et al. proposed a GNN model called hots-connection graphs that captures not only individual flow features, but also their relationships. They reported that the proposed model can detect and prevent future attacks. The performance of their GNN model was compared against that of some advanced machine learning techniques using the CIC-IDS2027 dataset, showing superior results in terms of accuracy.

In [5], Atkinson et al.proposed a GNN-based auto-encoder scheme that can detect anomalies in boosted QCD jets. A decoder that instantly reconstructs the multidimensional edges and node features is also proposed using the Inner Product Layers concept. The proposed decoder is equipped with an edge-reconstruction network that is capable of learning the graph structures by reconstructing the entire graph.

In [6], Jianguo et al. proposed a GCN-based model that includes both the features extraction and the nodes' relationship for the purpose of anomaly detection. The proposed GCN model is tested against few machine learning-based models such as logistic regression and random forest models, showing its superiority in terms of accuracy.

In [7], Cai et al. focused on node feature extraction to obtain graph embedding in static graphs. A GNN model called Structural Temporal GNN (StrGNN) is proposed, which can handle both static and dynamic graphs. Their experiments showed that StrGNN can detect anomalous edges in dynamic graphs.

In [8], Chaudhary et al. proposed a GNN-based model to detect anomalies in network environments. The proposed model is based on statistical graph properties such as Betweenness centrality, Degree centrality and Closeness centrality, to define the properties of suspicious nodes, in view of detecting them. Experiments are conducted to validate the effectiveness of the proposed technique, showing promising accuracy level.

In [9], Wu et al. talk about the importance of the Industrial Internet of Things (IIoT) in the digital transformation of traditional industries towards an industry where services, data analysis and many devices are connected to the internet. Such industries, referred to as Industry 4.0 have experienced emergency due to IIoT. The authors explain that using GNN in smart transportation, smart energy, and smart factory to detect anomalies has helped improve the success and productivity of industries, since GNN can process data that is non-Euclidean, such as data in various industry settings.

In [10], Lo et al. propose a new security detection solution. Their new intrusion detection mechanism, called E-GraphSAGE, is based on Graph Neural Networks (GNNs). This new approach makes if possible to extract both the graph edge features and the network's topological patterns in Internet of Things (IoT) environments. After conducting several experiments on multiple Network Intrusion Detection System (NIDS) datasets, the authors demonstrate that E-GraphSAGE outperforms other advanced intrusion detection methods.

In [11], Scarselli et al. explain that many relationships in various disciplines, whether it is molecular chemistry, patter recognition, natural language processing, or data mining, to name a few, can be modeled using graphs. The authors propose a supervised neural network model that they call graph neural network (GNN) that works on both graph-level and node-level classification problems. The new model can process multiple types of graphs, including directed, undirected, cyclic and acyclic graphs, and uses a supervised learning algorithm, which performance is evaluated based on a set of given training examples.

In [12], Zhang et al. introduce a new graph neural network model called HetGNN that can effectively handle heterogeneous structural graph information as well as heterogeneous attributes of nodes. Multiple experiments on four datasets show that HetGNN scores better than many other advanced graph mining techniques in tasks, such as node classification, link prediction and node clustering, to name a few.

In this paper, a supervised GNN model based on GCN is proposed for network anomalies detection, and its effectiveness is validated through experiments, yielding promising results in terms of accuracy score using two example datasets.

III. AEN GRAPH

The AEN model as a new approach that focuses on long-term attacks [1]. As a new graph framework, it can be utilized to captured the events that occur continuously in a network, as well as the dynamicity and uncertainty of the activity of the network [1]. A partial view of an instance of a generated AEN graph from a subset of the ISOT CID dataset [13] containing various formats of files such as syscall logs, captures of TCP packet, and others, is captured in Fig. 1.

An AEN graph has the following characteristics [1]:

TABLE I: Types and distributions of nodes in the AEN graph for the ISOT CID dataset [15].

Node label	Number of elements
ACCOUNT	606
ALERT	12672
DOMAIN	252
HOST	252
IP	252
IP_RANGE	195
ORGANIZATION	91
LOCATION	81
Total	**14401**

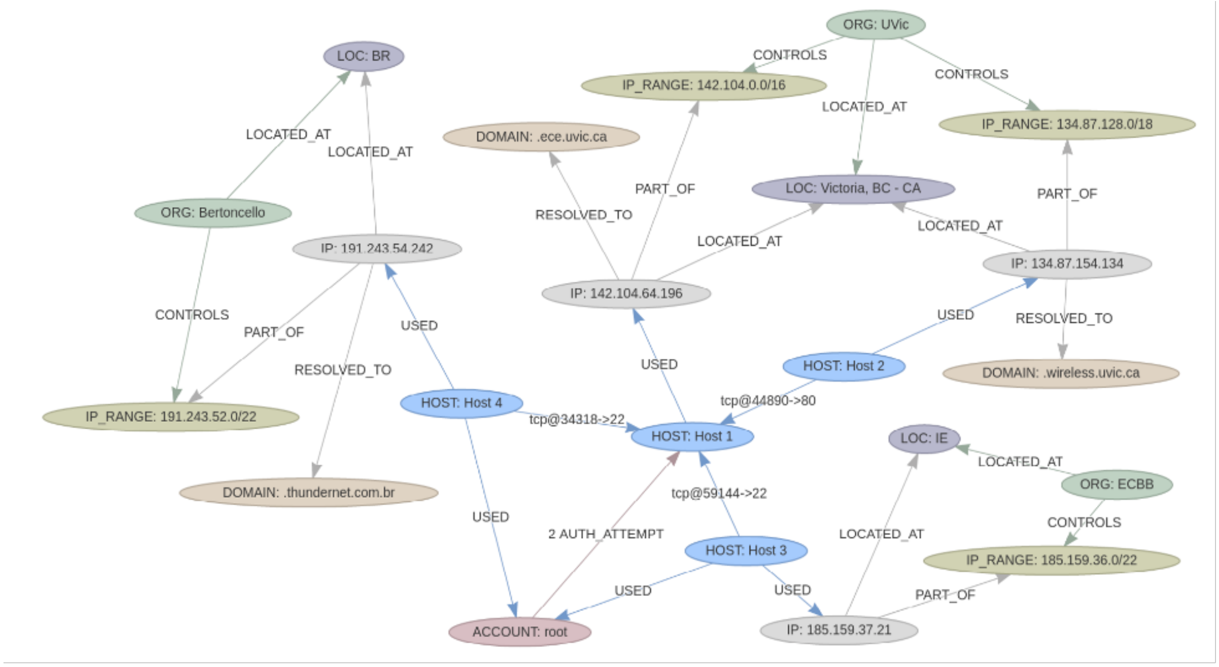
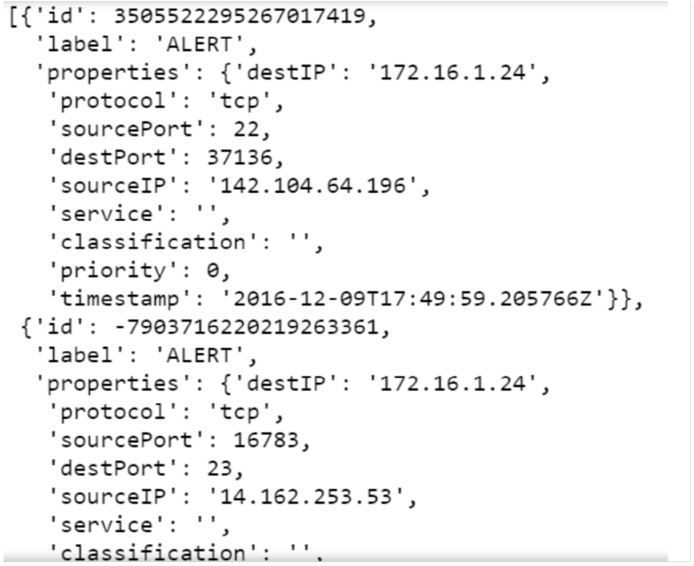


Fig. 1: Snapshot of a AEN graph based on a subset of the ISOC CID dataset [15].



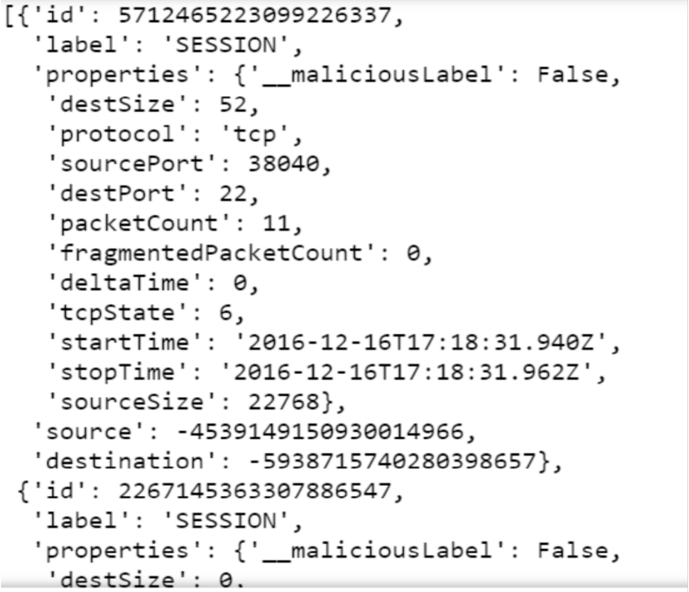
```
[{'id': 3505522295267017419,
  'label': 'ALERT',
  'properties': {'destIP': '172.16.1.24',
   'protocol': 'tcp',
   'sourcePort': 22,
   'destPort': 37136,
   'sourceIP': '142.104.64.196',
   'service': '',
   'classification': '',
   'priority': 0,
   'timestamp': '2016-12-09T17:49:59.205766Z'}},
 {'id': -7903716220219263361,
  'label': 'ALERT',
  'properties': {'destIP': '172.16.1.24',
   'protocol': 'tcp',
   'sourcePort': 16783,
   'destPort': 23,
   'sourceIP': '14.162.253.53',
   'service': '',
   'classification': '',
```

Fig. 2: Snapshot of node's elements [15]

- Every node element has an ID, a label representing its type, and its associated properties fields as depicted in Fig. 2. Nodes can be active or passive and have several links (edges) between them, each with a direction. For the ISOT CID dataset [13], the types of nodes and their distributions are shown in Table I.
- Similarly, every edge element in the AEN graph has an ID, a label and associated properties' fields. An example is shown in Fig. 3 for the ISOT CID dataset [13], where the data for two edges are shown, with "SESSION" label or type assigned to these edges and the associated properties. For this dataset, the types and distributions of the edges in the AEN graph are shown in Table II.

Typically, constructing an AEN graph requires that the features be extracted from the data sources, which can be either internal, meaning that the source relies within the company perimeters, or external, meaning that the source of data is outside the company perimeters, for instance, the data are originated from firewall logs, anti-virus logs, network traffic logs, to name a few.

After generating the AEN graph model, it can be analyzed using the following steps:



```
[{'id': 5712465223099226337,
  'label': 'SESSION',
  'properties': {'__maliciousLabel': False,
   'destSize': 52,
   'protocol': 'tcp',
   'sourcePort': 38040,
   'destPort': 22,
   'packetCount': 11,
   'fragmentedPacketCount': 0,
   'deltaTime': 0,
   'tcpState': 6,
   'startTime': '2016-12-16T17:18:31.940Z',
   'stopTime': '2016-12-16T17:18:31.962Z',
   'sourceSize': 22768},
  'source': -4539149150930014966,
  'destination': -5938715740280398657},
 {'id': 2267145363307886547,
  'label': 'SESSION',
  'properties': {'__maliciousLabel': False,
   'destSize': 0.
```

Fig. 3: Snapshot of edge's elements [15].

TABLE II: Types and distributions of edges in the AEN graph for the ISOT CID dataset [15].

Edge label	Number of elements
AUTH_ATTEMPT	262258
SESSION	115451
ALERT_TRG_BY_HOST	12672
HOST_USED_IP	252
IP_RES_DOM	252
IP_LOCATED_AT	252
IP_PART_RANGE	252
ORG_CTRL_RANGE	196
ORG_LOC_AT	119
Total	**391704**

- Snapshot extraction: this consists of obtaining a complete visualization of the data at a specific time. This helps to focus on nodes or edges that are present during the time of extraction.
- Community or cluster detection: this consists in clustering the graph nodes into smaller groups (referred to as clusters), in such a way that the number of relationships between each group or cluster is relatively small compared to the number of edges between different clusters [3].
- Clusters matching: this refers to matching the clusters based on the number of similar nodes or edges between them.
- Graph Edit Distance: this refers to measuring the difference between two clusters. GED can be described as the cost of changing one graph (G1) to another (G2) [4]

IV. Proposed GNN Model

The proposed GNN model is based on GCN. It consists of:

- Two layers (1 input and 1 output). The input layer takes the node features and embeddings, and the output layers produces a probability, where 1 represents an anomaly and 0 represents a normal network behavior.
- 16 hidden layers are considered.
- ReLU: this represents the rectified linear activation function, a non-linear function that outputs 0 if the input is negative, otherwise it outputs the input value. Compared to other activation functions such as the sigmoid or hyperbolic tangent functions, ReLU can accelerate the training speed of the model [14].
- Log_softmax: this is a mathematical function that computes the logarithm of the softmax function that takes a vector of real numbers and returns a probability.

The GNN model is trained for a number of epochs using the following parameters:

- Optimizer: this refers to the mechanism that the model uses to update itself based on the training data. The optimizer helps to improve the model performance. In our case, the optimizer called Adam is considered.
- Linear regression (lr): this model uses 1e-1 as the linear regression to predict the outcome.
- Loss: the loss function determines how the model measures its performance based on the training data. The proposed model uses nll_loss as loss function.

Based on aforementioned GNN model, our proposed algorithm is shown in Algorithm 1.

Algorithm 1 Graph Neural Network

```
Phase 1: Preprocess the dataset
for each dataset in .csv format do
    Create dataframe from dataset
    if dataset not in .csv format then
        Convert dataset to .csv format
        Create dataframe from dataset
    end if
    Build graph from dataframe
    for each dataframe do
        if label equals 0 or 1 then
            Retrieve label
        else if label does not equal 0 or 1 then
            Rename label
            Retrieve label
        end if
        Create edge index from graph coordinates
        Select embeddings
        Normalize embeddings
    end for
end for
Phase 2: Train and Test the model
for each processed dataset do
    Split data into train, validate and test masks
    Train model
    Evaluate model
end for
```

V. Performance Evaluation

In our experiments two datasets are considered respectively. The DDoS dataset consists of data generated from a DDoS attack and it contains both normal and abnormal traffic. On the other hand, the Tor-nonTor dataset consists of read-world traffic composed of non-Tor or benign traffic, such as emails, chats, audio-streaming, video-streaming, and browsing data, to name a few.

As performance metric, we consider the accuracy score from the confusion matrix, obtained as:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

where TP denotes true positive, TN denotes true negative, FP denotes false positive, and FN denotes false negative values. For our results, we have obtained an accuracy score of 76% when using the DDoS dataset and 88% accuracy score when using the Tor-nonTor dataset.

VI. Conclusion

This paper has proposed a new a supervised GCN-based model with two layers, based on the AEN paradigm, for efficient anomaly/threat detection in network environment, capable of addressing volumetric attacks and long-term threats that traditional security tools cannot detect. Preliminary experimental results using two labelled datasets, namely, the DDoS and TOR-nonTOR datasets, have shown an accuracy score of 76% with the DDoS dataset and 88% with the TOR-nonTOR dataset, respectively. Future works include comparing the performance of the proposed detection model against that of few benchmark models, and testing it using other meaningful available datasets.

References

[1] Traore I., Quinan P.G., and W. Yousef, ”The activity and event network (AEN) model: Graph elements and construction”, Technical Report, ECE-2020-01, Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Jan. 2020.

[2] Karagiannakos S., ”Graph Neural Networks : An overview”, https://towardsdatascience.com/graph-neural-networks-an-overview-dfd363b6ef87 (Last visited Sept. 1, 2022)

[3] Shchur O., Mumme M., Bojchevski A., Gunnemann S. , ”Pitfalls of Graph Neural Network Evaluation”, In arXiv:1811.05868, https://doi.org/10.48550/arXiv.1811.05868 (Last visited Sept. 1, 2022)

[4] Pujol-Perich D., Suárez-Varela J., Cabellos-Aparicio A., Barlet-Ros P., ”Unveiling the potential of Graph Neural Networks for robust Intrusion Detection”, In *arxiv*, Vol. arXiv.2107.14756, https://doi.org/10.48550/arXiv.2107.14756 (Last visited Sept. 9, 2022)

[5] Atkinson O., Bhardwaj A., Englert C., Ngairangbam V.S., Spannowsky M., ”Anomaly detection with Convolutional Graph Neural Networks”, In arXiv:2105.07988, JHEP 08 (2021) 080, https://doi.org/10.1007/JHEP0828202129080 (Last visited Sept. 9, 2022)

[6] Jianguo J., Jiuming C., Tianbo G., Kim-Kwang R.C, Chao L., Min Y., ”Anomaly Detection with Graph Convolutional Networks for Insider Threat and Fraud Detection”, IEEE Military Communications Conference (MILCOM), 2019, pp. 109-114, doi: 10.1109/MILCOM47813.2019.9020760.

[7] Cai L., Chen Z., Luo C., Gui J., Ni J., Li D., Chen H. ”Structural Temporal Graph Neural Networks for Anomaly Detection in Dynamic Graphs”, In Proc. of the 30th ACM International Conference on Information Knowledge Management (CIKM), Oct. 30, 2021, https://doi.org/10.1145/3459637.3481955, pp.3747-3756.

[8] Chaudhary A., Mittal H., Arora A., ”Anomaly Detection using Graph Neural Networks,” In Proc. of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 346-350, doi: 10.1109/COMITCon.2019.8862186.

[9] Wu Y., Dai H-N, Tang H., ”Graph Neural Networks for Anomaly Detection in Industrial Internet of Things,” IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9214-9231, June 15, 2022, doi: 10.1109/JIOT.2021.3094295.

[10] Lo W., Siamak L., Mohanad S., Marcus G., Marius P., ”E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT”, NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, 2022, ,doi: 10.1109/NOMS54207.2022.9789878 (Last visited Sept. 9, 2022)

[11] Scarselli F., Gori M., Tsoi A., Hagenbuchner M., Monfardini G., ”The Graph Neural Network Model”, IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80, Jan. 2009, doi: 10.1109/TNN.2008.2005605 (Last visited Sept. 9, 2022)

[12] Zhang C., Song D., Huang C., Swami A., Chawla N., ”Heterogeneous Graph Neural Network”, Association for Computing Machinery, 2019, https://doi.org/10.1145/3292500.3330961 (last visited Sept. 9, 2022)

[13] Aldribi A., Traore I., B. Moa. ”Data Sources and Datasets for Cloud Intrusion Detection Modeling and Evaluation”. Mishra B., Das H., Dehuri S., Jagadev A. Cloud Computing for Optimization: Foundations, Applications, and Challenges. Studies in Big Data. (39): 333-366. Springer, 2018.

[14] ReLu, https://deepai.org/machine-learning-glossary-and-terms/relu (Last visited Sept. 9, 2022)

[15] Quinan P. G., Traore I., Gondhi U R., and Woungang I., ”Unsupervised Anomaly Detection using a New Knowledge Graph Model for Network Activity and Events”, In: Renault E., Boumerdassi S., Mühlethaler P. (eds) Machine Learning for Networking. MLN 2021. Lecture Notes in Computer Science, vol 13175. Springer, Cham. https://doi.org/10.1007/978-3-030-98978-1-8