Detecting Internet Worms, Ransomware, and Blackouts Using Recurrent Neural Networks

Zhida Li, Ana Laura Gonzalez Rios, and Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
Email: {zhidal, anag, ljilja}@sfu.ca

***Abstract*—Analyzing and detecting Border Gateway Protocol (BGP) anomalies are topics of great interest in cybersecurity. Various anomaly detection approaches such as time series and historical-based analysis, statistical validation, reachability checks, and machine learning have been applied to BGP datasets. In this paper, we use BGP update messages collected from Réseaux IP Européens and Route Views to detect BGP anomalies caused by Slammer worm, WannaCrypt ransomware, and Moscow blackout by employing recurrent neural network machine learning algorithms.**

***Index Terms*—network anomalies, BGP, RIPE, Route Views, machine learning, recurrent neural networks**

I. Introduction

Border gateway protocol (BGP) is an incremental path vector routing protocol that manages network reachability information and optimally routes data between Internet autonomous systems (ASes). An AS is a collection of BGP routers (peers) within a single administrative domain. ASes are identified with a unique number allocated by a corresponding regional Internet registry (RIR): African network information center (AFRINIC), American registry for Internet numbers (ARIN), Asia-Pacific network information centre (APNIC), Latin America and Caribbean network information centre (LACNIC), and Réseaux IP Européens network coordination centre (RIPE NCC). BGP relies on the Transmission Control Protocol (TCP) (port 179) for reliable router–to–router communication using four message types: open, keepalive, update, and notification. BGP update and withdrawal messages are notifications about changes in network topology and reachability. BGP data are used to analyse the Internet topology and hierarchy, infer AS relationships [1], and evaluate various intrusion and anomaly detection mechanisms [2], [3]. Data are collected using BGP trace collectors (RIPE [4] and Route Views [5]), route servers, looking glasses, and the Internet routing registries. Various BGP data collections may be combined to provide a more complete Internet topology [6].

BGP is prone to anomalies that impede successful exchange of reachability messages and may generate a large volume of anomalous updates messages [2], [7]. Hence, several modifications have been proposed to improve BGP security [8], [9]. BGP anomalies include worms (Slammer), ransomware attacks (WannaCrypt), routing misconfigurations [10], Internet Protocol (IP) prefix hijacks [11], and link failures [12] (Moscow blackout). BGP anomalies [13], [14] may be classified using various machine learning algorithms such as support vector machine [15], recurrent neural networks (RNNs), and broad learning system [16].

In this paper, we classify network anomalies emanating from Slammer worm, WannaCrypt ransomware, and Moscow blackout by extracting BGP update messages from RIPE and Route Views data collection sites. Performance of long short-term memory (LSTM) [17] and gated recurrent unit (GRU) [18] RNNs classification algorithms is compared.

The paper is organized as follows: BGP data collection sites are introduced in Section II. Details of BGP anomalies (Slammer, WannaCrypt, and Moscow blackout) are given in Section III while extracted BGP datasets are described in Section IV. The experimental procedure and comparison of machine learning algorithms are given in Section V. We conclude with Section VI.

II. BGP Data Collections

BGP routing information is shared by Internet service providers (ISPs) located in various geographical locations [6]. BGP trace collectors receive BGP messages from their peers and periodically store the routing updates and tables into publicly available archives. Routing tables contain numerous entries from each peering AS that indicate the preferred paths to destination prefixes at a given time. Routing messages indicate alternative paths and backup links. BGP routing update messages are available from global BGP monitoring systems such as RIPE [4] and Route Views [5]. They may be collected using Quagga [19], a suite derived from the multi-server routing software Zebra [20]. BGP update messages are stored in multi-threaded routing toolkit (MRT) format.

A. RIPE

The RIPE routing information service (RIS) [21] is a RIPE NCC project established in 2001 to collect and store routing data from several ASes worldwide. The main collector is located at NCC and consists of a route collector, database, and user interface. Remote route collectors (rrcs) installed at major topologically interesting Internet points use the Quagga routing software to collect BGP data. Routes are collected directly from the AS border routers at the rrc or via multi-hop BGP peering from nearby routers. The raw data are collected using state dumps while batches of updates for each rrc are

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant R31-611284.

periodically made available. The Zebra tool is used to collect BGP update messages every 15 min before July 23, 2003 and every 5 min afterwards while the BGP routing tables are stored every 8 h. RIS currently consists of 25 rrcs: Europe (16), North America (4), Asia (2), South America (2), and Africa (1).

B. Route Views

Route Views [5] is the University of Oregon project to collect real-time BGP routing data from various backbone routers and locations worldwide. The publicly available data have been used for routing analysis, AS path visualization, analysis of IPv4 address space utilization, topological studies, and generation of geographic host locations. Backbone routers (Cisco, Juniper), configured as IPv4 or IPv6 Route-Views-like route servers, connect as peers via multi-hop BGP sessions.

Route Views project employs three types of collectors: FRRouting, Quagga, and Cisco. FRRouting and Quagga collectors are based on Zebra. BGP update messages and routing tables are stored in MRT format and collected every 15 min and 2 h, respectively. Data from Cisco collectors are generated every 2 h starting at 00:00 by using the Cisco command line interface to extract routes and their attributes. There are 31 Route Views collectors (16 FRRouting, 14 Quagga, and 1 Cisco) distributed across RIRs: ARIN (14), LACNIC (6), APNIC (5), AFRINIC (3), and RIPE NCC (3).

III. BGP Anomalies

BGP anomalies are harmful changes in the protocol's behaviour and may consist of single updates (invalid AS numbers, invalid or reserved IP prefixes, a prefix announced by an illegitimate AS, AS-PATH without a physical equivalent) or a set of updates (longest and shortest paths, changes in the behaviour of BGP traffic over time) [2]. These anomalies are classified as: direct intended anomalies, direct unintended anomalies, indirect anomalies, and link failures. BGP hijackings are direct intended anomalies where the attacker redirects routes from a valid AS by claiming the ownership of a prefix or sub-prefix. Denial of service (DoS), distributed DoS (DDoS), man-in-the-middle, and phishing attacks employ BGP hijackings. BGP misconfigurations are direct unintended anomalies that may cause announcements of used (hijacking) or unused (leaked routers) prefixes. The origin misconfigurations occur when non-owned prefixes are accidentally announced or private ASes are not filtered while export misconfigurations appear when BGP policies are accidentally configured. BGP indirect anomalies occur when Internet web servers are attacked, which generates BGP instabilities such as routing overload. For example, during the Slammer worm attack a critical BGP instability was caused by a significant increase in the number of announcements of BGP updates. A BGP link failure causes reachability or connectivity loss between private (dedicated connection) or public (service provider) BGP peers. The Moscow power system blackout (2005) and Mediterranean cable break (2008) resulted in BGP link failures that affected cities in more than 20 countries.

Worms are self-replicating codes that exploit systems vulnerabilities and propagate via networks [22], [23]. They employ email applications or scan engines to spread to various hosts and may carry other malware as their payload. While antivirus systems may require several hours to identify worms, an Intrusion Detection System (IDS) is capable of detecting worms faster because they take large portion of a network bandwidth. Slammer [24], Nimda [25], and Code Red I [26] are well-known worms that exploited vulnerabilities of Microsoft Structured Query Language (SQL) and Internet Information Services (IIS).

Ransomware attacks rely on advanced cryptography to lock the victim's data until a ransom is paid. They may be classified as: cryptoworm, ransomware-as-a-service (RaaS), and automated active adversary [27]. Cryptoworms replicate themselves to other hosts for maximum reach and impact. RaaS attacks, sold on the dark web as distribution kits, are typically deployed via malicious spam e-mails or exploit kits. In case of automated active adversary ransomware, the attackers scan the Internet for systems with weak protection, enter the system, and plan the attack for maximum damage. Ransomware attacks rely on tools and processes such as runtime packers and exploits. Runtime packers are compressed executable-files used to avoid detection of attacks until they have completed their core task while exploits (EternalBlue, Windows Event Viewer process, CVE-2018-8453) are tools that ensure that the attacks gain administrative privileges by taking advantage of the vulnerabilities in an operating system. A ransomware may store the encrypted data on the same (overwrite) or available (copy) disk sectors. During the encryption, data are partially or fully renamed. Well-known ransomware attacks include WannaCrypt [27], Petya, and Locky [28].

Power system blackouts are the loss of electrical power to end users and are caused by failures or overload of transmission lines, failures of automatic emergency control systems, malfunctions of protection devices, or human errors. Blackouts are critical to environment and public safety and, hence, investigating their cascading process is important to determine triggering events, evaluate the consequences, and develop preventive solutions and automatic protection systems [29].

A. Slammer

Slammer [24], [30] is the fastest worm that self-propagated by using the User Datagram Protocol (UDP). It commenced on January 23, 2003 at 05:31 (GMT) and lasted 14.5 h. This worm does not store itself in the memory of affected hosts. It only exists as a network packet and acts by running processes in the victim's host. During the Slammer attack, SQL servers and PCs with Microsoft SQL server data engine (MSDE) were infected by exploiting the buffer overflow vulnerability in Microsoft SQL server 2000 resolution service. The code replicated itself by infecting new vulnerable machines through scanning randomly generated IP addresses and sending packets to UDP port 1434. The number of infected machines doubled approximately every 9 s. This infection speed caused a DoS attack in affected networks.

The SQL server resolution service that operates on UDP port 1434 allows clients to host multiple instances of SQL Server on a single machine. Three restricted bytes that exploit the SQL Server vulnerabilities are: 0x04 (buffer overflow), 0x08 (buffer overflow), and 0x0A (DoS). The first byte in the single UDP packet sent by Slammer is 0x04, indicating to SQL Server that the remaining data of the packet corresponds to the name of the online database. Because Slammer appends a name with more than 16 bytes to the end of this UDP packet omitting the telltale "00", the stack is overflowed, the return address is overwritten, and a full control of the SQL Server process is gained without the need to authenticate. Once the SQL Server is reprogrammed, Slammer replicates to the randomly generated IP addresses.

B. WannaCrypt

WannaCrypt (WannaCry) is a cryptoworm ransomware that works by gaining administrative privileges and employs the EternalBlue exploit and DoublePulsar backdoor in systems running Microsoft Windows 7 [27], [31]. It lasted from May 12, 2017 to May 15, 2017 and infected over 230,000 computers in 150 countries. A victim's data files are encrypted using 128-bit advanced encryption standard (AES) in cipher block chaining (CBC) mode. After the encryption is completed, data files are renamed by adding the extension ".wncry" while the string "WannaCrypt!" is added to the combination of encrypted AES key and data. Wannacrypt may copy or overwrite the data after encryption. It uses the Volume Shadow Service (vssadmin.exe) Windows utility to delete previous versions of the locked data. By manipulating the Windows Boot Configuration Data (bcdedi.exe), the attack: (1) prevents Windows diagnostics-and-repair feature to automatically run after a third unsuccessful boot or (2) attempts a normal boot even in case of a failed boot, shutdown, or checkpoint. WannaCrypt flushes buffers to ensure that all encrypted data are only located in the storage drive. It replaces the Windows desktop wallpaper with a message to inform the victim that data have been locked and to demand a ransom. After the ransom is paid, the risk remains that decryption of data fails.

WannaCrypt spreads throughout a network by attempting to connect to port 445. After the connection is established, the ransomware scans for the Windows server message block (SMB) EternalBlue vulnerability and checks if it is infected with the DoublePulsar backdoor. EternalBlue exploits the wrong casting, wrong parsing, and non-paged pool allocation defects of the SMB protocol as well as an address space layout randomization (ASLR) bypass. Exploiting the wrong casting and parsing defects causes buffer overflow and overwrite while the non-paged pool allocation and ASLR allow placing the shellcode at a predefined executable address [32]. EternalBlue then implants the DoublePulsar backdoor in the victim's host to send the cryptoworm payload using dynamic link library (DLL) injection [31], [32]. WannaCrypt replicates by querying for the non-existing domains:

- www[.]iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com
- www[.]ifferfsodp9ifjaposdfjhgosurijfaewrwergwea[.]com.

Its replication may be prevented if the victims receive a response indicating that these domains are registered [33].

C. Moscow Blackout

The Chagino substation of the Moscow energy ring experienced a transformer failure on May 24, 2005 at 20:57 (MSK) [34], [35]. The event caused a complete shutdown of the substation and a blackout that affected all customer until 16:00 (MSK) of May 26, 2005. At 11:00 (MSK) on May 25, 2005, Unified Energy System of Russia set up an Emergency Response Center to eliminate the blackout. The dispatching control staff and automatic protective devices stopped the cascading failure in 2 h and 20 min and power was restored to all socially important facilities and vital infrastructure in Moscow by 18:00 (MSK). The Moscow city transportation system regained power at 21:00 (MSK). Power was fully restored to all customers on May 26, 2005.

During the blackout, the Internet traffic exchange point MSK-IX was disconnected from 11:00 to 17:00 (MSK) [36]. Routing instabilities were observed because several MSK-IX ISP peers lost connectivity for an extended period [37] and primarily affected ASes in Russia and the APNIC RIR [38]. Based on reports [34], the peak power outage occurred during the morning peak load and its duration was 4 h [39]. Hence, in this study, we consider as anomalies data collected from 7:00 to 10:59 (MSK) on May 25, 2005.

IV. BGP Datasets

BGP datasets are extracted from update messages downloaded from RIPE [4] and Route Views [5] collection sites. The data collected during periods of Internet anomalies include five days of Slammer, eight days of WannaCrypt, and five days of Moscow blackout events: the days of the attack as well as two days prior and two days after the attack. Granularity of collected routing records is 1 min of routing records. The update messages are processed using *BGP_Anomaly_Detection* tool. 37 features, classified as AS-path and volume features, are extracted. Each day contains data points extracted from the BGP update messages [13]. The created matrices for Slammer, WannaCrypt, and Moscow blackout consist of collected data points and extracted features. Training and test datastes for Slammer, WannaCrypt, and Moscow blackout from RIPE and Route Views are shown in Table I.

Examples of features that exhibit visible differences in patterns during regular and anomalous events for the Slammer, WannaCrypt, and Moscow blackout BGP datasets are shown in Fig. 1. For both collection sites, the number of BGP announcements and the number of announced network layer reachability information (NLRI) prefixes for Slammer, WannaCrypt, and Moscow blackout increase. However, some features better illustrate the anomalies in RIPE dataset due to missing data in Route Views for Moscow blackout. The selected collectors rrc04 (RIPE) contains 8 peer ASes and 8 routers while route-views2 (Route Views) contains 37 peer ASes and 45 peer routers [6]. This larger number of the

TABLE I
Duration of Analyzed BGP Events and Number of Data Points in BGP Datasets

Collection	**Dataset**	**Regular**	**Anomaly**	**Regular**	**Anomaly**	**Regular**	**Anomaly**	**Collection date**	
site		(min)	(min)	(training)	(training)	(test)	(test)	**Start**	**End**
RIPE	Slammer	6,331	869	3,210	530	3,121	339	23.01.2003 00:00:00	27.01.2003 23:59:59
	WannaCrypt	5,760	5,760	2,880	3,420	2,880	2,340	10.05.2017 00:00:00	17.05.2017 23:59:59
	Msocow b/o	6,960	240	3,120	180	3,840	60	23.05.2005 00:00:00	27.05.2005 23:59:59
Route Views	Slammer	6,319	869	3,198	530	3,121	339	23.01.2003 00:00:00	27.01.2003 23:59:59
	WannaCrypt	5,760	5,760	2,880	3,420	2,880	2,340	10.05.2017 00:00:00	17.05.2017 23:59:59
	Msocow b/o	6,865	130	3,075	85	3,790	45	23.05.2005 00:00:00	27.05.2005 23:59:59

update messages collected by route-views2 better illustrates the presence of anomalies.

Several features extracted from RIPE and Route Views datasets are visualized in scattered plots shown in Fig. 2, Fig. 3, and Fig. 4. These graphs indicate spatial separation for regular and anomalous classes. Separation into two distinct classes is more visible for Slammer and WannaCrypt in Route Views data while separation of the Moscow blackout data is more prominent in RIPE. Better separation of spatial patterns leads to higher classification accuracy.

V. Experimental Procedure and Performance Comparison

We process BGP raw data from RIPE and Route Views and perform two-way classification to identify regular (0) and anomalous (1) data based on Slammer worm, WannaCrypt ransomware, and Moscow blackout events using the *BGP_Anomaly_Detection* tool. Deep learning RNN (LSTM and GRU) models are utilized due to their unique structure and capability to classify time series data.

We label data points corresponding to Slammer, WannaCrypt, or Moscow blackout as anomalous data and employ supervised machine learning to classify anomalies. BGP anomalies caused by Slammer and WannaCrypt resulted in visible changes in volume (number of BGP announcements and BGP withdrawals) and AS-path (average AS-path length and average edit distance) features. In case of the BGP link failures experienced during the Moscow blackout, some affected ASes found alternative routes, which resulted in an inconclusive period of the Internet anomaly and a narrower window compared to the power system downtime [34]. BGP changes are primarily observed in volume features (number of BGP announcements, number of announced NLRI prefixes, and number of interior gateway protocol packets).

A. Deep Learning: Multi-Layer Networks

Deep learning neural networks are trained to identify important features in the input data by adjusting weights in each iteration. Their notable advantage is the back-propagation method that calculates gradients and updates the weights [40], [41]. Furthermore, they may achieve desired results by adjusting the number of hidden nodes, hidden layers, optimization algorithms, and activation functions. The numbers of hidden nodes and layers are chosen depending on the size of the dataset. Note that adding hidden layers may not achieve higher accuracy because of over-fitting.

RNN is a class of neural networks that is often applied to time series datasets. An important advantage of RNNs is their ability to use contextual information from the input sequential data. LSTM [17] is a type of RNNs that consists of forget, input, and output gates that learn relevant long-term dependencies in sequential input data by transferring the cell state through the network. GRU [18] is a variation of LSTM with a simpler structure that contains only two gates: reset and update gates. Performance of deep learning models often improves by including additional hidden layers and nodes, which depends on the length of the dataset and the number of features. To prevent over-fitting, we consider models with only up to four hidden layers and a small number of hidden nodes near the output layer. We evaluate performance of LSTM and GRU models with 2 (LSTM_2 and GRU_2), 3 (LSTM_3 and GRU_3), and 4 (LSTM_4 and GRU_4) hidden layers. A model with 4 hidden layers is shown in Fig. 5.

We implement deep learning RNN models using one GPU (NVIDIA GeForce GTX 1080 GPU) on a Dell Alienware Aurora with 32 GB memory and Intel Core i7 7700K processor. Python 3.6 running on Ubuntu 16.04 was used to generate simulation results. PyTorch [42], an open source machine learning library, is employed to create deep learning models while function *torch.optim.Adam()* is used to optimize the deep learning RNN models during training. Cross-validation is performed for various parameters. The best parameters for training models are shown in Table II.

TABLE II
Parameters of RNN Models Used in Cross-Validation

Parameter	**Value**	**Best selection**
Length of input sequence	5, 10, 20, 50, 100	Slammer: 10 WannaCrypt: 100 Moscow b/o: 100 (RIPE), 20 (Route Views)
Number of epochs	30, 50, 100	30
Number of hidden nodes	80, 64, 32, 16	Slammer: FC_1 = 80, FC_2 = 32, FC_3 = 16 WannaCrypt/Moscow b/o: FC_1 = 64, FC_2 = 32, FC_3 = 16
Dropout rate	0.2, 0.4, 0.6	0.4
Learning rate	0.01, 0.1	0.01

B. BGP_Anomaly_Detection Tool

The tool shown in Fig. 6 consists of eight modules:

Data_Download: The input to the data download module are the names of files with update messages, collection site

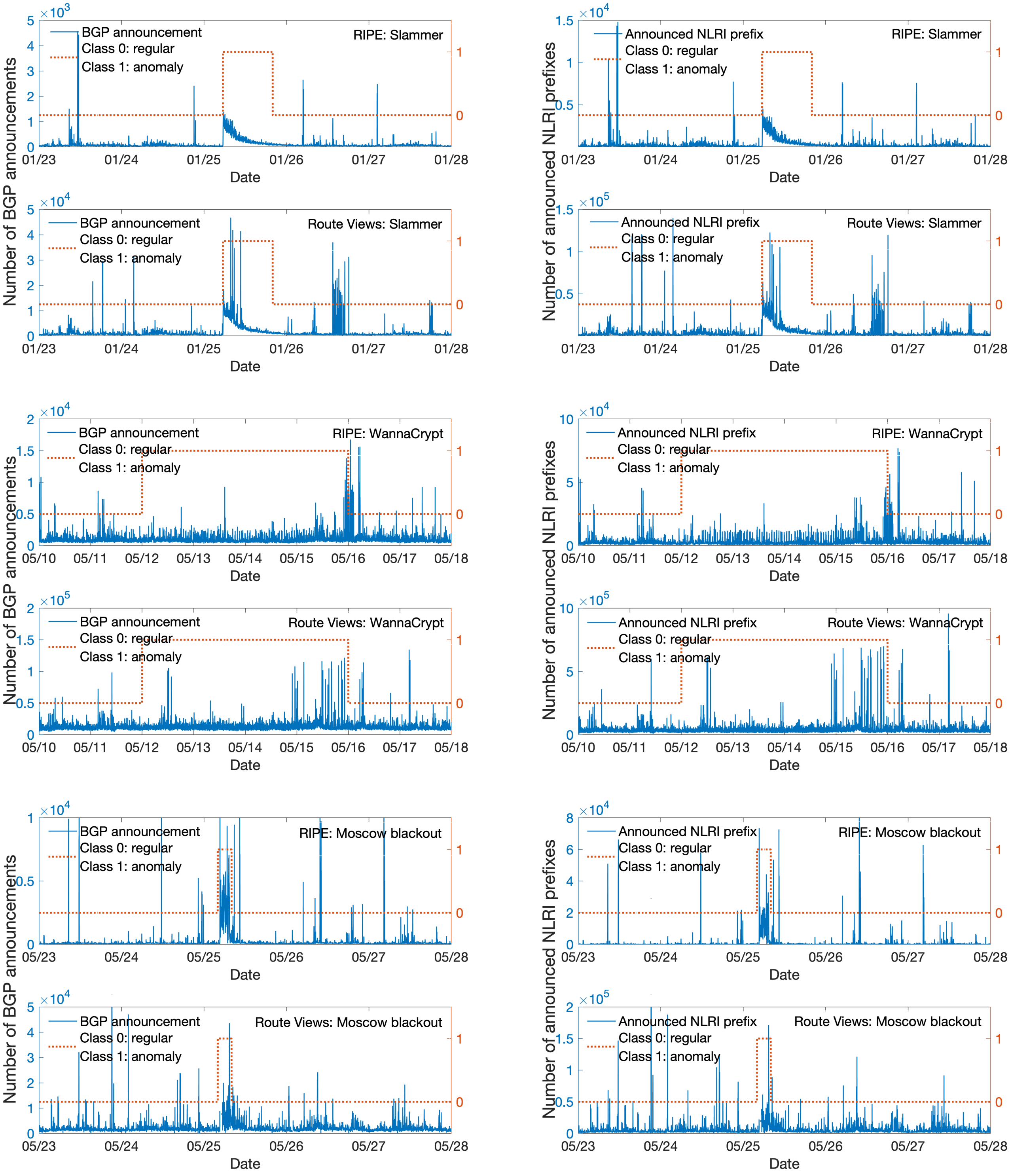

Fig. 1. Slammer (top), WannaCrypt (middle), and Moscow blackout (bottom): Number of BGP announcements and announced NLRI prefixes.

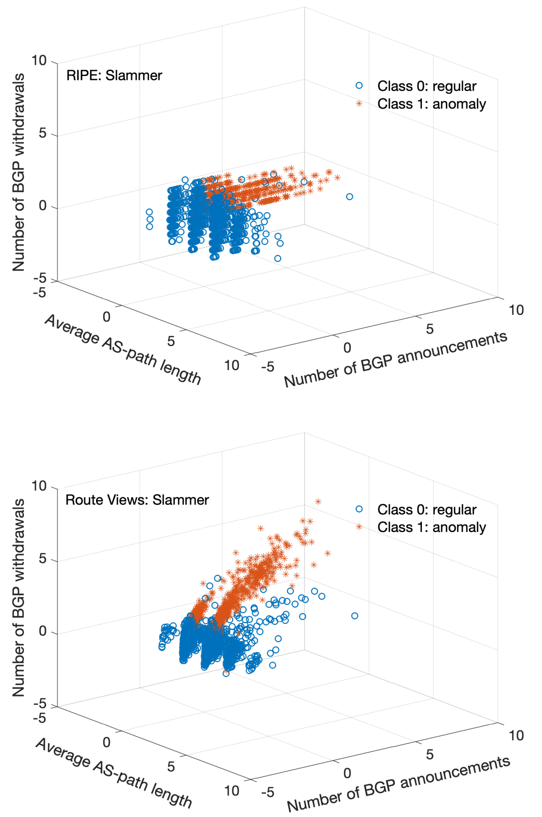

Fig. 2. Slammer: Average AS-path length vs. number of BGP announcements vs. number of BGP withdrawals.

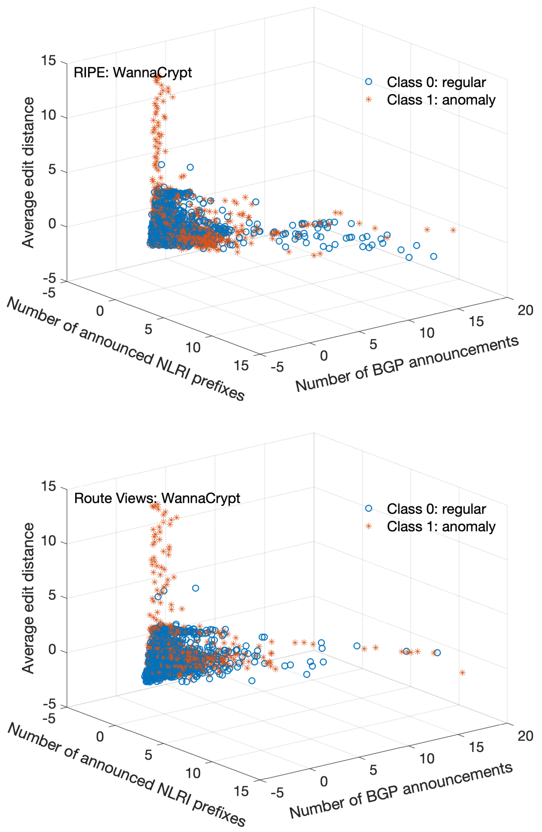

Fig. 3. WannaCrypt: Number of announced NLRI prefixes vs. number of BGP announcements vs. average edit distance.

(RIPE or Route Views), and collector name (RIPE rrc04 or Route Views route-views2). The output are update messages in ASCII format. Raw data from RIPE and Route Views are organized in folders labeled by the year and month of the collection date. The format of the selected and downloaded BGP update messages is *updates.yyyymmdd.hhmm.gz* or *updates.yyyymmdd.hhmm.bz2* for RIPE and Route Views datasets, respectively. BGP update messages are initially collected in MRT format. They are transformed from MRT to ASCII format by using the *zebra-dump-parser* [43] tool written in Perl. GMT time is used for all update messages in order to synchronize RIPE and Route Views collection times.

Feature_Extraction: A tool written in C# was used to generate datasets by extracting 37 numerical features from BGP update messages [13].

Data_Partition: This module is used to create the training and test datasets based on the percentages of anomalous data. Data are labeled based on the time intervals of collection. In our experiments, the Slammer and WannaCrypt training and test datasets consist of 60 % and 40 % of anomalous data, respectively while Moscow blackout training and test datasets consist of 75 % and 25 % (RIPE) and 65% and 35% (Route Views) of anomalous data, respectively.

Data_Processing: The module consists of feature selection and normalization steps. The most relevant features may be selected for both training and test datasets by using the extremely randomized trees (extra trees) [44] feature selection algorithm. (In this experiments, we do not preform feature selection.) The input to the module are: number of the most relevant features, file name, and labels of the dataset. By employing the *zscore* function, we generate datasets with $mean = 0$ and $standard\ deviation = 1$.

ML_Algorithms: The module contains various deep learning RNN models with a number of hidden layers. Input parameters are: RNN algorithm, number of hidden layers and nodes, number of epochs, learning rate, activation function, and dropout rate.

Parameters: Parameters for cross-validation are stored in this module. The best set of parameters selected by cross-validation is used in the training process.

ML_Models: This module generates machine learning models using training datasets.

Classification: Accuracy, F-Score, precision, sensitivity (recall), receiver operating characteristic (ROC) curves, and training time may be used to evaluate performance of classification algorithms.

C. Performance Comparison

We evaluate performance of deep learning RNN classification models based on accuracy and F-Score. Performance

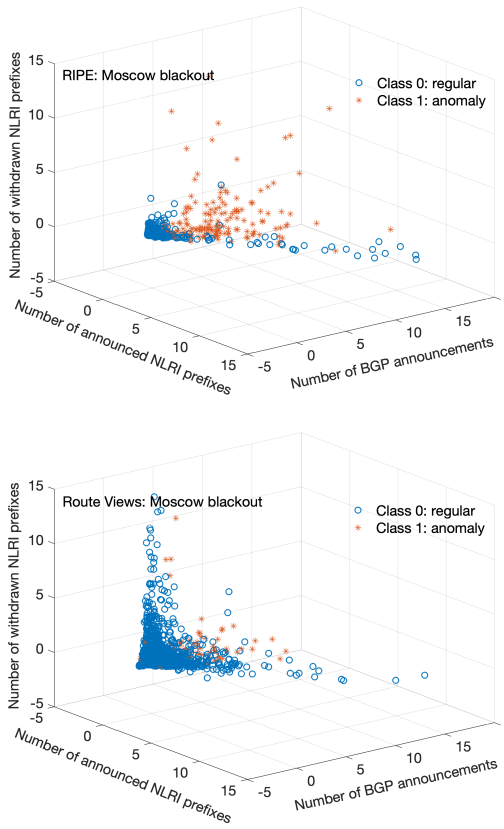

Fig. 4. Moscow blackout: Number of announced NLRI prefixes vs. number of BGP announcements vs. number of withdrawn NLRI prefixes.

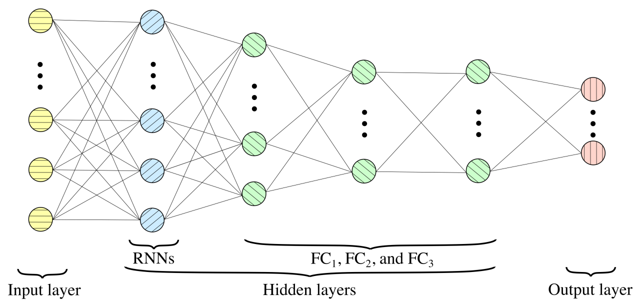

Fig. 5. Deep learning neural network model. It consists of 37 RNNs, 80 (Slammer)/64 (WannaCrypt)/64 (Moscow blackout) FC_1, 32 FC_2, and 16 FC_3 fully connected (FC) hidden nodes.

of LSTM and GRU models with various hidden layers using Slammer, WannaCrypt, and Moscow blackout datasets is shown in Table III. The best classification results for RIPE datasets are achieved using $LSTM_2$ (Slammer and Moscow blackout) and $LSTM_3$ (WannaCrypt) models. For the Route Views datasets, the best classification results are obtained using $LSTM_3$ (Slammer), GRU_3 and GRU_2 (WannaCrypt), and GRU_2 and GRU_4 (Moscow blackout) models. It has been observed that increasing the number of hidden layers may result in over-fitting. As expected, the best accuracy and F-Score generated by RNN models using Route Views datasets are higher than RIPE. Moscow blackout data collected by RIPE during the time of anomaly are more reliable than Route Views data, hence better classification results. Note that a much smaller number of anomalous data points (130) is collected during the four-hour interval.

TABLE III
PERFORMANCE OF LSTM AND GRU RNN MODELS: SLAMMER, WANNACRYPT, AND MOSCOW BLACKOUT DATASETS

		Accuracy (%)		**F-Score** (%)	
Model	**Dataset**	**RIPE**	**Route Views**	**RIPE**	**Route Views**
$LSTM_2$	Slammer	92.98	91.24	72.42	69.11
	WannaCrypt	58.08	67.23	61.48	70.14
	Moscow b/o	99.21	96.23	75.20	5.26
$LSTM_3$	Slammer	90.90	95.72	67.29	81.77
	WannaCrypt	65.48	64.35	63.22	67.16
	Moscow b/o	98.38	97.77	55.94	32.00
$LSTM_4$	Slammer	92.49	91.39	70.72	69.34
	WannaCrypt	57.94	72.29	62.42	73.86
	Moscow b/o	97.46	95.81	36.94	18.37
GRU_2	Slammer	91.88	92.60	69.42	72.59
	WannaCrypt	57.27	72.58	60.56	74.21
	Moscow b/o	97.64	98.30	41.77	32.99
GRU_3	Slammer	91.76	93.24	68.72	74.34
	WannaCrypt	52.85	72.63	53.96	74.14
	Moscow b/o	98.38	97.51	57.14	28.57
GRU_4	Slammer	92.14	93.15	70.11	74.04
	WannaCrypt	52.15	68.71	52.70	71.61
	Moscow b/o	97.92	97.20	49.06	35.15

VI. CONCLUSION

We considered BGP update messages from RIPE and Route Views data collection sites to classify Slammer, WannaCrypt, and Moscow blackout anomalous events. We implemented deep learning RNN (LSTM and GRU) models with a variable number of hidden layers. Models with two and three hidden layers often exhibited the best performance. The best accuracy and F-Score for Slammer and WannaCrypt were generated using BGP update messages collected by Route Views. In contrast, the best performance for classifying Moscow blackout was obtained using RIPE datasets due to missing data points in Route Views. Classification models for Slammer datasets offered better results because the data had better spatial separation between regular and anomalous classes.

REFERENCES

[1] H. Yan, R. Oliveira, K. Burnett, D. Matthews, L. Zhang, and D. Massey, “BGPmon: a real-time, scalable, extensible monitoring system,” in *Proc. Cybersecurity Appl. Technol. Conf. Homeland Secur.*, Washington, DC, USA, Mar. 2009, pp. 212–223.

[2] B. Al-Musawi, P. Branch, and G. Armitage, “BGP anomaly detection techniques: a survey,” *IEEE Commun. Surv. Tut.*, vol. 19, no. 1, pp. 377–396, 2017.

[3] K. Sriram, O. Borchert, O. Kim, P. Gleichmann, and D. Montgomery, “A comparative analysis of BGP anomaly detection and robustness algorithms,” in *Proc. Cybersecurity Appl. Technol. Conf. Homeland Secur.*, Washington, DC, USA, Mar. 2009, pp. 25–38.

[4] (2020, Sept.) RIPE NCC. [Online]. Available: https://www.ripe.net .

[5] (2020, Sept.) University of Oregon Route Views project. [Online]. Available: http://www.routeviews.org .

[6] B. Zhang, R. Liu, D. Massey, and L. Zhang, “Collecting the Internet AS-level topology,” *ACM Computer Communication Review (CCR)*, vol. 35, no. 1, pp. 53–62, Jan. 2005.

[7] Y. Song, A. Venkataramani, and L. Gao, “Identifying and addressing reachability and policy attacks in ‘secure’ BGP,” *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2969–2982, Oct. 2016.

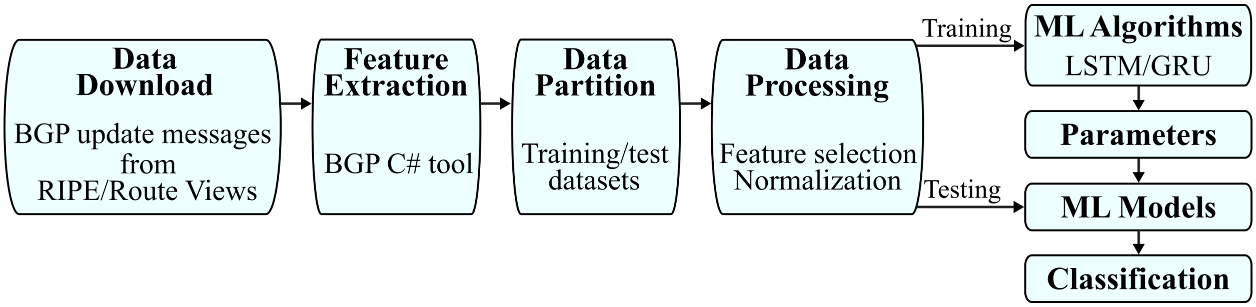

Fig. 6. Architecture of *BGP_Anomaly_Detection* tool for classifying network anomalies and its modules: data download, feature extraction, data partition, data processing, machine learning (ML) algorithms, parameter selection, ML models, and classification.

[8] D. Dolev, S. Jamin, O. Mokryn, and Y. Shavitt, “Internet resiliency to attacks and failures under BGP policy routing,” *Comput. Netw.*, vol. 50, no. 16, pp. 3183–3196, Nov. 2006.

[9] A. Lutu, M. Bagnulo, C. Pelsser, O. Maennel, and J. Cid-Sueiro, “The BGP visibility toolkit: detecting anomalous Internet routing behavior,” *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1237–1250, Apr. 2016.

[10] R. Mahajan, D. Wetherall, and T. Anderson, “Understanding BGP misconfiguration,” in *Proc. 2002 Conf. Appl. Technologies Architectures Protocols Comput. Commun.*, Pittsburgh, Pennsylvania, USA, Aug. 2002, pp. 3–16.

[11] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark, “Profiling BGP serial hijackers: capturing persistent misbehavior in the global routing table,” in *Proc. ACM Internet Meas. Conf.*, Amsterdam, Netherlands, Oct. 2019, pp. 420–434.

[12] J. L. Sobrinho and T. Quelhas, “A theory for the connectivity discovered by routing protocols,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 677–689, June 2012.

[13] Q. Ding, Z. Li, S. Haeri, and L. Trajković, “Application of machine learning techniques to detecting anomalies in communication networks,” in *Cyber Threat Intelligence*, A. Dehghantanha, M. Conti, and T. Dargahi, Eds. Berlin: Springer, 2018, pp. 47–70 and pp. 71–92.

[14] Z. Li, A. L. Gonzalez Rios, G. Xu, and Lj. Trajković, “Machine learning techniques for classifying network anomalies and intrusions,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Sapporo, Japan, May 2019, pp. 1–5.

[15] C. Cortes and V. Vapnik, “Support-vector networks,” *J. Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sept. 1995.

[16] C. L. P. Chen and Z. Liu, “Broad learning system: an effective and efficient incremental learning system without the need for deep architecture,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[17] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: a search space odyssey,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[18] K. Cho, B. van Merriënboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translations,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[19] (2020, Sept.) Quagga routing suite. [Online]. Available: https://www.quagga.net/index.html .

[20] (2020, Sept.) Zebra. [Online]. Available: http://www.zebra.org .

[21] (2020, Sept.) RIPE NCC routing information service. [Online]. Available: https://www.ripe.net/publications/docs/ripe-200 .

[22] (2020, Sept.) Malware 101 - Viruses, SANS Institute. [Online]. Available: https://www.sans.org/reading-room/whitepapers/incident/malware-101-viruses-32848 .

[23] (2020, Sept.) Threat chaos: making sense of the online threat landscape, Webroot Software, Inc. [Online]. Available: https://www.webroot.com/pdf/WP_Threat_0105.pdf .

[24] (2020, Sept.) MS SQL Slammer/Sapphire worm, SANS Institute GIAC Certifications. [Online]. Available: https://www.giac.org/paper/gsec/3091/ms-sql-slammer-sapphire-worm/105136 .

[25] (2020, Sept.) Responding to the Nimda worm: recommendations for addressing blended threats, Symantec, Cupertino, CA, USA. [Online]. Available: https://www-west.symantec.com/content/dam/symantec/docs/security-center/white-papers/nimda-worm-recommendations-blended-threats-01-en.pdf .

[26] (2020, Sept.) The Code Red worm, SANS Institute Information Security Reading Room. [Online]. Available: https://www.sans.org/reading-room/whitepapers/malicious/code-red-worm-85 .

[27] (2020, Sept.) How ransomware attacks, SophosLabs. [Online]. Available: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-ransomware-behavior-report.pdf .

[28] (2020, Sept.) Stemming the exploitation of ict threats and vulnerabilities, United Nations Institute for Disarmament Research (UNIDIR). [Online]. Available: https://unidir.org/files/publications/pdfs/stemming-the-exploitation-of-ict-threats-and-vulnerabilities-en-805.pdf .

[29] Y. V. Makarov, V. I. Reshetov, V. A. Stroev, and N. I. Voropai, “Blackout prevention in the United States, Europe, and Russia,” *Proc. IEEE*, vol. 93, no. 11, pp. 1942–1955, Nov. 2005.

[30] (2020, Sept.) Attack of Slammer worm - a practical case study, SANS institute. [Online]. Available: https://pen-testing.sans.org/resources/papers/gcih/attack-slammer-worm-practical-case-study-103632 .

[31] (2020, Sept.) Reverse engineering of WannaCry worm and anti exploit snort rules, SANS Institute. [Online]. Available: https://www.sans.org/reading-room/whitepapers/malicious/reverse-engineering-wannacry-worm-anti-exploit-snort-rules-38445 .

[32] (2020, Sept.) EternalBlue: a prominent threat actor of 2017–2018, Virus Bulletin. [Online]. Available: https://www.virusbulletin.com/uploads/pdf/magazine/2018/201806-EternalBlue.pdf .

[33] (2020, Sept.) Technical whitepaper tracking the WannaCry ransomware, Nominet. [Online]. Available: https://satisnet.co.uk/wp-content/uploads/2019/04/WannaCry-Whitepaper.pdf .

[34] (2020, Sept.) RAO “UES of Russia” annual report 2005. [Online]. Available: http://www.rustocks.com/put.phtml/EESR_2005_sec.pdf .

[35] (2020, Sept.) Report on the investigation of the accident in the UES of Russia on May 25, 2005. [Online]. Available: http://www.kef.ru/art_010.shtml .

[36] (2020, Sept.) Moscow electroshock. [Online]. Available: https://www.comnews.ru/content/13693 .

[37] S. Deshpande, M. Thottan, T. K. Ho, and B. Sikdar, “An online mechanism for BGP instability detection and analysis,” *IEEE Trans. Comput.*, vol. 58, no. 11, pp. 1470–1484, Nov. 2009.

[38] (2020, Sept.) North American Network Operators Group Mailing List Archive. [Online]. Available: https://archive.nanog.org/mailinglist/mailarchives/old_archive/2005-05/index.html .

[39] (2020, Sept.) Protecting electricity networks from natural hazards. [Online]. Available: https://www.osce.org/secretariat/242651 .

[40] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Netw.*, vol. 18, no. 5-6, pp. 602–610, July/Aug. 2005.

[41] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Mach. Learn.*, vol. 8, no. 3, pp. 229–256, May 1992.

[42] (2020, Sept.) PyTorch. [Online]. Available: https://pytorch.org/docs/stable/nn.html .

[43] (2020, Sept.) zebra-dump-parser. [Online]. Available: https://github.com/rfc1036/zebra-dump-parser .

[44] P. Geurts, D. Ernst, and L.Wehenkel, “Extremely randomized trees,” *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Apr. 2006.