Chapter 13

A NETWORK FORENSIC SCHEME USING CORRENTROPY-VARIATION FOR ATTACK DETECTION

Nour Moustafa and Jill Slay

Abstract Network forensic techniques help track cyber attacks by monitoring and analyzing network traffic. However, due to the large volumes of data in modern networks and sophisticated attacks that mimic normal behavior and/or erase traces to avoid detection, network attack investigations demand intelligent and efficient network forensic techniques. This chapter proposes a network forensic scheme for monitoring and investigating network-based attacks. The scheme captures and stores network traffic data, selects important network traffic features using the chi-square statistic and detects anomalous events using a novel correntropy-variation technique. An evaluation of the network forensic scheme employing the UNSW-NB15 dataset demonstrates its utility and high performance compared with three state-of-the-art approaches.

Keywords: Network forensics, cyber attacks, correntropy-variation technique

1. Introduction

Due to increases in the number and sophistication of network-based attacks, effective techniques and tools are required to conduct network forensic investigations. The WannaCry ransomware attack on Microsoft Windows systems in May 2017 infected more than 230,000 computers in about 150 countries [3]. Advanced techniques are required to analyze network traffic in order to identify potential security policy abuses and information assurance violations, along with their sources [4, 11, 27].

Although extracting network packets for forensic analysis is simple in theory, it requires accurate inspection due to large and diverse network flows [10, 11]. Accurate inspection requires an advanced feature

G. Peterson and S. Shenoi (Eds.): Advances in Digital Forensics XIV, IFIP AICT 532, pp. 225–239, 2018.
https://doi.org/10.1007/978-3-319-99277-8_13

selection method that selects only relevant information, including attack patterns. Identifying the key features supports the aggregation of network observations and investigations of the attack evidence [22].

Several commercial and open-source tools such as NetDetector Suite, PyFlag and Xplico have been developed to support network forensic investigations [6]. These tools collect and analyze flow information in network packets (e.g., IP addresses and port numbers) [9, 11]. However, this information is increasingly unreliable due to device mobility and dynamic IP address allocation [6]. Consequently, exploring dependencies of flow information without analyzing the transitions between flows is challenging.

This chapter presents a high-performance network forensic scheme that supports attack detection and attack source identification. The scheme relies on network traffic captures by the `tcpdump` sniffing tool and stores the captures in a MySQL database to support the analysis and aggregation of network traffic data. Important features related to anomalous activities are selected using the chi-square statistic [5, 15]. Following this, a correntropy-variation technique is applied to determine risk levels for normal and attack samples. The network forensic scheme is evaluated against three state-of-the-art approaches using the UNSW-NB15 dataset [18], which contains comprehensive data related to normal and abnormal network traffic.

2. Related Work

Several network forensic schemes have been proposed in the literature [11, 13, 16]. A typical scheme captures and logs network traffic data in a database to support attack investigations. Important traffic attributes are stored, including flow identifiers such as source/destination IP addresses and ports, along with statistical information about packets such as packet size and packet interval. Sometimes, network packets are marked at routers as network flows move from senders to receivers [13]. A number of machine learning and heuristic approaches have been proposed for modeling and detecting network attacks [7, 11]. These approaches typically employ a training phase to distinguish between normal and suspicious samples, and a testing phase in which the training results are applied to normal and suspicious samples [18].

A number of network forensic frameworks have been proposed [4, 10–12, 27]. Khan et al. [12] have proposed a traceback-based framework for identifying the origins of network packets, especially in investigations of distributed denial-of-service (DDoS) and IP spoofing attacks. Wang et al. [27] have developed a topology-assisted deterministic packet mark-

ing technique based on IP traceback for tracking denial-of-service and distributed denial-of-service attacks. Cheng et al. [4] have employed a cloud-based traceback architecture to tackle access control challenges in cloud systems; their goal is to prevent users from requesting traceback information for malicious reasons.

Converged-network-based frameworks have been proposed for investigating attacks on VoIP communications networks [10, 11]. Attackers have been known to target voice communications by injecting attack packets or modifying legitimate voice packets. Ibrahim et al. [10] have designed an evidence model for investigating attacks on VoIP communications networks by framing hypotheses based on the collected information. He et al. [9] have modeled network vulnerabilities using evidence graphs; network vulnerability evidence and reasoning techniques are used to reconstruct malicious scenarios and backtrack to the network packets to obtain the original evidence. Attack-graph-based frameworks have been employed to discover and visualize possible attack paths in networks [11]. Liu et al. [14] have combined Bayesian inference and evidence graphs; their approach reduces false positive rates by computing the posterior probabilities in the evidence graphs.

Khan et al. [11] have proposed a framework that addresses the scalability problem by distributing network forensic servers and data agent systems [11]. Tafazzoli et al. [23] have proposed a network forensic architecture that comprises five components: (i) collection and indexing; (ii) database management; (iii) analysis; (iv) analysis results communication; and (v) database for collecting and analyzing anomalous patterns.

Finally, network intrusion detection and prevention frameworks are widely used to monitor traffic and mitigate attacks. Network forensic schemes based on intrusion detection typically employ static and dynamic inspection of network traffic to detect anomalies [16]. Wang et al. [26] have proposed hybrid attack detection and distributed forensic techniques for machine-to-machine networks; their techniques specifically combat distributed denial-of-service attacks.

Although existing network forensic frameworks support attack and other incident investigations, they are falling behind in their ability to cope with the large traffic volumes encountered in modern networks. Additionally, network forensic techniques require considerable time and space resources, especially when investigating large-scale distributed networks without aggregating relevant flows that include attack traffic. These challenges have motivated the development of the proposed network forensic scheme for monitoring and investigating network-based attacks. The scheme captures and stores network data, selects impor-

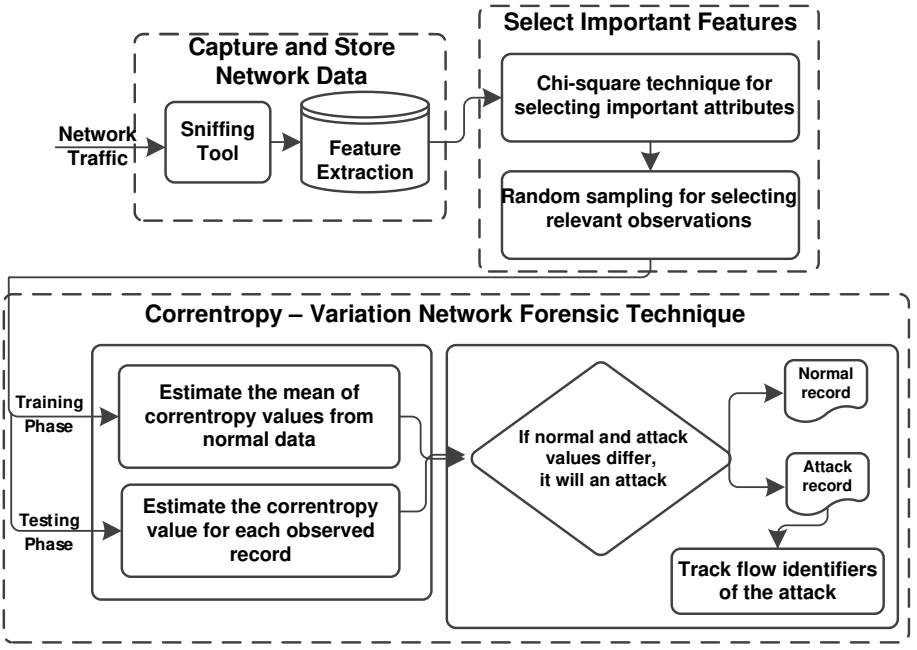

Figure 1. Proposed network forensic scheme.

tant network features using the chi-square statistic and helps investigate anomalous events using a novel correntropy-variation technique.

3. Network Forensic Scheme

The key tasks in network forensics are to log attacker activities and apply forensic techniques to analyze the logged data. This provides valuable information about the attacker and the attack mechanism [8].

Figure 1 illustrates the proposed network forensic scheme. The first step captures network packets via sniffing and stores the captured network packets in a database to facilitate attack investigations. The second step selects important features and removes extraneous and redundant information that might negatively impact attack investigations. The third step investigates attacks and their origins using the novel correntropy-variation technique.

3.1 Capturing and Storing Network Data

The large numbers and volumes of flows in modern networks demand an aggregation method that starts from capturing packets to storing them in a database for supporting network forensic investigations. The `tcpdump` tool is used to capture raw packets from network interfaces.

Network traffic is sniffed at edge devices, in particular, ingress routers in order to collect relevant flows based on their source and destination IP addresses and ports, and protocols. Following this, several packet features are extracted from the captured traffic using the Bro and Argus tools (which were also used to create the UNSW-NB15 dataset).

Traffic data features for each network flow across the network are stored in a MySQL database. The flows are recorded using MySQL Cluster CGE, which can handle large-scale data and support real-time processing.

MySQL functions are used to aggregate data using multiple attributes of the traffic features. This addresses the drawbacks of tools such as Netflow and sFlow that can handle only one feature at a time. Attack path investigations are simplified when the flows between source and destination IP addresses are counted and tracked. For example, distributed denial-of-service attacks send large numbers of pings to disrupt a target; if these pings are monitored and counted, then the attack origins can be identified using the correntropy-variation technique discussed in Section 3.3.

The non-stationary properties of flows between IP addresses and ports are totaled by applying the `Count` function for all possible combinations of flows:

- `Select COUNT(*) as flows, srcip, dstip from network_data group by srcip, dstip;`
- `Select COUNT(*) as flows, srcip, srcport from network_data group by srcip, srcport;`
- `Select COUNT(*) as flows, dstip, dsport from network_data group by dstip, dsport, srcport;`

In the queries, `flows` is a network flow between two IP addresses, `srcip` is the source IP address, `dstip` is the destination IP address, `srcport` is the source port number and `dsport` is the destination port number. Each query retrieves the number of flows that match the specified features.

The collected flows should not include duplicated flows or missing flows. Consequently, uniform random sampling is employed to select arbitrary samples in which no flow is included more than once. All the samples have the same selection probability; moreover, a given pair of samples has the same probability of selection as the other pairs. Uniform random sampling reduces data bias and simplifies data analysis [25].

3.2 Feature Selection

In addition to selecting relevant network flows, it is necessary to identify the important features in the flows. The chi-square (χ^2) feature selection method [5, 15] is employed due to its simplicity and ease of implementation in real-time applications. The χ^2 statistic measures the occurrences of two independent variables associated with their class label, following which the highest ranked variable is selected as an important feature. The χ^2 statistic is computed using the following equation:

$$\chi^2 = \sum_i \frac{(O_i - E)^2}{E} \tag{1}$$

where O_i is the observed value of a sample flow and E is the expected value (mean) of the sample flows.

3.3 Correntropy-Variation Technique

The correntropy-variation technique combines the correntropy measure [2] to estimate the similarities between normal and attack samples, and a variation threshold for discovering attack samples. Correntropy is a nonlinear similarity function that reveals the relationships between normal and anomalous samples whereas variation estimates how different anomalous samples are from normal samples.

The correntropy V_σ of two random variables, f_1 and f_2, is estimated by:

$$V_\sigma(f_1, f_2) = E[\kappa_\sigma(f_1 - f_2)] \tag{2}$$

where σ is the kernel size, $E[.]$ is the expected value of the features and $K_\sigma(.)$ is the Gaussian kernel function computed using the equation:

$$K_\sigma(.) = \frac{1}{\sqrt{2\Pi} \cdot \sigma} \exp(-\frac{(.)^2}{2\sigma^2}) \tag{3}$$

The joint probability density function ($P_{F_1,F_2}(f_1, f_2)$) is not identified. However, a finite number of samples $\{f_i, f_j\}_{i,j=1}^{M}$ are known. Consequently, the correntropy is computed using the equation:

$$\hat{V}_{M,\sigma}(A, B) = \frac{1}{M} \sum_{i,j=1}^{M} K_\sigma(f_i - f_j) \tag{4}$$

In order to apply the correntropy measure to multivariate network data, the correntropy is computed for normal and malicious samples.

The mean of the correntropy values of normal samples $corpy^{normal}$ is computed during the training phase using the following equation:

$$\mu(corpy^{normal}) = \frac{1}{N} \cdot corpy^{normal} \tag{5}$$

where N is the number of normal samples.

During the testing phase, the correntropy value $corpy^{test}$ is estimated for each sample using Equations (4) and (5).

A baseline between $\mu(corpy^{normal})$ and each $corpy^{test}$ value is created using the standard deviation measure δ, which estimates the amount of variation between the mean of the normal correntropy values and the correntropy values of test samples. If the absolute value of the variation is greater than or equal 2δ (i.e., two standard deviations), then the test sample is considered to be an attack; otherwise, it is normal:

- **Attack:** if $|\mu(corpy^{normal}) - corpy^{test}| \geq 2\delta$.
- **Normal:** otherwise.

This is because, at a distance of two standard deviations or more, the test sample is so far from the dispersion of normal correntropy values that it does not fit within the distribution of normal samples.

The absolute variation $|\mu(corpy^{normal}) - corpy^{test}|$ is normalized to obtain a risk level for each sample in the range $[0, 1]$. The normalized risk level RL of a (normal or test) sample is given by:

$$RL = \frac{|corpy^{normal/test} - min(corpy^{normal})|}{max(corpy^{normal}) - min(corpy^{normal})} \tag{6}$$

where $min(corpy^{normal})$ and $max(corpy^{normal})$ are the minimum and maximum correntropy values over all the normal samples, respectively. The risk levels, which specify the extent to which anomalous samples deviate from normal samples, enable the identification of attack samples with low false alarm rates.

The flow identifiers of attack samples are associated with their estimated risk levels. If the risk level has a value of one, then the attack poses the greatest risk to an organization because it sends many flows to a specific destination (e.g., a distributed denial-of-service attack). An attack sample with a risk level of zero poses the least risk to the organization. Table 1 provides information about selected samples in the UNSW-NB15 dataset along with their risk levels. Note that anomalous samples (i.e., attacks) have higher risk levels of at least 0.5 compared with normal samples that have risk levels less than 0.5.

Table 1. Selected samples with their risk levels.

Source IP Address	Source Port	Dest. IP Address	Dest. Port	Protocol	Label	Risk Level
149.171.126.14	179	175.45.176.3	33159	TCP	0	0.23
175.45.176.1	15982	149.171.126.14	5060	UDP	0	0.11
175.45.176.3	63888	149.171.126.14	179	TCP	0	0.25
175.45.176.2	7434	149.171.126.16	80	TCP	1	0.83
175.45.176.0	15558	149.171.126.13	179	TCP	1	0.72

4. Results and Discussion

This section presents the results and discusses their significance.

4.1 Dataset and Evaluation Metrics

The performance of the proposed scheme was evaluated using the UNSW-NB15 dataset because it comprises a large collection of contemporary legitimate and anomalous activities. The dataset contains approximately 100 GB of network packets and about 2,540,044 feature vectors maintained in four CSV files. Each vector contains 47 features and the class label. The data is categorized into ten classes, one class corresponding to normal activities and nine classes corresponding to security events and malware activities: (i) analysis; (ii) denial-of-service; (iii) exploits; (iv) fuzzers; (v) generic; (vi) reconnaissance; (vii) backdoors; (viii) shellcode; and (ix) worms.

The performance of the proposed scheme for identifying and tracking attacks was evaluated using two metrics: (i) accuracy; and (ii) false alarm rate (FAR):

- **Accuracy:** The accuracy is the percentage of legitimate and suspicious samples that are correctly identified:

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN} \times 100 \tag{7}$$

- **False Alarm Rate:** The false alarm rate is the percentage of normal and malicious samples that are incorrectly classified:

$$False\ Alarm\ Rate = \frac{FP + FN}{TP + TN + FP + FN} \times 100 \tag{8}$$

Note that TP and TN are the numbers of true positives and true negatives, respectively; and FP and FN are the numbers of false positives and false negatives, respectively.

Table 2. Features selected for investigating attacks.

Weight	Feature	Feature Description
0.592	sbytes	Source to destination bytes
0.558	swin	Source TCP window advertisement
0.552	dttl	Destination to source time to live
0.551	stcpb	Source TCP sequence number
0.550	dtcpb	Destination TCP sequence number
0.549	dwin	Destination TCP window advertisement
0.513	smean	Mean flow packet size transmitted by source
0.489	sload	Source bits per second

4.2 Pre-Processing and Feature Selection

The software used in the evaluation was developed using the R language and executed on a workstation with an i7 CPU and 16 GB RAM running the Windows 7 operating system. Three random samples, with 100,000, 200,000 and 300,000 items, were selected from the UNSW-NB15 dataset to extract important features using the chi-square statistic and investigate attack activities using the correntropy-variation technique. Table 2 lists the eight features that were selected based on their high chi-square values.

Feature vectors with five flow identifiers (i.e., source IP address, source port, destination IP address, destination port and protocol type) were selected using the uniform random sampling technique.

Table 1 shows five samples selected from the USNW-NB15 dataset, along with their computed risk levels. The risk levels were subsequently associated with their flow identifiers to analyze the evidence for attack activity. The chi-square statistic and uniform random sampling ensured the selection of features and samples that reflected legitimate and suspicious network activities.

4.3 Network Forensic Evaluation

Correntropy differentiates between legitimate and attack samples; this is because it estimates the nonlinear similarities between the samples. Figure 2 shows that the correntropy values of normal samples have fewer peaks than the correntropy values for attack samples. Thus, different attack types could be identified and investigated using the five flow identifiers associated with their risk levels.

Table 3 shows the results of evaluating the performance of the correntropy-variation technique. The evaluation metrics are the overall accu-

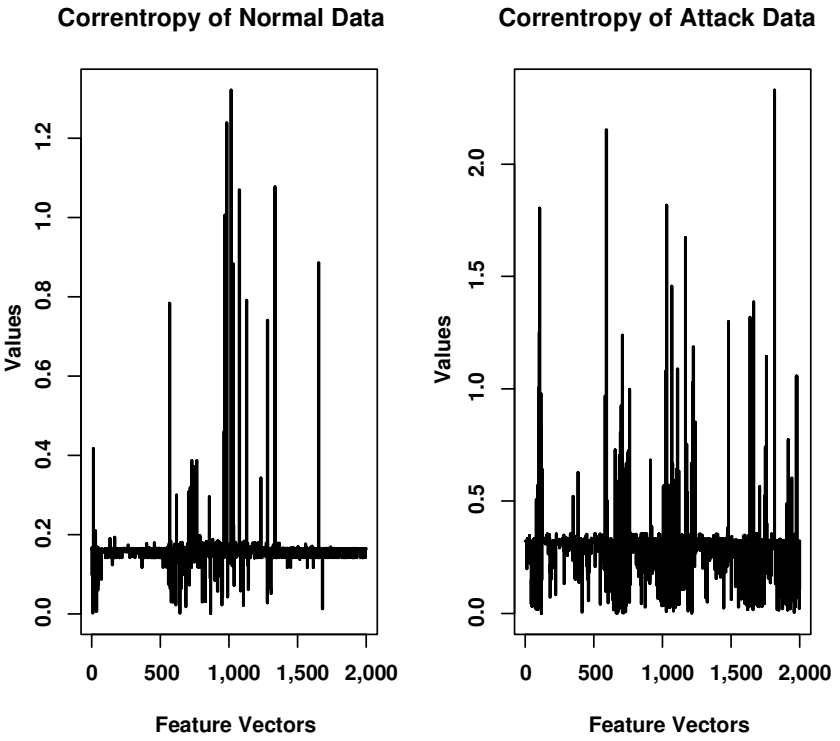

Figure 2. Correntropy of normal and attack samples.

Table 3. Performance of the correntropy-variation technique.

Sample Size	Accuracy	FAR
100,000	94.31%	5.69%
200,000	95.72%	4.28%
300,000	95.98%	4.02%

racy and false alarm rate for the features listed in Table 2. Note that the overall accuracy improves from 94.31% to 95.98% while the overall false alarm rate falls from 5.69% to 4.02% as the sample size increases from 100,000 to 300,000.

Table 4 shows that the proposed scheme recognizes different attack samples in the dataset. Specifically, the accuracy of detecting normal samples increases from 92.12% to 93.29% as the sample size increases from 100,000 to 300,000. Also, the accuracy of detecting malicious samples increases from a low of 45.82% for worms to a high of 97.55% for denial-of-service attacks.

Table 4. Comparison of accuracy values for three sample sizes.

Sample Types	Sample Size 100,000	Sample Size 200,000	Sample Size 300,000
Normal	92.12%	93.16%	93.29%
Analysis	88.26%	89.45%	90.22%
Denial-of-Service	95.71%	95.13%	97.55%
Exploits	76.47%	77.82%	77.19%
Fuzzers	64.33%	65.23%	66.28%
Generic	83.56%	87.52%	88.87%
Reconnaissance	58.38%	59.24%	60.32%
Backdoors	54.42%	71.23%	72.42%
Shellcode	65.76%	66.48%	65.98%
Worms	45.82%	45.92%	48.87%

The results obtained with the proposed correntropy-variation network forensic (CV-NF) technique are compared against those obtained with three state-of-the art approaches, namely the filter-based support vector machine (FSVM) technique [1], multivariate correlation analysis (MCA) technique [24] and artificial immune system (AIS) technique [20] using the UNSW-NB15 dataset. The results in Figure 3 reveal that the proposed technique outperforms the other three techniques in terms of accuracy as well as the false alarm rate.

The correntropy-variation network forensic technique produces the best results because it estimates the correntropy values for normal and test samples, and subsequently identifies samples that are more than two standard deviations away from the mean of normal samples as attacks. The filter-based support vector machine and artificial immune system techniques undergo training and validation with large numbers of normal and malicious samples. In contrast, the multivariate correlation analysis technique relies on correlations between attributes using a Gaussian mixture model; however, it may not precisely identify the boundaries between normal and malicious mixture models [19].

5. Conclusions

The network forensic scheme described in this chapter is designed for monitoring and investigating network-based attacks in real-time. The scheme involves three steps: capturing and storing network traffic data, selecting important traffic features and investigating anomalous traffic. The chi-square statistic and uniform random sampling are used to select sample features and samples, respectively; and the novel correntropy-variation technique is leveraged to identify samples with high risk levels

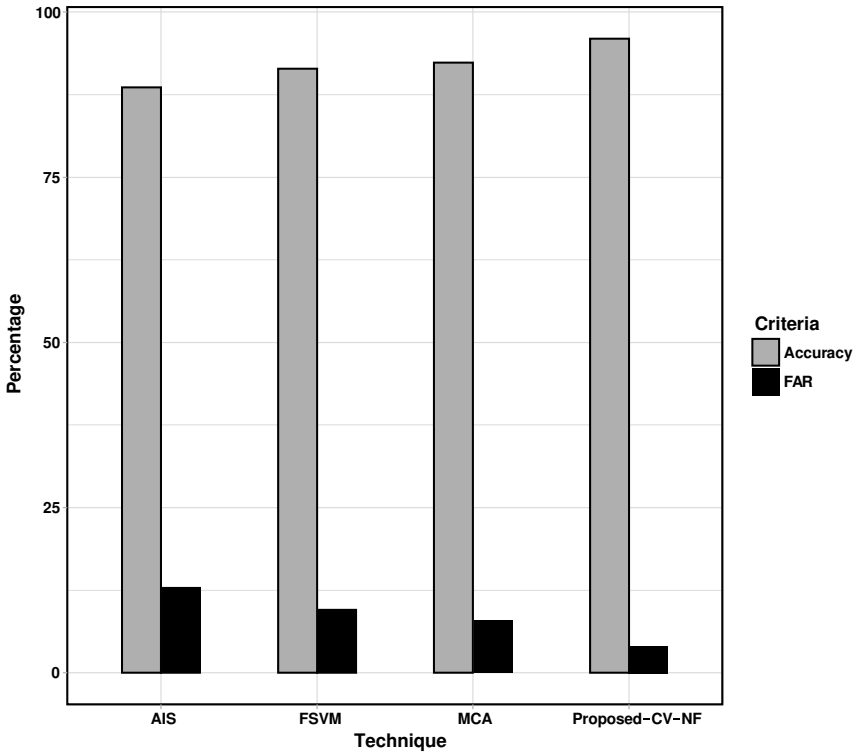

Figure 3. Comparison of the performance of the four techniques.

based on their flow identifiers (potential attacks). The case study employing the UNSW-NB15 dataset demonstrates the efficiency and efficacy of the network forensic scheme, especially its higher accuracy and lower false alarm rate compared with three state-of-the-art network attack detection approaches. Future research will attempt to re-target the proposed network forensic scheme for use in cloud and fog computing environments.

References

[1] M. Ambusaidi, X. He, P. Nanda and Z. Tan, Building an intrusion detection system using a filter-based feature selection algorithm, *IEEE Transactions on Computers*, vol. 65(10), pp. 2986–2998, 2016.

[2] R. Bao, H. Rong, P. Angelov, B. Chen and P. Wong, Correntropy-based evolving fuzzy neural system, to appear in *IEEE Transactions on Fuzzy Systems*.

[3] R. Brandom, A new ransomware attack is infecting airlines, banks and utilities across Europe, *The Verge*, June 27, 2017.

[4] L. Chen, D. Divakaran, A. Ang, W. Lim and V. Thing, FACT: A framework for authentication in cloud-based IP traceback, *IEEE Transactions on Information Forensics and Security*, vol. 12(3), pp. 604–616, 2017.

[5] Y. Chen and M. Chen, Using chi-square statistics to measure similarities for text categorization, *Expert Systems with Applications*, vol. 38(4), pp. 3085–3090, 2011.

[6] N. Clarke, F. Li and S. Furnell, A novel privacy preserving user identification approach for network traffic, *Computers and Security*, vol. 70, pp. 335–350, 2017.

[7] A. Diamah, M. Mohammadian and B. Balachandran, Network security evaluation method via attack graphs and fuzzy cognitive maps, *Proceedings of the Fourth International Conference on Intelligent Decision Technologies*, vol. 2, pp. 433–440, 2012.

[8] B. Hazarika and S. Medhi, Survey of real-time security mechanisms in network forensics, *International Journal of Computer Applications*, vol. 151(2), 2016.

[9] J. He, C. Chang, P. He and M. Pathan, Network forensic method based on evidence graph and vulnerability reasoning, *Future Internet*, vol. 8(4), article no. 9, 2016.

[10] M. Ibrahim, M. Abdullah and A. Dehghantanha, VoIP evidence model: A new forensic method for investigating VoIP malicious attacks, *Proceedings of the International Conference on Cyber Security, Cyber Warfare and Digital Forensics*, pp. 201–206, 2012.

[11] S. Khan, A. Ghani, A. Wahab, M. Shiraz and I. Ahmad, Network forensics: Review, taxonomy and open challenges, *Journal of Network and Computer Applications*, vol. 66, pp. 214–235, 2016.

[12] S. Khan, M. Shiraz, A. Wahab, A. Ghani, Q. Han and Z. Rahman, A comprehensive review of the adaptability of network forensic frameworks for mobile cloud computing, *The Scientific World Journal*, vol. 2014, article id. 547062, 2014.

[13] Y. Li, Y. Wang, F. Yang, S. Su and D. Yan, Deterministic packet marking based on the coordination of border gateways, *Proceedings of the Second International Conference on Education Technology and Computers*, vol. 2, pp. 154–161, 2010.

[14] C. Liu, A. Singhal and D. Wijesekera, A probabilistic network forensic model for evidence analysis, in *Advances in Digital Forensics XII*, G. Peterson and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 189–210, 2016.

[15] H. Liu and H. Motoda, *Computational Methods of Feature Selection*, Chapman and Hall/CRC, Boca Raton, Florida, 2008.

[16] J. Liu, G. Tian and S. Zhu, Design and implementation of a network forensic system based on intrusion detection analysis, *Proceedings of the International Conference on Control Engineering and Communications Technology*, pp. 689–692, 2012.

[17] W. Liu, P. Pokharel and J. Principe, Correntropy: Properties and applications in non-Gaussian signal processing, *IEEE Transactions on Signal Processing*, vol. 55(11), pp. 5286–5298, 2007.

[18] N. Moustafa and J. Slay, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 Network Data Set), *Proceedings of the Military Communications and Information Systems Conference*, 2015.

[19] N. Moustafa, J. Slay and G. Creech, Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation in large-scale networks, to appear in *IEEE Transactions on Big Data*.

[20] P. Saurabh and B. Verma, An efficient proactive artificial immune system based anomaly detection and prevention system, *Expert Systems with Applications*, vol. 60, pp. 311–320, 2016.

[21] A. Shalaginov and K. Franke, Big data analytics by automated generation of fuzzy rules for network forensic readiness, *Applied Soft Computing*, vol. 52, pp. 359–375, 2017.

[22] M. Srinivas and A. Sung, Identifying significant features for network forensic analysis using artificial intelligence techniques, *International Journal of Digital Evidence*, vol. 1(4), 2003.

[23] T. Tafazzoli, E. Salahi and H. Gharaee, A proposed architecture for network forensic systems in large-scale networks, *International Journal of Computer Networks and Communications*, vol. 7(4), pp. 43–56, 2015.

[24] Z. Tan, A. Jamdagni, X. He, P. Nanda and R. Liu, A system for denial-of-service attack detection based on multivariate correlation analysis, *IEEE Transactions on Parallel and Distributed Systems*, vol. 25(2), pp. 447–456, 2014.

[25] S. Thompson, *Sampling*, John Wiley and Sons, Hoboken, New Jersey, 2012.

[26] K. Wang, M. Du, Y. Sun, A. Vinel and Y. Zhang, Attack detection and distributed forensics in machine-to-machine networks, *IEEE Network*, vol. 30(6), pp. 49–55, 2016.

[27] X. Wang and X. Wang, Topology-assisted deterministic packet marking for IP traceback, *Journal of China Universities of Posts and Telecommunications*, vol. 17(2), pp. 116–121, 2010.