Anomaly Detection in Wireless Community Networks using PCA

Llorenç Cerdà-Alabern[1] and Gabriel Iuhasz[2]

[1] Universitat Politècnica de Catalunya, Barcelona, Spain
[2] West University, Timisoara, Romania
llorenc@ac.upc.edu, iuhasz.gabriel@e-uvt.ro

Abstract. In this paper we analyze anomaly detection in wireless community networks using Principal Component Analysis, PCA. In contrast to other works found in the literature, we have produced a dataset from a production network, gathering traffic and non-traffic features. We have investigated the ability of PCA to detect the failure of a gateway in the mesh. Our numerical results show that PCA is effective to detect the failure, but traffic spikes produced by irregular traffic patterns caused by users usage may be also marked as anomalies.

Keywords: anomaly detection · principal component analysis · wireless community networks.

1 Introduction

Anomaly detection (AD) is common term in computer networks for the problem of identifying deviations from the expected behavior [11]. Detecting such deviations is of interest for multiple reasons. A basic one is to predict network misbehavior that might end up in device failures. This is specially important in Wireless Community Networks (WCN). WCN are built by their own users installing wireless antennas on top of their houses. In WCN users do not only build the network infrastructure, but associate to constitute micro ISPs, providing access to the Internet and internal services managed by the community. WCN have proliferated, due to high throughput and cheap off-the-shelf wireless devices. One outstanding example is Guifi.net. Guifi.net started as a WCN in 2004, and at the time of writing it reports 36.886 working nodes [12]. Guifi.net has become a complex network where associations of self-providing users coexist with commercial network operators [9].

In this paper we analyze a dataset gathered from a production WCN which is part of Guifi.net. The dataset we have produced contains metrics from the linux kernel that reflect the state of the CPU, memory and network interfaces. Such metrics are typically gathered by network monitoring tools as Munin [2] and Nagios [3]. These tools, however, simply produce time series graphics of the gathered metrics, and it is left to the network administrator its analysis and interpretation. In this paper we will use Principal Component Analysis (PCA) for the metric analysis. Our goal is to use these metrics for anomaly detection.

2 Related Work

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior [11]. In statistics the term outliers is often used for this concept. The importance of anomaly detection is due to the fact that anomalies in data might reveal critical misbehavior in a wide number of scenarios. For instance, fraud detection for credit cards, insurance or health care, damage detection, etc. The significance of this topic has motivated a large number of articles, surveys and even books. For instance, the surveys [11] and [15] provide an extensive review of detection techniques developed in a wide number of domains.

In computer networks most work on anomaly detection has traditionally focused on the problem of security. In this context the term *intrusion detection systems*, IDS, if often used. IDS typically exploit anomalies in network traffic to detect a wide number of security attacks, as denial of service, DoS, reconnaissance, etc. See e.g. [20] for a description of different types of security attacks. An overview of IDS proposed in the literature can be found in [25].

In [18] the PCA analysis was introduced to perform a structural analysis of network traffic. Their proposal received a significant attention for IDS. For instance, in [27, 26] refinements to PCA analysis are done to detect spikes in traffic flows, which potentially identify network attacks. In [22] generate port-scans and snapshots attacks in a lab, merged with residential traffic. Attacks detection is done using a robust PCA approach. PCA to detect network traffic anomalies has also been criticized by a number of papers [24]. In [17] it is analyzed the weaknesses of PCA, and use *Commute Distance* to detect DoS attacks using traffic measures. In [7] it is argued that problems with PCA are due to flaws in its adoption, and appropriate ways to apply PCA are proposed.

3 Dataset

In this paper we will focus on a WCN called GuifiSants [14]. GuifiSants started in 2009 and is part of Guifi.net. The nodes of GuifiSants consists of antennas flashed with a linux Openwrt distribution [21] running the BMX6 mesh routing protocol [6, 8]. GuifiSants is a WCN deployed in a neighborhood of the city of Barcelona (Spain) called Sants. In 2012 *Universitat Politècnica de Catalunya* (UPC) joined GuifiSants for research purposes. At the time of writing there are around 60 nodes in GuifiSants. GuifiSants is a production network with some tens of users having this network as their only access to the Internet. In [13] a live monitoring web page of GuifiSants updated hourly is available. A technical analysis of GuifiSants can be found in [10].

The dataset has been built from data samples gathered from each node every 5-minutes. This is done using a permanent ssh connection from one central monitoring server to each node in the mesh, which is used to run standard system commands. The dump of these commands is then parsed to obtain the data. This method has the advantage that no changes or additional software need to

be installed in the nodes. This is an important condition, since the users are the owner of their nodes. So, only permission to install a public key to access the node with ssh for monitoring purposes is needed from the users.

The data is obtained reading the linux kernel variables available through the `/proc` file-system. For instance, `/proc/net/dev` to read counters with the number of bytes and packets transmitted and received over each interface; `/proc/stat` where there is information about kernel activity; `/proc/meminfo` for memory usage, etc. Kernel variables are of two types: (i) absolute values, for instance, the CPU 1-minute load average, and (ii) counters that are monotonically increased, for instance the number of transmitted packets. We have converted counter-type kernel variables to rates, by dividing the difference between two consecutive samples, over the difference of the corresponding timestamps in seconds. We have removed negative rate samples that occur when a node is rebooted, or when a counter reaches its maximum value and it is restarted.

4 PCA fundamentals

Let $\{x_i\}_{i=1}^m$ be a set of m samples or measurements of n features. Thorough the paper we will assume column vectors. Thus, x_i are column vectors of n components. Samples are organized in a matrix X of dimension $m \times n$, where each row is a sample and each column a feature:

$$X = \begin{bmatrix} x_1 \cdots x_m \end{bmatrix}^T \tag{1}$$

The idea of PCA is approximating the samples of n features by a projection on a lower dimensional space of dimension $l < n$:

$$t_i = P^T x_i \tag{2}$$

where the matrix P has dimension $n \times l$, and

$$P = \begin{bmatrix} u_1 \cdots u_l \end{bmatrix} \tag{3}$$

with u_i orthogonal unitary vectors of n components:

$$u_i^T u_j = \delta_{ij}, \quad \|u_i\| = 1,\ \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

The orthonormal vectors $\{u_i\}_{i=1}^l$ are chosen to maximize the variance of the projected samples $\{t_i\}_{i=1}^m$. This maximization condition yields (see e.g. [5]):

$$S\, u_i = \lambda_i\, u_i \tag{4}$$

where S is the samples' covariance matrix: $S = \frac{1}{m-1} X^T X$. Thus, $\{u_i\}_{i=1}^l$ are given by the right eigenvectors of the largest l eigenvalues of S.

PCA is a standard methodology applied to Statistical Process Control, SPC [7]. In this context it is first used a calibration set of samples, aka training set, to

compute the projection matrix P. Upon a new observation x_j, it is computed the *score*:

$$t_j = P^T x_j \tag{5}$$

with residual:

$$e_j = x_j - P\, t_j \tag{6}$$

4.1 Anomaly Detection

In order to detect anomalies of a new score t_j, the most popular indices used in SPC are the Hotelling's T^2 also known as the D statistic, and the squared prediction error (SPE), which is also known as the Q statistic [7, 4]:

$$D_j = \sum_{i=1}^{l} \left(\frac{t_{j,i} - \mu_i}{\sigma_i} \right) \tag{7}$$

$$Q_j = \sum_{i=1}^{n} e_{j,i}^2 \tag{8}$$

where $t_{j,i}$ is the i component of score t_j, $\{\mu_i, \sigma_i\}_{i=1}^{l}$ are the scores' i component mean and standard deviation of the calibration samples, and $e_{j,i}$ the features' i residual of sample x_j.

Assuming that the scores are approximately Normal distributed, the statistic at significance level α of new incoming data are given by [7, 16]:

$$D_\alpha = \frac{l\,(m^2-1)}{m\,(m-l)}\, F_{l,(m-l),\alpha} \tag{9}$$

$$Q_\alpha = \theta_1 \left[\frac{z_\alpha \sqrt{2\,\theta_2\, h_0^2}}{\theta_1} + 1 + \frac{\theta_2\, h_0\, (h_0 - 1)}{\theta_1^2} \right]^{1/h_0} \tag{10}$$

where F is the F distribution, $\theta_i = \sum_{j=l+1}^{\mathrm{rank}(X)} \lambda_j^i$, being λ_j the eigenvalues of the samples' covariance matrix S sorted in decreasing order; $h_0 = 1 - \frac{2\,\theta_1\,\theta_3}{3\,\theta_2^2}$; and z_α is the upper $(1-\alpha)$ standardized normal percentile.

4.2 Contribution Plots

Once an anomaly is detected, a diagnosis system is needed to determine its root-causes. The general approach in SPC is using contribution plots [19]. The idea of such plot is estimating the contribution of each observed feature of a sample to a particular statistic value considered as anomalous. There have been proposed several methods to build contribution plots [4]. In this paper we shall use complete decomposition contributions (CDC). CDC are easily computed and show a good diagnosis performance in a networking monitoring system [7]. Let

$x_{j,i}$ be the feature i of an anomalous sample x_j. Then, the CDC for the D and Q statistics defined above is given by [4]:

$$CDC_i^{D_j} = (\xi_i \, P \, \Lambda^{-1/2} \, P^T x_j)^2 \tag{11}$$

$$CDC_i^{Q_j} = e_{j,i}^2 \tag{12}$$

where ξ_i is the i column of the identity matrix, $\xi_i = \begin{bmatrix} 0 \cdots & 1 & \cdots 0 \end{bmatrix}^T$; P and Λ are the projection matrix and eigenvalues of the covariance matrix S, respectively (see (4)); and $e_{j,i}$ is features' i residual of sample x_j.

5 Numerical Results

As described in section 3, the dataset consists of passive node features gathered at 5 min intervals. Features include traffic and non traffic features. Traffic features are obtained from the counters available at `/proc/net/dev` linux /proc file-system. For each node we have considered the counters of received and transmitted bytes and packets over Ethernet and WiFi interfaces.

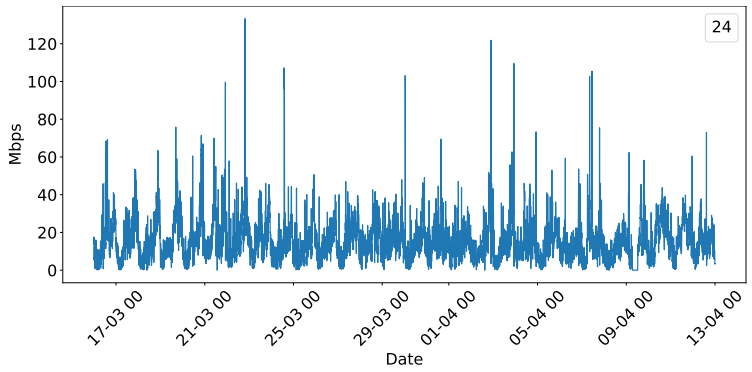

Fig. 1. Plot of the `sum.xb.rate-24` traffic feature (sum of received and transmitted traffic over node 24) during the 4 weeks of the training set. Peaks and valleys show daily activity and inactivity periods. Dates in the x-axis are formatted as day-month hour.

For each node we have also considered the sum of counter values for both types of interfaces (Ethernet and WiFi), and the sum and difference of received and transmitted bytes and packets over all interfaces. Recall that all counter features are then converted into rates, as explained in section 3. For instance, features `eth.txb.rate-24` and `sum.xb.rate-24` refer to the rates of transmitted bytes over Ethernet interfaces, and total number of received and transmitted bytes, respectively, in node 24 (see Fig.1).

Non traffic features include the number of processes, CPU load average, softirq, iowait, time processes executing in kernel and user mode, context switches,

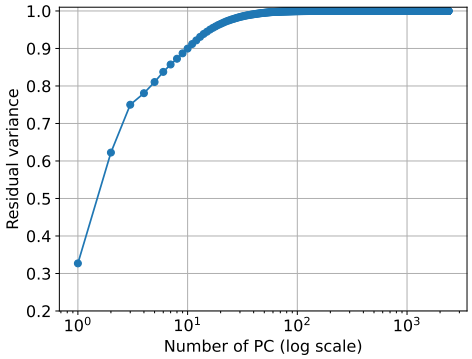

Fig. 2. Normalized residual variance vs number of principal components.

etc. A complete description of the dataset features can be found in [1]. Some memory related features where discarded because of their erratic behavior. These include the counters from `/proc/meminfo` and `/proc/vmstat` related with the cache, committed, active, buffers, mapped and vmalloc.

5.1 Methodology

Overall there where 63 nodes gathered in the dataset, with a total of 2387 features. In order to assess the efficacy of PCA to detect anomalies we have proceed as follows. On April the 14th 2021 one of the two gateways of the mesh (node 24) broke down and was replaced. Due to the node failure, samples from this node where missing between 01:55 and 17:40 of April the 14th. For the testing set we have used the samples gathered during an interval of 3 days when the failure occurs: April the 13, 14 and 15th. For the training set we have used the samples gathered during the 4 weeks previous to the testing set. Fig.1 show the `sum.xb.rate-24` traffic feature of node 24 during the training set.

5.2 PCA Analysis

We have carried out the PCA analysis using the python sklearn library [23]. In order to fit PCA, missing values have been set to 0. Missing values occur when a data cannot be gathered from a node, typically because is down. In order to assess the number of PCA components we have used the common methodology consisting on choosing a number of dimensions that preserve 95% of residual variance (see Fig.2). In our case we obtained that this condition is achieved for a number of PC equal to 16. So, we have fit PCA using the training set described above with a number of PCs equal to 16.

Once we have fitted PCA, we have computed the D and Q statistics (equations (7), (8)) over the testing set. Their values are shown in Fig.3. The horizontal line corresponds to a 0.99 significance level (equations (9), (10)). Values of D and Q surpassing the significance level are considered potentially anomalous.

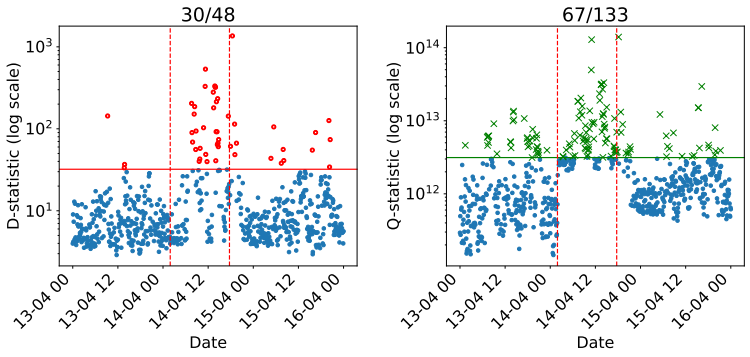

Fig. 3. D and Q statistics. The horizontal line corresponds to a 0.99 significance level. The vertical lines correspond to the interval of node 24 failure.

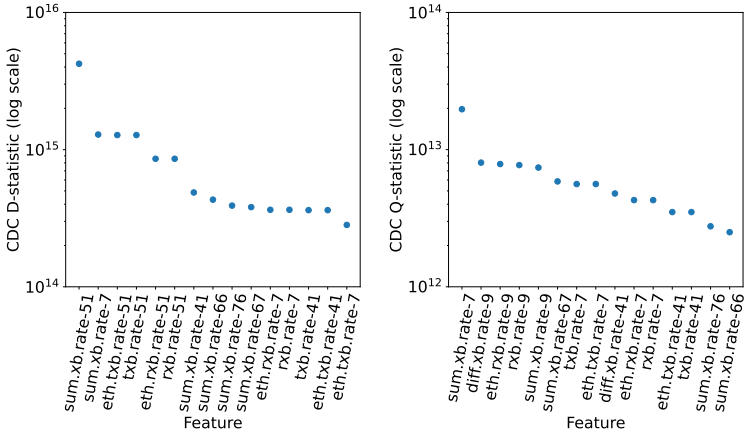

Fig. 4. Complete decomposition contribution, CDC, of the 15 features having the maximum contribution in the D and Q-statistics, sorted in decreasing order.

The vertical lines correspond to the interval of node 24 failure. Ideally it would be expected that most anomalous points fall inside this interval. Indeed, from the 3 days of the testing set, 30 out of 48 and 67 out of 133 anomalies of the D and Q statistics, respectively, fall inside the 16 hours that, approximately, last the failure of node 24. Node 24 is the main gateway of the mesh, but, being 2 gateways, the routing protocol directed the traffic to the second gateway during the failure, and the mesh continued having Internet connectivity.

Fig.4 shows the CDC plot of the sample having the maximum D and Q-statistic value inside the interval of node 24 failure. The plot shows the 15 features having the maximum contribution. It can be noticed that all the plotted features are traffic related. This is an expected result, since traffic features have large absolute values, compared with other non-traffic features. One could consider the possibility of scaling all features to have a unit standard deviation. This

is a typical approach when all the features are of different type, and the same weight is desired for all of them. However, in our scenario nodes having the largest amount of traffic are core nodes in the mesh, thus, having most significant behavior. Therefore, it is convenient not to scale the traffic features.

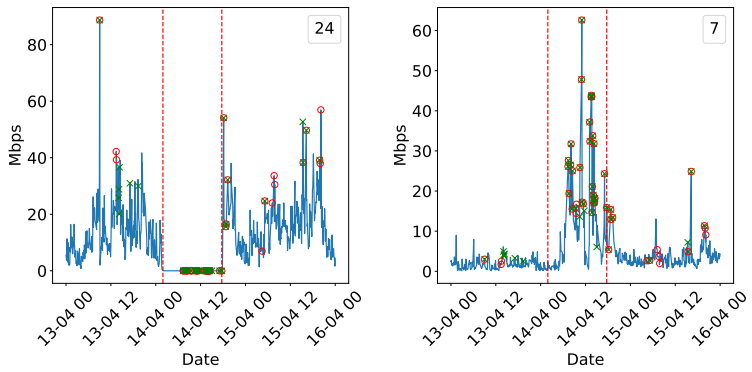

Fig. 5. Total traffic in node 24 (failed node) and node 7 (contributing the most to Q-statistic sample) during the testing set. Vertical lines correspond to the failure interval. Marked samples correspond to D and Q-statistic with values higher than the 0.99 significant level.

It can be observed in Fig.4 that the feature having the maximum contribution in the Q-statistic is `sum.xb.rate-7`, that is, the total amount of traffic of node 7. Fig.5 shows the total traffic of the failed node, node 24, and node 7. The samples where the D and Q-statistic have values higher than the 0.99 significant level are also marked. It can be observed that during the node 24 failure the rate is 0, since no samples of this node are available during failure. On the other hand, rate of node 7 significantly increases during the failure interval. This is because node 7 is close to the second gateway of the mesh. This makes that upon node 24 failure the traffic that would have been routed towards node 24 is rerouted towards node 7.

We can conclude that the failure of the gateway (node 24) causes a redistribution of traffic in the mesh that produces a significant increase of traffic in some nodes. This fact is well captured by the PCA analysis, where the amount of anomalies detected in the interval where node 24 fails is remarkably higher. Fig. 5 also shows that outside the failure interval there are also traffic spikes that makes the D and Q-statistic to have values higher than the 0.99 significant level. Most of these spikes are not anomalies, but consequence of the users usage of the network. For instance, when a user downloads a large size web page, or a large file. Therefore, we can conclude too that the simple PCA analysis carried out in this paper can easily produce false negatives due to the irregular traffic pattern of users traffic.

6 Conclusion

In this paper we have carried out a PCA analysis of a wireless community network. In contrast to other works found in the literature, we have built a dataset from a production network, gathering samples every 5 minutes intervals. Data samples do not only have traffic features, but non-traffic ones as CPU usage, memory usage, etc.

In our analysis we have studied the ability of PCA to detect the failure of a main gateway of the mesh. We have used a 4 weeks period for the training set, and a 3 days period for the testing set. During 16h in the middle of the testing set the failure of the gateway occurs. We have found that the redistribution of traffic in the mesh that occurs during the gateway failure is well captured by the PCA analysis, where the number of anomalies detected by PCA significantly increases. Outside the failure interval a number of traffic spikes are also marked as anomalies. This spikes, however, are false negatives, since they are consequence of irregular traffic patterns of users traffic.

We conclude that a simple PCA approach can be an efficient method to detect irregular changes in a set of features that can be gathered in a wireless mesh network. Therefore, it can be an effective approach to anomaly detection in such networks. However, it can also easily produce false negative, e.g. due to traffic spikes produced by irregular traffic patterns caused by users usage.

Acknowledgment

This project has received funding through the CHIST-ERA Consortium of European Funding Agencies, including: Engineering and Physical Sciences Research Council (EPSRC) under grant agreement EP/T022345/1; Spanish grant PID2019-106774RB-C21 and Generalitat de Catalunya through 2017-SGR-990.

References

1. Description of the dataset used in the DIPET project use case anomaly detection. http://mortitx.pc.ac.upc.edu/ llorenc/read-me-csv-ede.html, online; accessed August-2021
2. Munin networked resource monitoring tool. http://munin-monitoring.org
3. Nagios, The Industry Standard In IT Infrastructure Monitoring. https://www.nagios.org
4. Alcala, C.F., Qin, S.J.: Analysis and generalization of fault diagnosis methods for process monitoring. Journal of Process Control **21**(3), 322–330 (2011)
5. Bishop, C.M.: Machine learning and pattern recognition. Information science and statistics. Springer, Heidelberg (2006)
6. BMX6 mesh networking protocol. http://bmx6.net, online; accessed January-2021
7. Camacho, J., Pérez-Villegas, A., García-Teodoro, P., Maciá-Fernández, G.: Pca-based multivariate statistical network monitoring for anomaly detection. Computers & Security **59**, 118–137 (2016)

8. Cerdà-Alabern, L., Neumann, A., Maccari, L.: Experimental evaluation of bmx6 routing metrics in a 802.11an wireless-community mesh network. In: 2015 3rd International Conference on Future Internet of Things and Cloud. pp. 770–775 (2015)
9. Cerdà-Alabern, L., Baig, R., Navarro, L.: On the guifi.net community network economics. Computer Networks **168**, 107067 (2020)
10. Cerdà-Alabern, L., Neumann, A., Escrich, P.: Experimental evaluation of a wireless community mesh network. In: The 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM'13. ACM, Barcelona, Spain (Nov 3–8, 2013)
11. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)
12. Guifi.net: Open, Free and Neutral Network Internet for everybody. http://guifi.net/en, online; accessed 13-January-2021
13. GuifiSants: qMp Sants-UPC. http://dsg.ac.upc.edu/qmpsu/index.php/, online; accessed January-2021
14. GuifiSants: Xarxa oberta, lliure i neutral del barri de sants. http://sants.guifi.net/, online; accessed January-2021
15. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial intelligence review **22**(2), 85–126 (2004)
16. Jackson, J.E., Mudholkar, G.S.: Control procedures for residuals associated with principal component analysis. Technometrics **21**(3), 341–349 (1979)
17. Khoa, N.L.D., Babaie, T., Chawla, S., Zaidi, Z.: Network anomaly detection using a commute distance based approach. In: 2010 IEEE International Conference on Data Mining Workshops. pp. 943–950. IEEE (2010)
18. Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E.D., Taft, N.: Structural analysis of network traffic flows. In: Proceedings of the joint international conference on Measurement and modeling of computer systems. pp. 61–72 (2004)
19. Miller, P., Swanson, R.E., Heckler, C.E.: Contribution plots: a missing link in multivariate quality control. Applied mathematics and computer science **8**(4), 775–792 (1998)
20. Northcutt, S., Novak, J.: Network intrusion detection. Sams Publishing (2002)
21. OpenWrt Project: OpenWrt Project: Welcome to the OpenWrt Project. https://openwrt.org/, online; accessed January-2021
22. Pascoal, C., De Oliveira, M.R., Valadas, R., Filzmoser, P., Salvador, P., Pacheco, A.: Robust feature selection and robust pca for internet traffic anomaly detection. In: 2012 Proceedings Ieee Infocom. pp. 1755–1763. IEEE (2012)
23. Pedregosa, F., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
24. Ringberg, H., Soule, A., Rexford, J., Diot, C.: Sensitivity of pca for traffic anomaly detection. In: Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems. pp. 109–120 (2007)
25. Tidjon, L.N., Frappier, M., Mammar, A.: Intrusion detection systems: A cross-domain overview. IEEE Communications Surveys Tutorials **21**(4) (2019)
26. Zaidi, Z.R., Hakami, S., Landfeldt, B., Moors, T.: Real-time detection of traffic anomalies in wireless mesh networks. Wireless Networks **16**(6), 1675–1689 (2010)
27. Zaidi, Z.R., Hakami, S., Moors, T., Landfeldt, B.: Detection and identification of anomalies in wireless mesh networks using principal component analysis (pca). Journal of Interconnection Networks **10**(04), 517–534 (2009)