Networking Anomaly Detection using DSNS and Particle Swarm Optimization with Re-Clustering


*Computing Science Department, State University of Londrina (UEL), Londrina, Brazil
†School of Elect. & Comp. Engineering, University of Campinas (UNICAMP), Campinas, Brazil
‡Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal
E-mails: {moisesflima, lucas.dias.sampaio, brunozarpelao}@gmail.com, joeljn@ieee.org, {taufik, proenca}@uel.br

Abstract—This paper presents an anomaly detection method using Digital Signature of Network Segment (DSNS) and Particle Swarm Optimization-based clustering (PSO-Cls). The PSO algorithm is an evolutionary computation technique whose main characteristics include low computational complexity, ability to escape from local optima, and small number of input parameters dependence, when compared to other evolutionary algorithms, e.g. genetic algorithms (GA). In the PSO-Cls algorithm, swarm intelligence is combined with K-means clustering, in order to achieve high convergence rates. On the other hand, DSNS consists of normal network traffic behavior profiles, generated by the application of Baseline for Automatic Backbone Management (BLGBA) model in SNMP historical network data set. The proposed approach identifies and classifies data clusters from DSNS and real traffic, using swarm intelligence. Anomalous behaviors can be easily identified by comparing real traffic and cluster centroids. Tests were performed in the network of State University of Londrina and the obtained detection and false alarm rates are promising.

Index Terms—Anomaly detection, PSO, Baseline, DSNS, K-means clustering algorithm.

I. INTRODUCTION

Continuous advances in communication technology have driven the creation of a wide services variety on the communication networks. However, maintain the proper functioning of these networks has become a difficult task due to the large traffic volume carried. Thus, automating management tasks such as anomaly detection are essential to ensure information security and resources availability. Network anomalies impact on the quality of services provided, resulting in degradation of network performance and even in the interruption of its operations [1] [2].

Historically, methods for detecting anomalies are divided into two categories. Signature-based anomaly detection systems search for previously known anomaly patterns in network traffic. On the other hand, profile-based systems create models that represent the normal behavior of network. Anomalies are detected when real traffic deviates from the expected behavior. Despite usually presenting high false alarm rates, methods based on normal profiles are more promising due to their flexibility and the ability to detect new types of anomalies. The main techniques used in profile-based systems are: statistical models, data-mining and machine learning [3].

In this work, we propose a data mining based anomaly detection system, aiming to detect volume anomalies, using Simple Network Management Protocol (SNMP) monitoring. The method is novel in terms of combining the use of Digital Signature of Network Segment (DSNS) [4] with the evolutionary technique called Particle Swarm Optimization (PSO) [5] and K-means clustering algorithm [6], applied in a real data set.

PSO is a high efficient heuristic technique with low computational complexity, developed in 1995 by Kennedy and Eberhart [5] inspired by social behavior of bird flocking. The DSNS is a baseline that consists of different normal behavior profiles to a specific network device or segment, generated by the GBA tool (Automatic Backbone Management) [4], using data collected from SNMP objects. The proposed anomaly detection system uses the K-means algorithm in order to clusterize the traffic collected by SNMP agents and its respective DSNS. The PSO is combined with the K-means in order to improve performance and quality of the solution in the clusterization and calculation of clusters centroids.

Tests were carried out using a real network environment in the State University of Londrina (UEL), Brazil. Numerical results have been shown that the obtained detection and false alarm rates are promising. We also implemented the deterministic method proposed in [2] in order to detect anomalies on the same dataset, so that both methods could be compared.

This paper is organized as follows. The Section II presents related works on the network anomalies. The traffic model characterization is detailed in Section III. Section IV describes the swarm optimization aspects. Section V details the proposed anomaly detection approach, while Section VI discusses the adopted tests setup and the respective performance results. Finally, the main conclusions and future work are offered in Section VII.

II. RELATED WORK

Several studies have been conducted in order to propose efficient approaches for detection and classification of anomalies. The establishment of a normal model and the need of increasing anomaly detection rate with lower false alarm rate are still challenging tasks. In recent years, several works such as [7]–[9]
were developed in the area of anomaly detection. Though using different approaches, they have the same goal of maximizing the detection rate while minimizing the rate of false alarms.

Ensafi et al. [10] proposed the use of K-means as an anomaly detection method. In order to address the local convergence problem and high false alarm rate originated by K-means algorithm, the authors combined two techniques of soft computing: fuzzy logic and swarm intelligence. The proposed method consists of two phases: training phase, which the better particle throughout the generations is able to be found, and the detection phase, which uses the Euclidean distance between the cluster centroid and the input data to verify if a real traffic point is normal or anomalous. Tests were conducted using the classical training dataset KDDCup99 [11]. Results indicated that the method presents some capability for detecting anomalies, despite revealing high false alarm rates.

In [12], Ma et al. propose a Radial Basis Function neural network (RBFNN) for network anomaly detection. The method consists in embed a radial basis function in a two-layer feed-forward neural network. It is very important to specify a strong training algorithm, in order to find the neural network parameters. Hence, a swarm intelligence technique, named Quantum-Behaved Particle Swarm Optimization (QPSO), is employed to train the RBF network. In order to improve the ability of QPSO to escape local optima, the authors associated the Quantum PSO with gradient descent (GD) algorithm forming the QPSO-GD algorithm. Aiming to validate the proposed approach, an experiment was conducted using the KDDCup99 training dataset. Results showed that the hybrid QPSO-GD-RBFNN is a stronger optimizer than QPSO and GD algorithm implemented separately for training the RBFNN.

Zarpelão et al. [2] proposed a normal profile-based anomaly detection system, using data collected from SNMP objects for volume anomaly detection. The method applies simple parameterized deterministic algorithms over data collected from SNMP objects and their normal behavior profiles generated by the GBA tool (Automatic Backbone Management) [4]. Behavior deviations detected in different SNMP objects are correlated by using a dependency graph, which represents the relationships between the monitored objects. Satisfactory results were achieved in the experiments performed on the UEL network. In many test cases, detection and false alarm rates were better than the goals required by the network administrator. Hence, in this work we present a performance comparison between our proposal and the deterministic algorithm suggested in [2].

III. TRAFFIC CHARACTERIZATION: BLGBA AND DSNS

The first step to detect anomalies is to adopt a model that characterizes the network traffic efficiently, which represents a significant challenge due to the non-stationary nature of network traffic. Large networks traffic behavior is composed by daily cycles, where traffic levels are usually higher in working hours and are also distinct for workdays and weekends. So an efficient traffic characterization model should be able to trustworthily represent these characteristics. Thus, the GBA tool is used to generate different profiles of normal behavior for each day of the week, meeting this requirement. These behavior profiles are named Digital Signature of Network Segment (DSNS), proposed by Proença in [4] and applied to anomaly detection with great results in [2].

Hence, the BLGBA algorithm was developed based on a variation in the calculation of statistical mode. In order to determine an expected value to a given second of the day, the model analyzes the values for the same second in previous weeks. These values are distributed in frequencies, based on the difference between the greatest $G_{aj}$ and the smallest $S_{aj}$ element of the sample, using 5 classes. This difference, divided by five, forms the amplitude $h$ between the classes, $h = (G_{aj} - S_{aj})/5$. Then, the limits of each $L_{ck}$ class are obtained. They are calculated by $L_{ck} = S_{aj} + h * k$, where $C_k$ represents the $k$ class ($k = 1 \ldots 5$). The value that is the greatest element inserted in the class with accumulated frequency equal or greater than 80% is included in DSNS.

The samples for the generation of DSNS are collected second by second along the day, by the GBA tool. Two types of DSNS are generated: the bl-7 consisting of one DSNS for each day of the week, and the bl-3 consisting of one DSNS for the workdays, one for Saturday and another one for Sunday.

Figure 1 shows charts containing one week of monitoring of UEL network. Data were collected from SNMP object ipInReceives, at the University’s Web server in the period of 04/20/2009 to 04/26/2009. The data collected are represented in green and the respective DSNS values by the blue line. The charts show that traffic has a periodic behavior, where traffic levels are higher during the working hours, from 8 a.m. to 6 p.m. It is possible to observe a great adjustment between the behavior of real traffic and the DSNS.

IV. PARTICLE SWARM OPTIMIZATION AND K-MEANS CLUSTERING

The PSO is an evolutionary computation technique created by Kennedy and Eberhart [5] in 1995, based on birds social behavior. PSO is powerful since it is able to escape from global optima while keeps a simple structure. Unlike GA algorithm, PSO does not use operators like recombination and mutation, which contributes to the complexity reduction, but not reduces its efficiency. On the other hand, K-means is a clustering algorithm used in solving the well known clustering problem and it is classified as a method of unsupervised learning, aiming to classify a dataset into clusters, so that each data point is classified as belonging to the cluster with the closer mean [13]. While K-means is classified as a simple algorithm, it suffers from the absence of diversity mechanism to escape from local optimum. In order to overcome this drawback and simultaneously keeps computational complexity under control, mainly because for high-dimensional problems complexity is a concern, the K-means algorithm can be associated to PSO [13], resulting in the PSO-CIs algorithm.

In PSO, the solutions in the search space are called particles. Each particle has a fitness value, which is measured by the function to be optimized, and a speed that drives its flight, moving through space in search of the problem’ best solution.
The PSO principle is the movement of a group of particles, randomly distributed in the search space, each one with its own position and velocity. The position of each particle is modified by the application of velocity in order to reach a better performance [5]. The interaction among particles is inserted in the calculation of particle velocity. Hence, at each iteration, the speed and position of all particles from a population of size $M$ are updated. If the best values for local or global solutions were founded, the respective best candidate-vector is updated, where $p_{\text{best}}^i$ is the best value obtained so far by each particle in the population of size $M$, and $p_{\text{best}}^g$ is the best value obtained by all particle so far. The best local and global particles are column-vectors wise, with dimension $D$.

In the PSO strategy, each candidate-vector at $n$th iteration, defined as $p_i[n]$ with $D \times 1$ dimension, is used for the velocity calculation of next iteration as (1), where $\omega$ is the inertia weight; $U_{i1}[n]$ and $U_{i2}[n]$ are diagonal matrices with dimension $D$, and elements are random variables with uniform distribution $\sim U \in [0,1]$, generated for the $i$th particle at iteration $n = 1, 2, \ldots, N$; $p_{\text{best}}^i$ and $p_{\text{best}}^g$ are the best global position and the best local positions found until the $n$th iteration, respectively; $\phi_1$ and $\phi_2$ are acceleration coefficients regarding the best particles and the best global positions in influences in the velocity updating, respectively.

$$v_i[n+1] = \omega \cdot v_i[n] + \phi_1 \cdot U_{i1}[n](p_{\text{best}}^i - p_i[n]) + \phi_2 \cdot U_{i2}[n](p_{\text{best}}^g - p_i[n])$$ (1)

The $i$th particle’s position at iteration $n$ is a clustering candidate-vector $p_i[n]$ of size $D \times 1$. The position of each particle is updated using the new velocity vector (1) for that
The PSO algorithm consists of repeated application of the velocity and position updating equations until a stop criteria is found. The stop criteria can be a fixed number of iteration or determined by the non-improvement in the solution when the algorithm evolves.

In order to reduce the likelihood that the particle might leave the search universe, maximum velocity \( V_m \) factor is added to the PSO model (1), which will be responsible for limiting the velocity in the range \([±V_m]\). The adjustment of velocity allows the particle to move in a continuous but constrained subspace, been simply accomplished by (3).

\[
y_i[n] = \min \{ V_m; \max \{-V_m; v_i[n]\}\} \tag{3}
\]

From (3) it is clear that if \( |v_i[n]| \) exceeds a positive constant value \( V_m \) specified by the user, the \( i \)th particle’ velocity is assigned to be \( \text{sign}(v_i[n])V_m \), i.e. particles velocity on each of \( D \)-dimension is clamped to a maximum magnitude \( V_m \). If we could define the search space by the bounds \([P_{\text{min}}; P_{\text{max}}]\), then the value of \( V_m \) will be typically set to \( V_m = \tau(P_{\text{max}} - P_{\text{min}}) \), where \( 0.1 \leq \tau \leq 1.0 \) \cite{14}.

To elaborate further about the inertia weight, we note that a relatively larger value of \( w \) is helpful for global optimum, and lesser influenced by the best global and local positions\(^1\), while a relatively smaller value for \( w \) is helpful for convergence, i.e., smaller inertial weight encourages the local exploration as the particles are more attracted towards \( p_{i,\text{best}} \) and \( p_{g,\text{best}} \) \cite{15}. In this work, for simplicity, we have adopted an unitary inertia weight value.

V. NETWORK ANOMALY DETECTION MODEL BASED ON SWARM INTELLIGENCE

The elements of the proposed network anomaly detection system can be seen in Figure 2. The GBA tool \cite{4} is responsible for the collection of real traffic samples and generation of the DSNS. The PSO-Cls system calculates the cluster centroids from real traffic and DSNS. Then, the PSO Alarm system can analyze the distance between cluster centroids and real traffic samples, aiming to find the existence of anomalies.

The process for anomaly detection of the proposed system is divided into two stages, as follow:

1. The PSO-Cls system groups traffic data is collected from SNMP objects and their respective DSNS every 300 seconds, which are analyzed individually. Firstly, traffic data and DSNS from each 300-seconds interval are clustered simultaneously.

Then, a centroid for each cluster found is calculated, which represents the expected behavior for the traffic samples of the cluster. The pseudo code used for clustering data and calculate the centroids can be seen in Algorithm 1. The clustered data and clusters centroids generated in this stage are used in the next step.

\(^1\) Analogous to the idea of the phenomenon that it is difficult to diverge heavier objects in their flight trajectory than the lighter ones.

2. The PSO Alarm system is responsible for analyzing the results generated by the previous step, verifying if there were anomalies in the analyzed interval. The PSO Alarm system checks how close each sample of traffic movement is from their corresponding cluster centroid. The distance measure adopted in this work is the Euclidean distance, which consists of the straight line distance between two points. A sample is considered anomalous if the Euclidian distance between it and their respective cluster centroid, exceeds a threshold value \( \lambda \). Then, PSO Alarm system triggers an alarm to notify the network administrator, according Algorithm 2.

Algorithm 1 PSO-Cls Based Anomaly Detection

Function \text{PSO-Cls} \text{ system}

\text{Input:} \text{ real traffic, DSNS}

\text{Output:} \text{clustered traffic and DSNS, cluster centroids}

1. \text{Input data are clustered randomly}
2. \text{Population is initialized uniformly distributed in } \mathbb{U}[P_{\text{min}}; P_{\text{max}}]
3. \text{For } n = 1 \text{ to } N
    \text{For } i = 1 \text{ to } M
        \text{//velocity calculation}
        \text{ } v_i[n + 1] = \omega \cdot v_i[n] + \phi_1 \cdot U_{11}[n](p_{i,\text{best}} - p_i[n]) + \\
        \text{ } \phi_2 \cdot U_{12}[n](p_{g,\text{best}} - p_i[n])
        \text{//speed bounds}
        \text{ } v_i[n] = \min \{ V_m; \max \{-V_m; v_i[n]\}\}
        \text{//update the position}
        \text{ } p_i[n + 1] = p_i[n] + v_i[n + 1], \quad i = 1, \ldots, M
    \text{If } p_i \in [P_{\text{min}}; P_{\text{max}}]
        \text{Calculate the fitness value of } p_i \text{ and Update } p_{i,\text{best}} \text{ and } p_{g,\text{best}}
    \text{endIf}
\text{endFor}
\text{endFor}

4. \( p_{g,\text{best}} \) determines the cluster Centroids

End Function

\( P_{\text{min}}, P_{\text{max}}: \) minimum and maximum values of the input data

---

978-1-4244-5637-6/10/$26.00 ©2010 IEEE
Algorithm 2 PSO Alarm

Function PSO Alarm system
Input: clustered traffic and DSNS, cluster centroids
Output: detected anomalies
\[ X = \text{clustered traffic and DSNS} \]
\[ Z = \text{cluster centroids} \]
For each sample \( x \) of \( X \)
If \( D(x, Z) < \lambda \)
\( x \) is normal
Else
\( x \) is an anomaly, triggers an alarm
EndIf
EndFor
End Function

\( D \) = euclidean distance, \( \lambda \) = maximum distance threshold

VI. NUMERICAL RESULTS

In order to validate our anomaly detection system, a real environment was used to test the system. Data used in the experiment was collected during the 04/20/2009 to 04/26/2009 week, from ipInReceives objects of UEL main Web server. ipInReceives determines the number of IP packets received by the network element. Figure 1 presents charts with collected data and its respective DSNS.

As seen in section V, for each traffic sample, PSO Alarm system calculates the Euclidean distance between incoming data sample and its respective cluster centroid, aiming to verify whether the sample is anomalous. Every time the real traffic shows a significant deviation from the DSNS, implies in a substantial variation on the Euclidean distance values, characterizing a traffic volume anomaly. So, if this distance exceed the \( \lambda \) threshold value, the PSO Alarm system triggers an alarm to notify the network administrator.

The evaluation of the proposed anomaly detection system is based on two performance metrics: the detection rate, which consisting of the detection probability given by (4), and the false alarm rate, which represents the probability of alarms that not show significant variation between real traffic and the DSNS, according (5). The variables used to calculate the detection and false alarm rate are:

- correctly_detected: number of anomalies that were correctly detected.
- occ_anomalies: number of anomalies occurred in traffic.
- false_anomalies: number of alarms that do not correspond to an anomalous situation.
- total_alarms: number of generated alarms.

\[ \text{detection rate} = \frac{\text{correctly_detected}}{\text{occ_anomalies}} \] (4)

\[ \text{false alarm rate} = \frac{\text{false_anomalies}}{\text{total_alarms}} \] (5)

Aiming to validate the effectiveness of the proposed method, tests have been performed throughout a week test (Figure 1) of real data traffic generated in UEL network with different values of \( \lambda \), aiming find the best values for detection and false alarm rates. Figure 3 shows the alarms generated by the proposed system, for each day of the week test. The y-axis represents the Euclidean distances between samples and cluster centroids, and x-axis the time they occurred. Figure 4 describes the performance of PSO-Cls algorithm in terms of the trade-off between detection and false alarm rates, given \( \lambda \) as a parameter. One can observe that the suitable \( \lambda \) values on a ROC diagram [16] are in the range \( \lambda \in [1350; 1800] \). The results obtained after 100 iterations of PSO-Cls algorithm for each value of \( \lambda \), confirm that the method is useful for anomaly detection, achieving the best detection rate \( \times \) false alarm rate, 99.13% and 5.02% respectively, when \( \lambda = 1500 \).

It was also implemented a simplified version of the deterministic network anomaly detection algorithm proposed in [2]. The method is based on the mechanism of hysteresis and uses a parameter called \( \delta \) to reduce the possibility of generating false alarms [2]. Experiments have been carried out over entire data traffic from the considered week test, taking into account different values of \( \delta \), in order to determine the trade-off between detection and false alarm rates. Figure 5
indicates that the deterministic algorithm is capable to achieve 71.42% of detection rate and 5.71% of false alarm rate, when \( \delta = 12 \). This \( \delta \) value results in the best detection \( \times \) false alarm rate trade-off.

Fig. 4. Detection rate \( \times \) false alarm rate and \( \lambda \times \) false alarm rate for PSO-CIs algorithm.

The comparison between the best results obtained by the anomaly detection method proposed in this work and the deterministic method proposed in [2], shows that the PSO-based anomaly detection system has a gain of 27.71% in detection rate value and a reduction of 0.69% in the false alarm rate over the deterministic method. These results confirm that the model proposed in this work is more suitable for anomaly detection on a real network environment.

![Detection rate vs False alarm rate](image1)

Fig. 5. Detection rate \( \times \) false alarm rate and \( \delta \times \) false alarm rate for the deterministic network anomaly detection method.

Additionally in order to validate the choice of 300-seconds for the analysis interval as seen in section V, tests were performed with different interval sizes varying in the range of [50; 900] seconds, over the week test with \( \lambda \in [1350; 1800] \). Results showed that when considering the trade-off between detection rate and false alarms rate the anomaly detection system is more efficient using a 300-seconds interval, but could not be included in this work with complete proofs because of lack of space.

VII. CONCLUSION

The experiments’ results, obtained by the combination of the DSNS and the Particle Swarm optimization with re-clustering applied to the anomaly detection problem in a real network environment, showed that the method proposed in this work is capable of increasing the detection rate, while reducing false alarms.

Analyzing data collected from SNMP objects, the PSO-based anomaly detection algorithm has been shown robustness against false alarm while holds high anomaly detection rates, achieving 99.13 anomaly detection rate with 5.02 false alarm rate for \( \lambda = 1500 \) threshold, confirming that the method is feasible for anomaly detection.

Future work includes the application of the proposed model on the simultaneous monitoring of several SNMP objects, aiming the false alarm rate reduction through the correlation of SNMP objects.

REFERENCES


