Understanding the Influence of Graph Kernels on Deep Learning Architecture: A Case Study of Flow-based Network Attack Detection

Liya Su*†, Yepeng Yao*†‡, Zhigang Lu*†, Baoxu Liu*†
†*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*
**Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
{suliya, yaoyepeng, luzhigang, liubaoxu}@iie.ac.cn

***Abstract*—Flow-based network attack detection technology is able to identify many threats in network traffic. Existing techniques have several drawbacks: i) rule-based approaches are vulnerable because it needs all the signatures defined for the possible attacks, ii) anomaly-based approaches are not efficient because it is easy to find ways to launch attacks that bypass detection, and iii) both rule-based and anomaly-based approaches heavily rely on domain knowledge of networked system and cyber security. The major challenge to existing methods is to understand novel attack scenarios and design a model to detect novel and more serious attacks.**

In this paper, we investigate network attacks and unveil the key activities and the relationships between these activities. For that reason, we propose methods to understand the network security practices using theoretic concepts such as graph kernels. In addition, we integrate graph kernels over deep learning architecture to exploit the relationship expressiveness among network flows and combine ability of deep neural networks (DNNs) with deep architectures to learn hidden representations, based on the communication representation graph of each network flow in a specific time interval, then the flow-based network attack detection can be done effectively by measuring the similarity between the graphs to two flows. The proposed study provides the effectiveness to obtain insights about network attacks and detect network attacks. Using two real-world datasets which contain several new types of network attacks, we achieve significant improvements in accuracies over existing network attack detection tasks.

***Index Terms*—Network attack detection, graph kernel, deep learning architecture**

I. Introduction

With the advent of the Internet, networked systems and applications get increasingly more complex than ever before and become an important part of our society in almost all areas. In today's cyberspace, most cyber-criminals utilize network resources to conduct their malicious activities. Moreover, recent advanced attacks tend to involve multiple hosts to conceal malicious behaviors of real attackers by using cross-host and multi-step attack methods. For instance, distributed link-flooding attacks are effective DDoS attacks that deplete the bandwidth of certain network links by using multiple bots with real IP addresses to direct low-intensity flows to multiple targets [1]. On the one hand, an attacker may make use of a large number of entities for an attack. On the other hand, sizable networks may yield too many probable attack targets [2]. The battle against these kinds of network attacks is becoming more difficult, since they send low-intensity, individual flows that are identical and unable to distingushed from legitimate flows. Therefore, improving the design of current flow-based network attack detection models to make them more suitable for providing flexible network attack detection is urgent.

Cross-host network attacks have been the major kind of network attack in nowadays. These attacks that are the main threats for security over the Internet have caused special attention. However, investigating and detecting network attacks across multiple hosts is still challenging. Traditionally, the network security community has focused on measuring network traffic information and detecting attacks by using a number of statistical techniques. At the same time, network communication flows can be manipulated to impact the quality of services or conceal malicious activities that can compromise network security. Unfortunately, existing flow-based network attack detection methods are inadequate to figure out the relationship and impact of cross-host attacks.

In recent years, deep learning and graph kernels are two emerging learning techniques that are able to facilitate the efficiency of network attack detection approaches for securing networks. Deep learning can represent a network attack detection task, e.g., generating embedding for a network flow, as a neural network whose parameters can be trained end-to-end, so that it depends upon as limited cyber security domain knowledge as possible. Graph kernels can capture and compare structural information from network communication representation graphs effectively. In order to take most the advantages of both techniques, there is a bunch of research work devoting to developing attack detection methods leveraging deep learning and graph kernels. However, rare effort is made to enhance the design of current detectors to make full use of those two advanced techniques.

Recent research has shown that these requirements are not adequately tackled in previous researches. **Requirements.** We briefly describe two of the requirements for network attack detection that we will address in this work. **First,** the requirement that we can immediately notice is to capture and compare structural information. Just considering individual

‡Corresponding author: yaoyepeng@iie.ac.cn

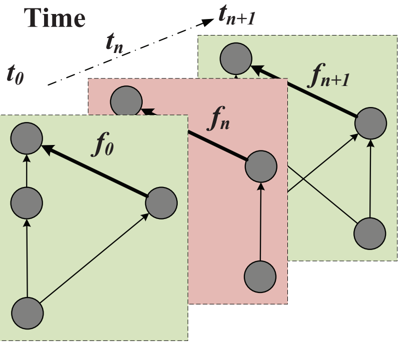


Fig. 1. Network flow data with structural information and contextual information.

nodes or edges with their attributes separately is not enough. Instead of learning representations for just the nodes of the graph, like [3], our work focus on learning a feature representation for the structural information of subgraphs. The above mentioned work can not be applied directly to our work. **Second,** only considering the structural information without the context is insufficient to distinguish whether a network flow is triggered with or without other attack flows, as shown in Fig. 1. It is meaningful for the detection approach to capture and compare the contextual information to make the attack detection approach more accurate.

Although current graph kernels could meet the first requirement finely, the second requirement which is more domain-specific and challenging. Some recent work study deep learning on graphs to capture the contextual information as well as structural information. However there are still many unexplored issues in conducting deep learning to handle various directed attributed graphs like network communication graphs. Taking network communication graphs as inputs to a deep neural network requires represent graphs as feature vectors in a way preserving the communication relationships and flow contextual information.

Inspired by the advantages of deep learning and graph kernels, in this paper, to research on recent network attack trends and perform efficient flow-based network attack detection, we investigate several new types of network attacks and propose the integrated use of deep learning and graph kernels to understand the network attacks. Our goal is to make a step towards accurate attack detection through both structural and contextual information that overcomes the inaccuracy of current methods. Further, we propose an integrated framework to achieve network attack detection.

Contributions. Facing the aforementioned requirements, in this paper, we propose and develop a novel network attack detection framework. The main contributions of this paper are as follows:

- We provide an in-depth understanding that graph kernels offer an elegant way to capture the heterogeneity and similarity of network attacks and enable the detection design at the deep learning model.
- We design a novel framework to detect malicious flows and improve the accuracy of network attack detection process, by integrating graph kernels and deep learning into an end-to-end architecture.
- We perform an in-depth evaluation to evaluate the efficacy of network attack detection. After evaluating our framework on two real-world datasets, we demonstrate that the proposed framework effectively outperforms the traditional machine learning methods and baseline deep neural network methods in terms of accuracy.

The rest of the paper is organized as follows. Section II presents the preliminaries of this work and describes several case studies. The proposed attack detection framework is discussed in Section III followed by the empirical evaluation in Section IV. A summary of some previous work related to this paper are provided in Section V. At last, Section VI concludes this paper.

II. Preliminaries and Case Studies

Since this work is directed towards understanding the influence of graph kernels on deep learning by taking several novel network attacks in flow-based data as use cases, we first describe the problem and analyze new trend of attacks in more detail. Then, we elaborate on a study on some network attack cases discovered in our research. Further, we analyze the techniques the attacker employ in the network attacks from the network communication perspectives.

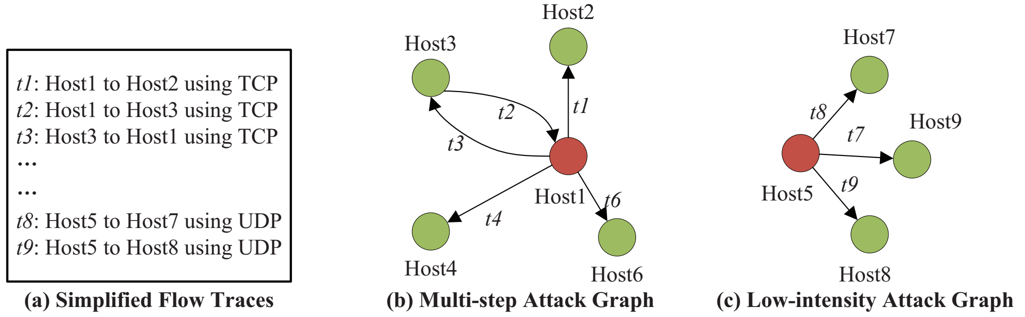
A. Problem Definition

In particular, we focus on capturing structural information of flow-based network traffic data and address the issue of calculating structural similarity between network flows.

The key point in the problem is a proper definition of flow context. This relies on directional flow-based data. Flows describe statistical information on communications between hosts and typically include *Source IP, Source Port, Destination IP, Destination Port, Protocol, Bytes, Packets, TCP-Flags, Timestamp and Duration* as flow meta features. Given a set of network flows, existing methods have the ability to detect several kinds of malicious network flows, but some kinds of attacks can be evaded by modifying several flow meta features. Indeed, different structural information in different communication graphs are correlated. For example, when the flows come from the similar sources and go towards the similar destinations.

Given a directed attributed graph $G < V, E >$, the purpose of the network attack detection is to relate a set of hosts to a set of vertices. As described in [1], an attack at time step t attacks a set of host $H_{attack}(t) \in H$, influencing the quality of a set of vertices $V_{attack}(t) \in V$. The goal of the attack detection is to find the malicious host set and detect the malicious communication edges until time step $t + \alpha$.

Assume that $p(h_t)$ expresses the probability of h acting as a malicious host at time step t, $p(f)$ is the probability of f being an attack flow at time step t, and $p(h, f_t)$ is the probability of h sending f at time step t. The goal is to seek



t1: Host1 to Host2 using TCP
t2: Host1 to Host3 using TCP
t3: Host3 to Host1 using TCP
...
...
t8: Host5 to Host7 using UDP
t9: Host5 to Host8 using UDP

(a) Simplified Flow Traces **(b) Multi-step Attack Graph** **(c) Low-intensity Attack Graph**

Fig. 2. Graph model of two types of network attacks. (a)simplified flow traces corresponding to network attacks; (b)communication graph construction for multi-step attacks; (c)communication graph construction for low-intensity attacks.

to minimize the total possibility of $p(h)$ over all time steps. In the neural networks model, system needs to extract the flow meta information about f and seek to minimize $p(f)$. However, attackers can do little changes to evade the single time step detection with small effort. While it is difficult to evade from the whole structural communication information which graph kernels have the ability to describe, that is, graph kernels can capture the structural similarity between $p(f_t)$ and $p(f_{t+1})$ to compare $p(h, f_t)$ and $p(h, f_{t+1})$.

For estimating the similarity between structural communication informations on pair of communication graphs, we use graph kernel functions. We construct a network communication graph for each traffic flow using a specific time interval $[t, t+\alpha]$.

B. Case Studies

In general, understanding the motivations and operations of several new network attack types plays a vital role in the challenge of addressing these threats. However, understanding how deep learning architecture work is challenging, so they are often treated as black boxes. As looking into the trained weights of a deep learning model may be a complicated way to understand which type of information is more important for the model, we decided to focus on the different communication graph model of network attacks. Fig. 2 shows graph models of two new types of network attacks.

1) Multi-step Attack: A multi-step attack differs from traditional one-off network attacks as it is launched in multiple steps with a single specific malicious objective inside the network, containing more than one distinct actions [4]. It involves different steps that may not be malicious when implemented separately, but all steps are necessary to complete successfully network attack, as shown in Fig. 2(b). For example, Mirai [5] begins its attack by scanning the whole Internet for devices that run interactive sessions, such as Telnet and SSH, then attempts to log in using default IoT passwords. Once successful, Mirai sends the scanned addresses and associated passwords to a collection server, which asynchronously triggers a loader to infect the device. Infected hosts scan for additional victims and accept attack commands from a C&C server.

The detection methods assume there are relationships among attack flows, which suppose to be the same attack scenario.

2) Low-intensity Attack: A low-intensity attack is launched with sending legitimate, low-intensity traffic flows towards attack targets. It could compute a large set of IP addresses whose advertised routes cross the same link, and then direct its bots to send low-intensity traffic to those addresses, as shown in Fig. 2(c). For example, link-flooding attacks [1] [6] seek to block the paths connecting to the target. The attacker first constructs a link-map around the target, then the attacker floods critical links by sending traffic to decoy servers. Finally, all paths from the target to the gateways are cut-off because of link congestion.

The goal of the detection methods is to keep the network running without any blocked links and to discover probable targets and attackers. Therefore, the flow meta information on the network should be detected and the incoming flows to different destinations should be analysis to expose probable attackers and their target.

III. Integrated Model of Graph Kernels and Deep Learning

In this section, we propose a framework to integrate graph kernels into deep learning architecture for improving flow-based network attack detection. Since we observe that existing graph kernels, such as shortest path kernel, capture the structural information well but fail to capture contextual information, we employ a LSTM fully convolutional networks on time axis. We give an overview of the architecture of the framework in Fig. 3.

A. Framework Architecture Overview

To take full advantage of structural features, convolutional neural network (CNN) could be adopted to capture adjacent relations among the network flows, along with employing recurrent neural network (RNN) on time axis.

The fully convolutional networks are effective learning models for time series analysis problems [7]. In our proposed framework, the fully convolutional block is blended by a basic LSTM block followed by dropout and the fully convolutional block consists of three stacked convolutional blocks. The

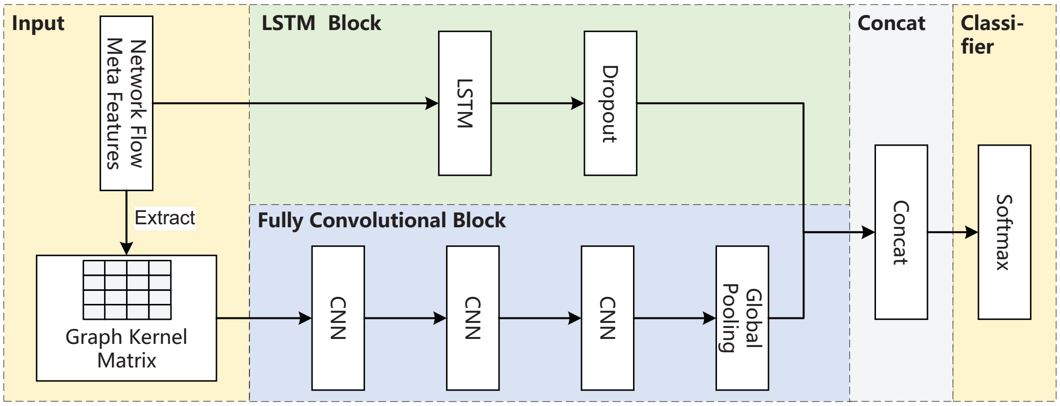


Fig. 3. Overview of the Network Attack Detection by Integrating Graph Kernels and Deep Learning Framework

architecture of the proposed framework can be seen in Fig. 3.

B. Graph Structural Information Extraction

The task in this stage is to extract a graph kernel matrix that is computationally inexpensive to compute and is length-independent of the complexity of the network. Kernel method is an effective approach to learn graph data similarity by computing an inner product as the similarity matrix between two graphs. A matrix of pairwise communication structural similarities is created from the communication graphs.

We selected shortest path kernels [8] which captured diverse aspects of the network communication and are efficiently computable, for the reason that shortest path kernels compare the sorted hosts and the length of shortest path that are common between two graphs. The shortest path $d(i,j)$ represents the distance of the shortest path from i to j, reflecting the expectation that nodes with low degrees of separation are likely to interact. In a communication graph context, the shortest path can also encode the structural information of two hosts sending or receiving from each other over time.

The kernel matrix K_{sp} corresponds to a directed attributed graph, where the network hosts are the vertices and the flow meta features are the edge attributes. For each pair of graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ with $S_1 < V, E >$ and $S_2 < V', E' >$, the shortest path kernel is defined as:

$$K_{sp}(\mathcal{G}_i,\mathcal{G}_j) = \langle\varphi_{sp}(\mathcal{G}_i),\varphi_{sp}(\mathcal{G}_j) = \sum_{e\in E}\sum_{e'\in E'}\kappa(e,e')\rangle \quad (1)$$

where κ is shortest path kernel defined on the edges of S_1 and S_2, which measures the similarity between local sub-structures centered at e and e'.

C. Integrating Graph Kernels and Deep Learning

The fully convolutional block and LSTM block process the network flow meta features and graph kernel matrixes, respectively. The fully convolutional block treats the graph kernel matrixes as a time interval with various of communication records. Given a time interval of length α, the fully convolutional block will receive the graph constructed by network flows in α time interval. In contrast, the LSTM block in the proposed framework receives the network flows with a single time step.

The fully convolutional block consists of three stacked convolutional blocks, and then the global pooling is applied after the final convolutional block. At the same time, the network flow meta features are conveyed into a LSTM block. The output of the global pooling layer and the LSTM block is concatenated and passed to a softmax classification layer.

IV. Evaluations

We further evaluate our graph kernels and deep learning integration model on the UNSW-NB15 [9] and CIC-IDS-2017 [10] datasets, respectively. The dataset statistics are shown in Table I. Further, we adopt an algorithm called t-SNE [11] for visualizing high dimensional graph kernel spaces in two-dimensional spaces, to illustrate the calculated graph structural similarities.

A. Datasets

UNSW-NB15 Dataset: This dataset is created by establishing a synthetic environment at the UNSW cybersecurity lab. The data presents a mixture of the real modern normal and the contemporary synthesized attack activities of the network traffic. It had 47 features including new low-footprint attack scenarios and 9 significant series of the cyberattack. We select only 8 network flow meta features from this datasets, namely *dur, sbytes, dbytes, Sload, Dload, Spkts, Dpkts, Sintpkt, Dintpkt*.

CIC-IDS-2017 Dataset: This dataset is a newer dataset that contains a more recent form of attacks such as high-intensity high level layer attacks generated using botnets and low-intensity attacks generated using Slowloris tool. We select only 8 network flow meta features from this datasets, namely *Flow Duration, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Flow Bytes/s, Flow Packets/s, Average Packet Size*.

Both the two datasets have ground truth of the network flows. We divide it into training and test data sets using a ratio of 60% to 40%. The whole prototype is deployed on the

Ubuntu 16.04 64-bit OS. Python3, Keras library and Scikit-learn library are used as the software frameworks.

For evaluation, we report precision, recall, and F_1-measure for the attack detection results achieved by all methods. Every experiment was conducted on a DELL R720 server consisting of a 16 CPU cores with 128 GB of RAM and NVIDIA Quadro GPU.

TABLE I
DATASET STATISTICS

Name	# Catigeries	# Instances	Attack Flow Ratio
UNSW-NB15	10	2,540,044	0.145
CIC-IDS-2017	15	2,830,743	0.167

We implemented a proof-of-concept version of the proposed framework (see Section III). To training the deep learning method, we set the batch size to 128, and the learning rate is set to 0.001. The epochs of training as 50, and the dropout rate is set to 0.5. The time interval t is $60s$.

B. Experimental Design

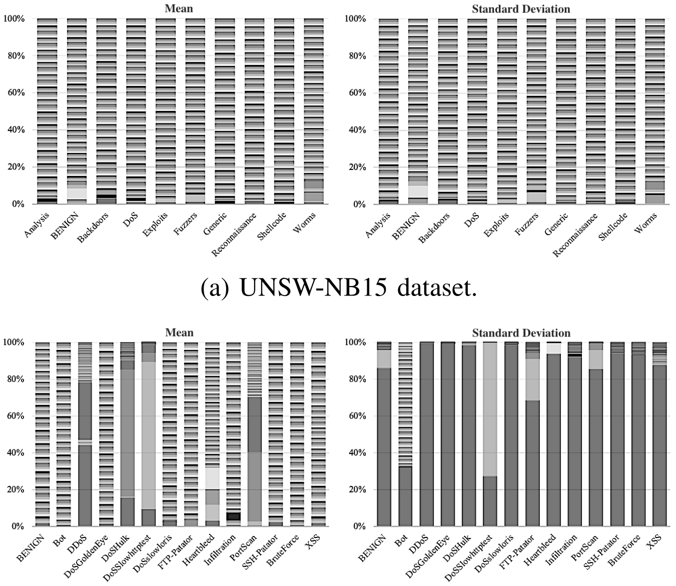
We conduct three experiments: deep neural network models as baseline, integrating graph kernels at input level of deep neural network models, and integrating graph kernels at output level of deep neural network models on two real-world datasets. We select 60 seconds as the time window to reconstruction communication graph as it was shown as a suitable classification point. All experiments were repeated 10 times and we report the means of evaluation matrixes for all algorithms. The best results are underlined. For the compared methods, the LSTM and CNN deep neural networks are considered as baselines. To provide a better overview of the performance of the baseline approach on the statistical network flow features, the overall precision, recall and F_1-measure are presented in Table II.

Both the datasets are divided into two sets: training set and test set, which are the mixture of both attacks and benign flows. Then, an integrated model is trained to predict on the training set and the prediction errors on the training set are fit to a multivariate Gaussian using maximum likelihood estimation. The loss function that we use is the sigmoid cross entropy. The threshold for discriminating between attacks and benign values is then determined via maximizing the accuracy value with respect to the threshold. At last, the trained model is then used to predict on the test set and the results are recorded and compared.

C. Graph Structural Information Statistics and Visualization

To evaluate the influence of communication structural information extracted by graph kernels, we use the learned representations as input for a statistical analysis approaches, and then a qualitative analysis based on t-SNE visualizations.

First, we extract graph structural information from both datasets, respectively, and select 100 records for each attack categories randomly. Then, we measure the statistical distributions. Fig. 4 shows the distributions of the graph structural information. As we can see from the figure, the graph structural information has a good distinguish ability, especially for Denial of Service (DoS) and PortScan attacks.



(a) UNSW-NB15 dataset.

(b) CIC-IDS-2017 dataset.

Fig. 4. The statistical measurement for graph structural information of the two datasets.

Also, we visualize the extract information using the t-SNE algorithm. Fig. 5 illustrates the visualization results. It is observed that the diversities of representation on both datasets are prominent, where some of the intra-class distance can be larger than the inter-class distance and the distribution is heterogeneous.

D. Performance on Deep Learning with Graph Kernels

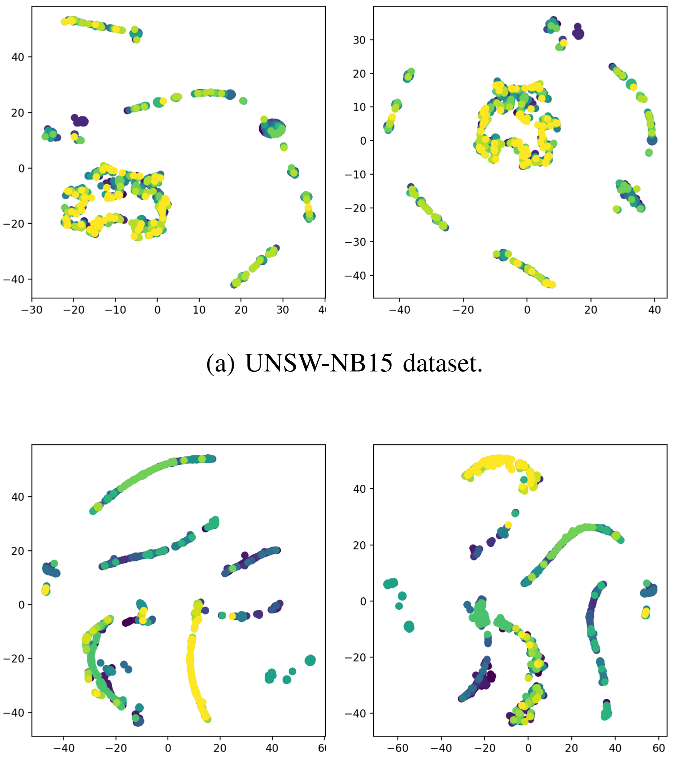
To understand necessity of graph kernels, we further measured the performance of the proposed framework.

Table II details the results of the experiments. From the results, we can see that our framework significantly outperforms baseline methods. We also compare the detection accuracy on the network attack detection task against the random forests classifier proposed in [10] and [12].

After training and obtaining representations of network flow meta features and graph kernels, the results of precision, recall and F_1-measure are shown in Table II, respectively, which we can observe the proposed framework is consistently above baseline methods, and achieves statistically significant improvements on all metrics. As can be seen, compared with the baseline deep learning architecture, the proposed framework improve the precision by 1.2%-2.2% on UNSW-NB15 and 3.4%-12.4% on CIC-IDS-2017, as well as improve the recall by 0.2%-9.6% on UNSW-NB15 and 2.6% on CIC-IDS-2017. Therefore, we can draw the conclusion that the proposed framework maintains a more decent performance in attack detection tasks compared with other methods.

TABLE II
PRECISION, RECALL AND F_1-MEASURE OVER TWO DATASETS OF DIFFERENT METHODS. THE BEST PERFORMANCE IS SHOWN IN **BOLD**.

Evaluation Methods	**UNSW-NB15**			**CIC-IDS-2017**		
	Precision	***Recall***	***F_1-measure***	***Precision***	***Recall***	***F_1-measure***
CNN Baseline	0.967	0.901	0.933	0.864	0.961	0.911
LSTM Baseline	0.977	0.995	0.988	0.954	0.961	0.957
Our Framework	**0.989**	**0.997**	**0.993**	**0.988**	**0.987**	**0.987**
Random Forests [10] [12]	0.999	0.910	0.953	0.980	0.970	0.970



(a) UNSW-NB15 dataset.

(b) CIC-IDS-2017 dataset.

Fig. 5. The t-SNE visualization for graph structural information of the two datasets.

V. RELATED WORK

In this section, we discuss the existing research on network attack detection techniques and categorize as follows:

A. Flow-based Network Attack Detection

Traditional network attack detection methods use deep packet inspection (DPI) to detect malicious activity in the network traffic. However, DPI becomes unsuitable for high-speed network links. Also, the DPI is not practical when most packet content is encrypted. Due to the limitations of the DPI-based attack detection systems, researchers are focusing on flow-based attack detection systems [13]. Since only the flow records are considered, flow-based attack detection process is efficient and independence from encrypted payload.

Winter et al. [14] propose a flow-based intrusion detection method using SVM based one-class classification. After training, the one-class SVM detects the malicious flows with a false alarm rate of 2%. Casas et al. [15] propose an unsupervised network intrusion detection system capable of detecting unknown network attacks to detect the malicious flows and the evaluation results show that their method is well performed on MAWI and KDD99 datasets. Hajisalem et al. [16] propose a new hybrid network attack classification method based on Artificial Bee Colony and Artificial Fish Swarm algorithms and their experimental results on NSL-KDD and UNSW-NB15 datasets demonstrate that the proposed method outperforms in terms of performance metrics and can achieve 99% detection rate and 0.01% false positive rate.

B. Deep Neural Network based Network Attack Detection

Deep learning has been shown tremendous performance on a variety of application areas, such as image, speech and video analysis tasks, while they are high-dimensional inputs and have high computational requirements. For network attack detection, deep neural networks increase the detection rate of known attacks and reduce the false positive rate of unknown attacks.

Alom et al. [17] train deep belief network models for identifying any kind of unknown attack in dataset and evaluated the performance on intrusion detection. Their proposed system not only detects attacks but also classifies them and achieves 97.5% accuracy for only fifty iterations. Alrawashdeh et al. [18] explore the attack detection capabilities on various kinds of deep learning architectures. They outperform the former works in both detection speed and accuracy and achieve a detection rate of 97.9% presenting machine learning approaches for predicting attacks with reasonable understanding. Zhang et al. [19] propose a specially designed CNN to detect web attacks. They found only some preprocessing is needed whereas the tedious feature extraction is done by the CNN itself and the experimental results show that the designed CNN has a good performance. Chawla et al. [20] use RNN as their model for host-based intrusion detection systems which determine normal behavior based on sequences of system calls.

C. Graph based Network Attack Detection

The ability to exploit the potential relationships of communication patterns in network traffic has been the focus of many existing studies. Several graph-based and machine learning techniques have been investigated over the last two decades for detecting network attack detection [21].

Chen et al. [22] propose a spectral decomposition approach for single graph analysis that integrate multiple features, containing graph walk statistics, centrality measures and graph distances to reference nodes. When applying to network attack detection, their approach can effectively indicate anomalous connectivity pattern and provide discriminative basis for attack classification. Yao et al. [23] proposed an effective method for deep graph transfer feature extraction to classify network attack based on network flows.

VI. Conclusions

Graph kernels and deep learning have been successfully applied to many graph based attack detection techniques. In this paper, we present a novel framework of performing deep learning on the network communication graph by coupling graph kernels and convolutional based neural network design. We have implemented a prototype of the proposed framework and evaluated it on two real-world network traffic traces. Our evaluation results have demonstrated the accuracy of the proposed framework. Our research also shows a successful integrated application of deep learning and graph kernels on computer security problems.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2017YFC0821804-2, No. 2018YFB0803602, No. 2016QY06X1204), Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDC02040100, No. XDC02030200, No. XDC02020200) and National Natural Science Foundation of China (No. 61802404). This research is also supported by Key Laboratory of Network Assessment Technology, Chinese Academy of Sciences and Beijing Key Laboratory of Network Security and Protection Technology. We thank many people for their helpful comments on drafts of this paper and the anonymous reviewers for their feedbacks on the paper.

References

[1] C. Liaskos, V. Kotronis, and X. Dimitropoulos, "A novel framework for modeling and mitigating distributed link flooding attacks," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[2] M. S. Kang, S. B. Lee, and V. D. Gligor, "The crossfire attack," in *2013 IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 127–141.

[3] M. Ring, A. Dallmann, D. Landes, and A. Hotho, "IP2vec: Learning similarities between ip addresses," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 657–666.

[4] J. Navarro, A. Deruyver, and P. Parrend, "A systematic survey on multi-step attack detection," *Computers & Security*, vol. 76, pp. 214–249, 2018.

[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 1093–1110.

[6] D. Gkounis, V. Kotronis, C. Liaskos, and X. Dimitropoulos, "On the interplay of link-flooding attacks and traffic engineering," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 2, pp. 5–11, 2016.

[7] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2017.

[8] K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *Fifth IEEE international conference on data mining (ICDM'05)*. IEEE, 2005, pp. 8–pp.

[9] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[10] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.

[11] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[12] V. K. Rahul, R. Vinayakumar, K. Soman, and P. Poornachandran, "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2018, pp. 1–6.

[13] M. F. Umer, M. Sher, and Y. Bi, "Flow-based intrusion detection: Techniques and challenges," *Computers & Security*, vol. 70, pp. 238–254, 2017.

[14] P. Winter, E. Hermann, and M. Zeilinger, "Inductive intrusion detection in flow-based network data using one-class support vector machines," in *2011 4th IFIP international conference on new technologies, mobility and security*. IEEE, 2011, pp. 1–5.

[15] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Computer Communications*, vol. 35, no. 7, pp. 772–783, 2012.

[16] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[17] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *2015 National Aerospace and Electronics Conference (NAECON)*. IEEE, 2015, pp. 339–344.

[18] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2016, pp. 195–200.

[19] M. Zhang, B. Xu, S. Bai, S. Lu, and Z. Lin, "A deep learning method to detect web attacks using a specially designed cnn," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 828–836.

[20] A. Chawla, B. Lee, S. Fallon, and P. Jacob, "Host based intrusion detection system with combined cnn/rnn model," in *Proceedings of Second International Workshop on AI in Security*, 2018.

[21] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[22] P.-Y. Chen, S. Choudhury, and A. O. Hero, "Multi-centrality graph spectral decompositions and their application to cyber intrusion detection," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4553–4557.

[23] Y. Yao, L. Su, and Z. Lu, "DeepGFL: Deep Feature Learning via Graph for Attack Detection on Flow-Based Network Traffic," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 579–584.