GLASS: A Graph Learning Approach for Software Defined Network Based Smart Grid DDoS Security

Keerthiraj Nagaraj
Department of Electrical and Computer Engineering,
University of Florida,
Gainesville, USA
k.nagaraj@ufl.edu

Allen Starke
Department of Electrical and Computer Engineering,
University of Florida,
Gainesville, USA
allen1.starke@ufl.edu

Janise McNair
Department of Electrical and Computer Engineering,
University of Florida,
Gainesville, USA
mcnair@ece.ufl.edu

***Abstract*—In recent years, smart grid communications (SGC) has evolved to use new technologies not only for data delivery but also for enhanced smart grid (SG) security and reliability. Software Defined Networks (SDN) has proved to be a reliable and efficient architecture for handling diverse communication systems due to their ability to divide responsibilities of the network using control plane and data plane. This paper presents a graph learning approach for detecting and identifying Distributed Denial of Service (DDoS) attacks in SDN-SGC systems (GLASS). GLASS is a two phase framework that (1) detects if SDN-SGC is under DDoS attack using supervised graph deep learning and then (2) identifies the compromised entities using unsupervised learning methods. Network performance statistics are used for modeling SDN-SGC graphs, which train Graph Convolutional Neural Networks (GCN) to extract latent representations caused by DDoS attacks. Finally, spectral clustering is used to identify compromised entities. The experimental results, obtained by analysis of an IEEE 118-bus system, show the average throughput for compromised entities is able to maintain 84% of normal traffic level with GLASS, compared to achieving only 4% of normal throughput caused by DDoS attacks on compromised entities without the GLASS framework.**

I. Introduction

The traditional power grid system is no longer a viable option due to increasing user demand, upsurge in availability of renewable energy resources, outdated infrastructure, and the need for increased security and reliability. The Smart Grid (SG) constitutes advanced technologies such as Supervisory Control and Data Acquisition (SCADA), Advanced Metering Infrastructure (AMI), Plug-In Electric Vehicles (PEVs) and Renewable Energy Resources (RERs), and offers a variety of new features such as demand side management, a two-way communication framework and real time pricing [1]. These additional features and technologies in SG also create new challenges that have to be addressed. RERs are expected to become a major source of power in the SG by the year 2050. These renewable energy resources are collected using different sources/technologies, are often geographically distributed, disparate in nature and operation, and have lower generation capacity in comparison to the conventional energy resources. To efficiently couple the conventional with the renewable energy resources, there is a need for timely and reliable SG communication [2].

In addition, with new communication technology, there are various kinds of cyber attacks that may occur such as Man-In-The-Middle attacks, Jamming, Black hole attacks, Energy Drain attacks, Sinkhole attacks, and Denial of Service attacks. A successful cyber-attack on SG in the Ukraine during December of 2016 caused a blackout which impacted 225,000 customers [3]. Many cyber attacks also have the potential of leading SG into total collapse [4]. A resilient SG should have the ability to detect, identify and react to sudden system failures through natural causes or intentional attacks on the network. For instance, during Denial of Service attack on SG, a Phasor Measurement Unit (PMU) or multiple PMUs might get compromised which will result in the loss of measurement data from the relevant part of the grid. In such cases, SG will lose the system observability of all the associated substations in the grid, which might result in temporary power outages or blackouts.

Software Defined Networks (SDN) has proved to be a reliable and efficient architecture for handling diverse communication systems. With its ability to separate data plane from the control plane, SDN can provide control and management for various SG entities such as utility, smart meters, and AMI, and play an instrumental role in evolving SG to integrate new services, resources, technologies and needs as the SG evolves. In this paper, we propose a graph learning approach for an SDN based SG security (GLASS) framework. GLASS adopts a two phase strategy for detecting and identifying Distributed Denial of Service (DDoS) attacks. To our knowledge, this is the first study to use supervised and unsupervised graph learning approaches to detect and identify cyber attacks in a SDN-SGC system. This paper specifically makes the following contributions:

- Uses graph convolutional neural networks (GCN) to detect DDoS attacks for various attack scenarios;
- Uses spectral clustering to identify DDoS compromised entities; and
- Analyzes the impact of the detection, identification and mitigation of DDoS attacks on network performance (throughput, transmission delay).

The remainder of the paper is organized as follows. Section II provides background information on SDN-SGC architecture and graph learning approaches. Section III provides details about the proposed GLASS framework for DDoS detection and identification. Results of a case study are discussed in Section IV. Finally, Section V concludes the paper by summarizing results and providing future research directions for the GLASS framework.

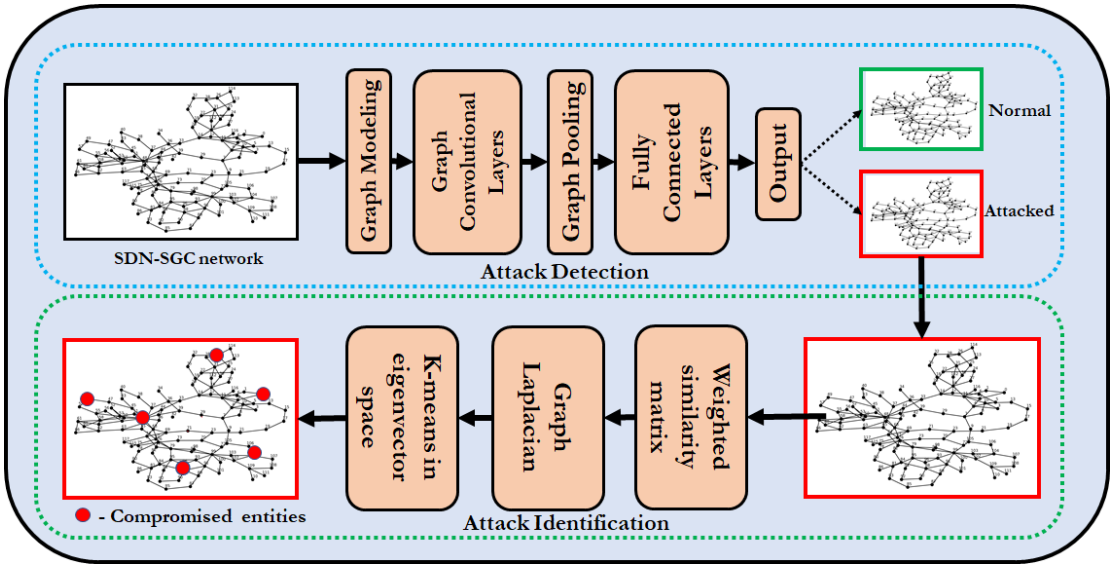

Fig. 1. System architecture for GLASS framework showing attack detection and identification steps

II. Background Information

Traditionally, SG communications consisted of very low rate exchange of serial data, and point-to-point data exchange between deterministic nodes. The incorporation of Phasor Measurement Units (PMUs) and dynamic networking infrastructure has created heterogeneous network structures that simultaneously experience different types of events, requiring more complex networking and control protocols. SG subsystems are local agents, composed of Distributed Energy Storage Systems (DESS), such as flywheels and grid-connected batteries, a Synchronous Generator, a PMU and a Distributed State Estimator (DSE). The PMUs in the smart grid communicate data to other PMUs to coordinate and analyze energy performance measures using reliable networking protocols such as Modbus RTU over TCP/IP.

As mentioned previously, SDN is a networking paradigm in which the forwarding hardware is decoupled from control decisions. The network intelligence is logically centralized in software-based controllers (the control plane), and the network devices become simple packet forwarding devices (the data plane) that can be programmed via an open interface. SDNs help to assemble new services and infrastructure quickly to meet dynamically changing environment objectives. Furthermore, the software implementation of the control plane and the built-in data collection mechanisms are excellent tools to implement additional analysis layers for network control. Extracting knowledge from collected data to understand and predict the state of the SG network will be crucial to implement security management in the SG. Our system will ingest network traffic performance statistics and monitor the data for anomalies/attacks.

A. DDoS attacks in SDN-SGC

This work focuses on developing methods for detection and mitigation of the Distributed Denial-of-Service (DDoS) cyber attacks discussed in [5], [6]. During Denial-of-Service (DoS) attacks, the attacker intentionally disrupts the transmission of data to/from a given node through an excessive amount of service requests to the victim node, consuming all available resources. The impact factor of DoS attacks are high. This type of attack increases network traffic at its victim node (i.e. arrival rate) to consume the victim's resources and extend queue length resulting in an increase in wait times or transmission delays as can be seen in [7]. This can cause nodes to shutdown, and negatively effect the entire network as a whole. Distributed Denial of Service (DDoS) has an even larger impact, since the attack occurs from multiple nodes, resulting in a higher arrival/attack rate. In the attack model considered, a PMU subsystem may be attacked in order to hinder or disconnect it from the rest of the network. If a PMU subsystem is attacked, the slowed/halted communication of data could introduce substantial errors into the SG system.

B. Supervised Learning using GCN

Supervised learning deals with the class of machine learning problems in which we have a labelled dataset guiding the model on what decisions to make while it is being trained. In these problems, learning, or the adaptation of the model, is supervised by the desired response. Traditional supervised deep learning approaches such as Multilayer Perceptron Neural Networks, Convolutional Neural Networks, etc cannot be directly applied on graph based data as they fail to extract latent representations from non-euclidean data generated by complex relationships and interdependence between various entities in graphs [8]. Recently, many deep learning approaches have been extended for graph based data, resulting in Graph Neural Networks, Graph Convolutional Neural Networks, and Graph Auto-encoders [9]. The Graph Convolutional Neural Networks (GCN) [10] concept was developed using concepts of convolutional neural networks and Graph Neural Networks. GCN is useful for solving real world problems such as link prediction, node classification and graph classification. GCN aims to learn the hidden layer repre-

sentations that encode features of nodes, local graph structures or even entire graphs.

C. Unsupervised Learning using Spectral Clustering

Unsupervised learning deals with the class of machine learning problems in which we do not have a labelled dataset to train a model. Instead we rely on capturing undetected patterns in the data through techniques like modeling probability densities, extracting embeddings, calculating distance measures etc [11] . One of the commonly used applications of unsupervised learning is clustering. One of the most popular clustering algorithms is Spectral Clustering, since it can be solved using simple linear algebra libraries and often produces better results than traditional clustering algorithms such as K-means algorithm [12]. Spectral clustering is useful for identifying "groups of nodes" which show similar behavior in a graph. Spectral clustering relies on weighted adjacency matrix of the graph to be indicative of different behaviors exhibited by different groups of nodes, for example, victim (compromised) nodes and normal nodes in the network.

III. GLASS FRAMEWORK

In this section, we discuss the design of GLASS framework.

A. Graph modeling

We model SDN-SGC as a weighted undirected Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W}_\mathcal{V}, \mathcal{W}_\mathcal{E})$ with $\mathcal{N}$ number of PMUs in the SG considered as the node set $\mathcal{V}$ and the connections between them considered as the edge set $\mathcal{E}$. The quality of communication between neighboring PMUs is quantified by various network performance metrics such as transmission delay, throughput etc. These performance metrics are used to form node attribute matrix $\mathcal{W}_\mathcal{V} \in \mathbb{R}^{\mathcal{N} \times \mathbf{f}_\mathcal{N}}$ and edge attribute matrix $\mathcal{W}_\mathcal{E} \in \mathbb{R}^{|\mathcal{E}| \times \mathbf{f}_\mathcal{E}}$, where $\mathbf{f}_\mathcal{V}$ and $\mathbf{f}_\mathcal{E}$ represent number of node attributes and edge attributes respectively.

We divide the entire length of simulation into $\mathcal{T}$ number of time steps and model SDN-SGC in each of these time steps as $\mathcal{G}_\mathbf{t}$ where $\mathbf{t} \in 1, 2, 3..., \mathcal{T}$. $\mathcal{G}^\mathbf{t}$ varies in each time step as $\mathcal{W}_\mathcal{V}$ and $\mathcal{W}_\mathcal{E}$ changes over time depending on the communication between PMUs in the SG. At time step $\mathbf{t}$, we use the transmission delay between neighboring PMUs to form $\mathcal{W}^\mathbf{t}_\mathcal{E}$ and the average transmission delay of a PMU with all of its neighboring PMUs to form $\mathcal{W}^\mathbf{t}_\mathcal{V}$. Note that $\mathcal{W}_\mathcal{V}$ and $\mathcal{W}_\mathcal{E}$ should be designed using information from the network that is representative of the problem at hand.

Let $\mathcal{W}^\mathbf{t}_\mathbf{adj} \in \mathbb{R}^{N \times N}$ be the weighted adjacency matrix of $\mathcal{G}^\mathbf{t}$ formed such that w_{ij} is edge weight between PMUs i and j if they are connected, else $w_{ij} = 0$. Let $\mathcal{W}^\mathbf{t}_\mathbf{deg} = \mathbf{diag}(d_1, d_2, ..., d_\mathcal{N})$ be weighted degree diagonal matrix such that $d_i = \sum_{j=1}^{\mathcal{N}} w_{ij}$. Let $\mathcal{Y} \in \mathbb{R}^{\mathcal{T} \times 1}$ denote a true label vector such that $\mathcal{Y}^\mathbf{t} = 1$ if SDN-SGC network is under DDoS attack in time step $\mathbf{t}$, else $\mathcal{Y}^\mathbf{t} = 0$.

B. DDoS detection

In the DDoS detection phase of our framework, we make use of the concepts of neural networks on graphs to train models that are guided to detect any intentional/unintentional anomalous behavior in SDN-SGC network. The objective of this phase is to detect whether a SDN-SGC network modeled by $\mathcal{G}^\mathbf{t}$ is under any DDoS attacks. We develop our DDoS detection models based on the concepts of graph convolutional layers and graph pooling proposed in [10], [13], [14]. As shown in Fig.1, the graph information modeled from SDN-SGC network is passed through multiple graph convolution layers, followed by a single graph pooling layer which is followed by multiple fully connected multilayer perceptron neural network layers and finally by a sigmoid layer to generate the decision of whether a given instance of SDN-SGC network modeled by $\mathcal{G}_\mathbf{t}$ is under attack or not.

1) Graph convolution (GCN) layers: Our implementation of graph convolutional layers is based on the implementation of GCN proposed in [10]. For a graph $\mathcal{G}_\mathbf{t}$ with weighted adjacency matrix $\mathcal{W}^\mathbf{t}_\mathbf{adj}$ and node attribute matrix $\mathcal{W}^\mathbf{t}_\mathcal{V}$, each graph convolutional layer outputs a matrix containing hidden node level representations termed as a hidden feature matrix $\mathcal{H}$, which is defined as

$$\mathcal{H} = \mathbf{f}(\mathcal{Z} \times \mathcal{W}^\mathbf{t}_\mathcal{V} \times \Theta) \tag{1}$$

where Θ is the matrix of learnable parameters for a given graph convolutional layer, $\mathbf{f}(.)$ is an activation function, and $\mathcal{Z}$ is defined as

$$\mathcal{Z} = \mathbf{I}_\mathcal{N} + (\widetilde{\mathcal{W}}^\mathbf{t}_\mathbf{deg})^{\frac{-1}{2}} \times \widetilde{\mathcal{W}}^\mathbf{t}_\mathbf{adj} \times (\widetilde{\mathcal{W}}^\mathbf{t}_\mathbf{deg})^{\frac{-1}{2}}. \tag{2}$$

The weighted adjacency matrix, $\widetilde{\mathcal{W}}^\mathbf{t}_\mathbf{adj}$, is defined as

$$\widetilde{\mathcal{W}}^\mathbf{t}_\mathbf{adj} = \mathbf{I}_\mathcal{N} + \mathcal{W}^\mathbf{t}_\mathbf{adj} \tag{3}$$

where $\mathbf{I}_\mathcal{N} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ is an identity matrix and $\widetilde{\mathcal{W}}^\mathbf{t}_\mathbf{deg}$ is modified weighted degree diagonal matrix calculated in a similar fashion as $\mathcal{W}^\mathbf{t}_\mathbf{deg}$, except using weights in $\widetilde{\mathcal{W}}^\mathbf{t}_\mathbf{adj}$ instead of $\mathcal{W}^\mathbf{t}_\mathbf{adj}$.

2) Graph Pooling: The hidden feature matrix $\mathcal{H}$ contains node level representations but we are interested in the graph classification task which requires graph level representation features. Graph pooling layers are useful in summarizing information from multiple $\mathcal{H}$ matrices which contains multiple representations for each node into one feature representation for the graph $\mathcal{G}_\mathbf{t}$. In our framework, we use mean pooling, a type of graph pooling layer to obtain graph level representation $\mathbf{g}^\mathbf{t}$ of graph $\mathcal{G}_\mathbf{t}$ by considering mean values of all of its node level representations.

3) Dense Neural Network layers: The graph level representation feature obtained from graph pooling layer is passed through multiple dense (fully connected) neural network layers with learnable weights succeeded by a sigmoid layer which completes the DDoS detection model. The output of sigmoid layer is used to predict whether $\mathcal{G}_\mathbf{t}$ is under any DDoS attacks.

The pseudocode of the DDoS detection phase of the proposed GLASS framework is presented in Procedure 1. Let graphs belonging to $\mathcal{T}_{train}$ time steps be used for training and remaining $\mathcal{T}_{test}$ graphs for testing of DDoS attack detection model. Let the model prediction vector for test data be $\mathcal{Y}_\mathbf{p} \in \mathbb{R}^{\mathcal{T}_{test} \times 1}$, where ${\mathcal{Y}_\mathbf{p}}^\mathbf{t} = 1$ if $\mathcal{G}_\mathbf{t}$ is predicted to be under any DDoS attacks, else ${\mathcal{Y}_\mathbf{p}}^\mathbf{t} = 0$.

Procedure 1 DDoS detection

SDN-SGC network $\rightarrow \mathcal{G}$ (Section III.A).

Train DDoS detection model:

Input: $\mathcal{G}$, $\mathcal{W}_{\mathbf{adj}}$, $\mathcal{Y}$

for $\mathcal{G}_{\mathbf{t}}$ where $t = 1 : \mathcal{T}_{train}$ **do**
    Calculate $\mathcal{H}$ using Eq. 1, 2 and 3 for each GCN layer.
    Obtain $\mathbf{g^t}$ by combining $\mathcal{H}$ from multiple GCN layers using graph pooling.
    Pass $\mathbf{g^t}$ to fully connected layers and sigmoid layer.
    Obtain error by comparing sigmoid layer output to $\mathcal{Y}^{\mathbf{t}}$
    Tune model parameter through back-propagating error
end for

Output: DDoS detection model with learned parameters.

Testing DDoS detection model:

Input: $\mathcal{G}$, $\mathcal{W}_{\mathbf{adj}}$

for $\mathcal{G}_{\mathbf{t}}$ where $t = 1 : \mathcal{T}_{test}$ **do**
    Obtain sigmoid layer output $\mathcal{Y}^{\mathbf{t}}_{sigmoid}$ through learned model parameters
    if $\mathcal{Y}^{\mathbf{t}}_{sigmoid} > \mathbf{0.5}$ **then**
        $\mathcal{Y}_{\mathbf{p}}{}^{\mathbf{t}} = 1$
    else
        $\mathcal{Y}_{\mathbf{p}}{}^{\mathbf{t}} = 0$
    end if
end for

Output: $\mathcal{Y}_{\mathbf{p}}$

C. DDoS identification

For the DDoS identification phase, we use spectral clustering to identify compromised PMUs in $\mathcal{G}^{\mathbf{t}}$ if $\mathcal{Y}_{\mathbf{p}}{}^{\mathbf{t}} = 1$. The intuition behind this idea is that only if SDN-SGC network is under an attack, then various PMUs (compromised and normal) exhibit different structural patterns in the corresponding parts of the network, and these underlying graph based patterns can be used to separate compromised PMUs from normal (not compromised) PMUs. It is crucial to notice that unless $\mathcal{Y}_{\mathbf{p}}{}^{\mathbf{t}} = 1$ for $\mathcal{G}^{\mathbf{t}}$, unsupervised learning techniques will not be confidently able to separate compromised PMUs from normal PMUs at time step $\mathbf{t}$. Hence, DDoS identification phase is always preceded by DDoS detection step outlined in Procedure 1 if $\mathcal{Y}_{\mathbf{p}}{}^{\mathbf{t}} = 1$.

The main tools used in spectral clustering are Graph Laplacian ($\mathcal{L}$) matrices, which are developed based on spectral graph theory. Normalized Graph Laplacian of a graph $\mathcal{G}^{\mathbf{t}}$ with weighted adjacency matrix $\mathcal{W}^{\mathbf{t}}_{\mathbf{adj}}$ and weighted degree diagonal matrix $\mathcal{W}^{\mathbf{t}}_{\mathbf{deg}}$ is calculated as

$$\mathcal{L}^{\mathbf{t}} = \mathbf{I}_{\mathcal{N}} - (\mathcal{W}^{\mathbf{t}}_{\mathbf{deg}})^{-1} \times \mathcal{W}^{\mathbf{t}}_{\mathbf{adj}} \tag{4}$$

where $\mathbf{I}_{\mathcal{N}} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is an identity matrix. Eigenvectors and Eigenvalues obtained through spectral decomposition of $\mathcal{L}^{\mathbf{t}}$ are used in conjunction with traditional k-means algorithm to divide compromised PMUs and normal PMUs in the SDN-SGC network. We classify all the PMUs in SG into 3 groups as following:

- **Primary victims** ($\mathcal{V}_{\mathbf{pv}}$) - Set of PMUs which are directly targeted during DDoS attacks.
- **Secondary victims** ($\mathcal{V}_{\mathbf{sv}}$) - Set of PMUs which are connected to primary victims (1-hop neighbors). These are of interest to us as DDoS attacks on primary victims have considerable impact on PMUs connected to them.
- **Normal PMUs** ($\mathcal{V}_{\mathbf{n}}$) - Set of PMUs which are neither primary victims nor secondary victims in the SG.

The pseudocode of the DDoS identification phase of the proposed GLASS framework for a time step $\mathbf{t}$ is presented in Procedure 2. The value of $\mathbf{k}$ is set to be three as we want to identify three groups of PMUs namely primary victims, secondary victims and normal PMUs through DDoS identification phase. After identifying three groups of PMUs, we calculate mean network performance using either average throughput ($\mu_{TH}[.]$) or average transmission delay ($\mu_{TD}[.]$) of each group and use their values to label all three groups. The intuition is that attacked PMUs would usually have higher performance deterioration than normal PMUs.

Procedure 2 DDoS identification

Input: $\mathcal{W}^{\mathbf{t}}_{\mathbf{adj}}$, $\mathcal{W}^{\mathbf{t}}_{\mathbf{deg}}$, $\mathcal{Y}_{\mathbf{p}}{}^{\mathbf{t}}$, $\mathbf{k} = 3$

if $\mathcal{Y}_{\mathbf{p}}{}^{\mathbf{t}} = 1$ **then**
    Calculate Graph Laplacian $\mathcal{L}^{\mathbf{t}}$ using Eq. 4
    Compute first $\mathbf{k}$ eigenvectors $u_1, ..., u_{\mathbf{k}}$ of $\mathcal{L}^{\mathbf{t}}$ corresponding to $\mathbf{k}$ of its smallest eigenvalues
    Form $\mathcal{U} \in \mathbb{R}^{\mathcal{N}\times\mathbf{k}}$ containing vectors $u_1, ..., u_{\mathbf{k}}$
    For $i = 1, 2, ..., \mathcal{N}$, let $a_i \in \mathbb{R}^{\mathbf{k}}$ be the vector corresponding to i^{th} row of $\mathcal{U}$
    Cluster vector points $(a_i)_{i=1,...,\mathcal{N}} \in \mathbb{R}^{\mathbf{k}}$ with $\mathbf{k}$-means algorithm into $\mathcal{V}^{\mathbf{t}}_{\mathbf{n}}$, $\mathcal{V}^{\mathbf{t}}_{\mathbf{pv}}$, $\mathcal{V}^{\mathbf{t}}_{\mathbf{sv}} \ni \big(\mu_{TH}[\mathcal{V}^{\mathbf{t}}_{\mathbf{n}}] > \mu_{TH}[\mathcal{V}^{\mathbf{t}}_{\mathbf{sv}}] > \mu_{TH}[\mathcal{V}^{\mathbf{t}}_{\mathbf{pv}}]\big)$ OR $\big(\mu_{TD}[\mathcal{V}^{\mathbf{t}}_{\mathbf{n}}] < \mu_{TD}[\mathcal{V}^{\mathbf{t}}_{\mathbf{sv}}] < \mu_{TD}[\mathcal{V}^{\mathbf{t}}_{\mathbf{pv}}]\big)$
else
    consider all PMUs $\mathcal{V} \in \mathcal{V}^{\mathbf{t}}_{\mathbf{n}}$
end if

Output: $\mathcal{V}^{\mathbf{t}}_{\mathbf{n}}$, $\mathcal{V}^{\mathbf{t}}_{\mathbf{pv}}$, $\mathcal{V}^{\mathbf{t}}_{\mathbf{sv}}$

IV. Results and Discussions

A. Implementation tools

Network performance statistics (i.e. transmission delays, and throughput) are generated using mininet and extensions to emulate the communication layer of IEEE 118-bus power grid system that commonly uses IEEE C37.118.2 or IEC 61850 over TCP/IP communication schemes. In this environment, anomalous network traffic (i.e. DoS attack) is generated using the tools, such as hping3 [7], to initiate TCP flooding attacks. In general, the packet arrival rate for the victim node is increased during the periods where DoS attacks take place. This action consumes network resources on the victim node resulting in an increase in transmission delays and decrease in throughput. Network performance metrics are recorded using network monitoring tools such as sflow-rt, t-shark, or wireshark and the open network operating system (ONOS) is used as the SDN controller. The implementation of the proposed GLASS framework and

TABLE I
GLASS FRAMEWORK RESULTS FOR IEEE 118-BUS BASED SDN-SGC NETWORK

Attack severity	$\lvert\mathcal{V}_{\mathbf{pv}}\rvert$	DDoS detection				DDoS identification	
		Accuracy	Precision	Recall	F1-score	TCR	ACR
		$\mu \pm \sigma_{sd}$	$\mu \pm \sigma_{sd}$	$\mu \pm \sigma_{sd}$	$\mu \pm \sigma_{sd}$	$\mu \pm \sigma_{sd}$	$\mu \pm \sigma_{sd}$
Low	1	97.01 ± 0.58	98.19 ± 0.56	95.80 ± 0.56	96.97 ± 0.60	99.86 ± 03.64	74.38 ± 39.55
	5	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	96.28 ± 08.21	100.0 ± 0.0
	10	99.98 ± 0.05	99.96 ± 0.09	100.0 ± 0.0	99.98 ± 0.49	88.61 ± 10.58	100.0 ± 0.0
High	1	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	99.80 ± 04.46	83.79 ± 35.20
	5	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
	10	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	99.59 ± 02.34

evaluation of results was conducted using Python libraries such as StellarGraph [15], NumPy, Pandas, SciPy, Matplotlib and Scikit-learn in the Anaconda environment.

B. Data description

In this data set, DDoS attacks are simulated for a SDN-SGC network. In a total of 3000 ($\mathcal{T}$) time steps (each time step constitutes a data sample which contains 118 data points, one for each bus) considered, DDoS attacks are introduced in 50% of the total samples, with each attack lasting for ten consecutive samples. The resultant data set (overall 354,000 data points) of this type has half the samples with DDoS attacks and another half with normal network performance samples. While training, we use same number of normal and attacked samples to train the GCN model to reduce class wise bias. In each of the attacked samples, a variation of (i.e. 1, 5, 10) buses were randomly chosen to be attacked. The network performance statistics generated for the communication layer were based on the M/M/c queue, i.e., $c \geq 1$, where packet arrivals were modelled after Poisson distribution and transmission delays were inherently modelled after the exponential distribution. The throughput is measured as the actual amount of information (i.e. network packets) that can flow through a connection at a given time, and transmission delay is measured as the time taken for packet to be transmitted between two PMUs.

C. Numerical results

The SDN-SGC network considered in this case study is based on IEEE 118-bus system. It contains 118 nodes (1 for each bus) and 186 network links (1 for each branch), with connections between them as considered in [16]. The performance of the proposed DDoS detection phase of the GLASS framework is evaluated using popular classification metrics [17] such as Accuracy, Precision, Recall and F1-score, while we define two metrics namely True Compromised Ratio (TCR) and All Compromised Ratio (ACR) for evaluating performance of DDoS identification phase of GLASS framework as the former popular metrics doesn't provide complete picture for the performance of an unsupervised learning algorithm such as spectral clustering. True Compromised Ratio (TCR) is calculated as the percentage ratio of compromised nodes identified correctly in $\mathcal{G}^{\mathbf{t}}$. All Compromised Ratio (ACR) is calculated as the percentage ratio of identified nodes truly belonging to the set of true compromised nodes in $\mathcal{G}^{\mathbf{t}}$.

The mean (μ) and standard deviation (σ_{sd}) values of accuracy, precision, recall and F1-score summarized for 10 different sets of training and testing data, and the mean (μ) and standard deviation (σ_{sd}) values of TCR and ACR summarized $\forall \mathcal{G}_{\mathbf{t}}$ where $t = 1 : \mathcal{T}_{test}$ are presented in Table I. $|\mathcal{V}_{\mathbf{pv}}|$ in Table I indicates number of PMUs directly targeted during DDoS attacks. A total of 3000 ($\mathcal{T}$) graphs (354,000 data points) were used for implementing steps mentioned in Procedures 1 and 2. The node attribute matrix is formed using average transmission delay associated with PMUs in the network. The GCN model trained during the detection phase is formed by 2 graph convolution layers each with 64 units and dropout of 0.2 to reduce the chance of overfitting, 2 fully connected layers with 32 and 16 units, and a sigmoid layer with all the layers containing ReLu activation function. The model is trained for 50 epochs with repeated stratified k-fold (5 folds, 2 repeats) technique. We have tested our framework for a variety of attack scenarios by considering low and high attack severity levels, and the number of directly attacked PMUs (primary victims) varying among 1,5 and 10. The values in Table I show that the proposed GLASS framework detects whether a given SDN-SGC network is under any DDoS attacks with high accuracy, precision, recall and F1-score, and also correctly identifies the compromised PMUs with high TCR and ACR values for all the attack scenarios.

D. Impact of attacks and mitigation on network performance

Network traffic/performance statistics collected by third party monitoring tools are stored in a database (InfluxDB, AWS, etc.) for analysis and/or extracted by a separate controller [18] tasked with hosting network security applications, like the proposed GLASS framework. Mitigation is applied by sending updated flow policies to the main SDN controller's northbound interface for reconfiguring the flow tables in switches. The controller limits the flow of TCP SYN packets to the compromised nodes, which improves the network performance considerably [19].

Figures 2 and 3 shows the variation of network performance in terms of average throughput and average transmission delay

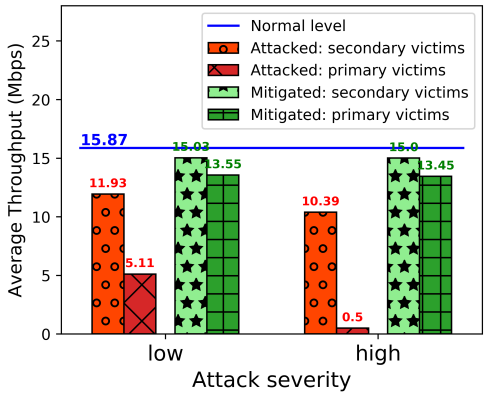

Fig. 2. Average throughput due to DDoS attacks and mitigation

for normal traffic to the nodes, for compromised nodes during DDoS attacks and after mitigation is applied to the compromised nodes, as identified by GLASS framework. The average throughput decreases drastically (4% of the normal traffic level) when subjected to DDoS attacks, but is improved (84% of the normal traffic level) after the mitigation policy is applied to the compromised nodes identified by the GLASS framework. Similarly, average transmission delay increases drastically (1415% of normal traffic delay) during DDoS attacks but is decreased closer to pre-attack levels (121% of normal traffic level) through the mitigation policy assisted by GLASS framework.

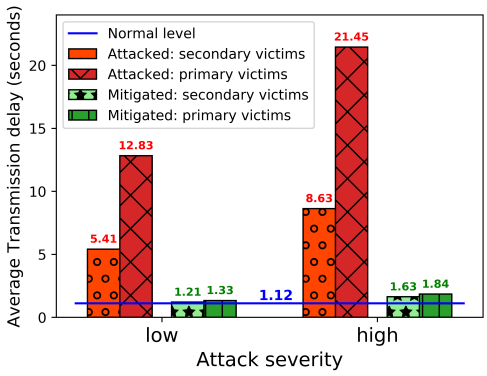

Fig. 3. Average transmission delay due to DDoS attacks and mitigation

V. Conclusions

This paper presents GLASS: A graph learning approach to detect and identify DDoS attacks in SDN-SGC system. The proposed framework works in two phases namely, DDoS detection and DDoS identification. DDoS detection employs graph convolutional networks designed for graph level classification tasks to detect whether a given SDN-SGC network is under any DDoS attacks. If DDoS attacks are detected from the first phase, DDoS identification phase which employs spectral clustering is then used to identify compromised entities in the network so that mitigation policies can be applied to improve the network performance degraded due to DDoS attacks. Experimental results show that the use of proposed GLASS framework resulted in improvement of network performance statistics such as average throughput and average transmission delay. The GLASS framework learns latent representations in the network caused due to introduction of DDoS attacks through supervised graph based deep learning and unsupervised spectral graph clustering approaches. In future work, the GLASS framework will be extended to learn latent representations in the network from unintentional factors like system failures (component malfunction) or intentional cyber attacks, such as False Data Injections, Jamming, and Man-In-The-Middle attack etc.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1809739.

References

[1] M. H. Rehmani, A. Davy, B. Jennings, and C. Assi, "Software defined networks-based smart grid communication: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2637–2670, 2019.

[2] F. Yu, P. Zhang, W. Xiao, and P. Choudhury, "Communication systems for grid integration of renewable energy resources," *Network, IEEE*, vol. 25, pp. 22 – 29, 11 2011.

[3] D. E. Whitehead, K. Owens, D. Gammel, and J. Smith, "Ukraine cyber-induced power outage: Analysis and practical mitigation strategies," in *2017 70th Annual Conference for Protective Relay Engineers (CPRE)*, 2017, pp. 1–8.

[4] S. Goel and Y. Hong, *"Security Challenges in Smart Grid Implementation," in Smart Grid Security*. Springer, London, 2015.

[5] M. H. Cintuglu, O. A. Mohammed, K. Akkaya, and A. S. Uluagac, "A survey on smart grid cyber-physical system testbeds," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 446–464, 2017.

[6] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[7] J. Gao, S. Chai, B. Zhang, and Y. Xia, "Research about dos attack against icps," *Sensors*, vol. 19, p. 1542, 03 2019.

[8] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *ArXiv*, vol. abs/1812.04202, 2018.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–21, 2020. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2020.2978386

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

[11] Rui Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[12] U. von Luxburg, "A tutorial on spectral clustering," *CoRR*, vol. abs/0711.0189, 2007. [Online]. Available: http://arxiv.org/abs/0711.0189

[13] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *AAAI*, 2018.

[14] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," *CoRR*, vol. abs/1902.06673, 2019. [Online]. Available: http://arxiv.org/abs/1902.06673

[15] C. Data61, "Stellargraph machine learning library," https://github.com/stellargraph/stellargraph, 2018.

[16] P. Chopade and M. Bikdash, "Modeling for survivability of smart power grid when subject to severe emergencies and vulnerability," in *2012 Proceedings of IEEE Southeastcon*, 2012, pp. 1–6.

[17] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *In Data Mining and Knowledge Discovery Handbook*, 2010, pp. 667–685.

[18] A. Starke, J. McNair, R. Trevizan, A. Bretas, J. Peeples, and A. Zare, "Toward resilient smart grid communications using distributed sdn with ml-based anomaly detection," in *IFIP Conference on Wired/Wireless Internet Communications*, vol. 1. IFIP, 2018, pp. 1–12.

[19] A. Starke, Z. Nie, M. Hodges, C. Baker, and J. McNair, "Denial of service detection mitigation scheme using responsive autonomic virtual networks (ravn)," in *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, 2019, pp. 1–6.