Practical Anomaly Detection based on Classifying Frequent Traffic Patterns

Ignasi Paredes-Oliva*, Ismael Castell-Uroz*, Pere Barlet-Ros*, Xenofontas Dimitropoulos† and Josep Solé-Pareta*

*UPC BarcelonaTech
Jordi Girona, 1-3
08034 Barcelona, Spain
Email: {iparedes,icastell,pbarlet,pareta}@ac.upc.edu

† ETH Zurich
Gloriastrasse 35
8092 Zurich, Switzerland
Email: fontas@tik.ee.ethz.ch

***Abstract*—Detecting network traffic anomalies is crucial for network operators as it helps to identify security incidents and to monitor the availability of networked services. Although anomaly detection has received significant attention in the literature, the automatic classification of network anomalies still remains an open problem. In this paper, we introduce a novel scheme and build a system to detect and classify anomalies that is based on an elegant combination of frequent item-set mining with decision tree learning. Our approach has two key features: 1) *effectiveness*, it has a very low false-positive rate; and 2) *simplicity*, an operator can easily comprehend how our detector and classifier operates. We evaluate our scheme using traffic traces from two real networks, namely from the European-wide backbone network of GÉANT and from a regional peering link in Spain. In both cases, we achieve an overall classification accuracy greater than 98% and a false-positive rate of approximately only 1%. In addition, we show that it is possible to train our classifier with data from one network and use it to effectively classify anomalies in a different network. Finally, we have built a corresponding anomaly detection and classification system and have deployed it as part of an operational platform, where it is successfully used to monitor two 10Gb/s peering links between the Catalan and the Spanish national research and education networks (NREN).**

***Index Terms*—Network Security, Anomaly Detection, Anomaly Classification, NetFlow**

I. Introduction

During the last years, the presence of cyber attacks has grown dramatically throughout the Internet [1]. Consequently, the detection of anomalies on network traffic data has been studied extensively. Despite significant research efforts, anomaly detection systems have not been yet widely adopted by network operators mainly because: 1) they generate a large number of false positives; 2) they use complex detection techniques that are often incomprehensible, i.e., a "black box", to network operator; and 3) they require learning a labeled "ground truth" dataset from the target network.

In addition to anomaly detection, anomaly classification, i.e., automatically identifying the type of a detected anomaly, has been marginally studied in the past. Most anomaly detection systems, e.g., [2]–[5], differentiate between normal and anomalous traffic, but they usually cannot distinguish different anomaly types. Moreover, analyzing the root cause of network anomalies is a matter of increasing importance. Once an anomaly has been flagged, network operators need to diagnose the type of the anomaly in order to act accordingly and take the appropriate mitigation measures. Presently, operators typically need to invest many work-hours to manually inspect and classify detected anomalies, which is an extremely complex, slow, and expensive process [6].

In this paper, we introduce a scheme that automatically detects and classifies anomalies in high-speed networks using traffic flow data, e.g., NetFlow. Our scheme combines simple and effective techniques from two worlds: data mining and machine learning. First, we use frequent item-set mining (FIM) to find a set of *frequent item-sets* (FIs). A frequent item-set is a large set of flows that have one or more flow features in common. For example, an attack such as a *Network Scan* will produce many flows with the same source IP address and destination port number, which after applying FIM will give rise to one frequent item-set with two items: the scanner IP address and the scanned port number. Second, our scheme builds a decision tree to classify frequent item-sets as anomalous or benign and to determine their specific type in case they are anomalous. A key novelty of our approach is that we classify FIs based on a set of FI features. Intuitively, we decompose observed traffic into distinct groups (FIs) of related traffic flows, which enables us to classify each FI with high accuracy. The decision tree classifies the type of an anomaly along a two-level hierarchy: a main class, e.g., *Port Scans, Network Scans or Denial-of-Service*, and a subclass, e.g., *UDP scan, SYN Scan, ACK Scan*, etc. A main advantage of our scheme is that it is conceptually simple and, therefore, easy to understand and configure by a network operator. Our scheme can be used as a stand-alone anomaly detector and classifier

or it can be combined with anomaly extraction [7], in which case it only performs anomaly classification.

We have thoroughly evaluated our scheme with real traces and manually verified the detected anomalies. An exhaustive analysis using data from the European-wide backbone network of GÉANT showed that our scheme has very high accuracy (approximately 98%). Finally, we have built a corresponding anomaly detection and classification system, have deployed it in a regional academic network and have further confirmed its effectiveness by reporting similar accuracy and low false positive rates (approximately 1%) when monitoring the two 10 Gb/s links that connect the Catalan and the Spanish NREN (RedIris). The decision trees used by our system were trained with labeled data from the GÉANT network and were found to be accurate when applied to classify anomalies in the regional academic network. This is particularly promising as it shows that it is possible to learn a traffic model in one network and use it effectively in a different network.

The rest of this paper is organized as follows. Section II briefly reviews the related work. Section III presents our anomaly detection and classification scheme. Section IV describes our evaluation in two network environments. Finally, Section V concludes our paper.

II. Related Work

Several works have proposed various anomaly detection techniques (for a survey of results refer to [8]). Among these, a number of network traffic anomaly detection methods use wavelets [2], Principal Component Analysis (PCA) [3], [4] or Kalman filters [5] to distinguish between normal and anomalous traffic, but do not provide information about the type of detected anomalies.

The problem of anomaly extraction has been recently treated in the literature [7], [9]. Among the existing proposals, the most relevant to our work [7] uses a frequent item-set mining (FIM) algorithm to identify the flows related to an anomaly from a given hint (e.g., an involved IP address) provided by an external anomaly detector. Our system uses a FIM algorithm (FPmax* [10]) not only to extract those flows associated to an anomaly but also to 1) detect anomalies if we do not have an external detector at our disposal; and to 2) classify anomalies.

To the best of our knowledge, only few works have addressed the problem of anomaly classification. Lakhina *et al*. [11] cluster the output of a PCA-based anomaly detector to identify anomalies with similar behavior. Human intervention is necessary to find out the correspondence between each reported cluster and the high-level anomaly that it is describing. Tellenbach *et al*. [12] classify changes to generalized entropy metrics of traffic feature distributions to identify the type of detected anomalies. They demonstrate that this approach can classify with accuracy of around 85% synthetic anomalies. Most related to our approach, Choi *et al*. [13] make use of parallel coordinate plots to find unique patterns of attacks that are easy to recognize visually by a human expert. In contrast, we propose a technique that can automatically classify network anomalies without requiring later manual inspection.

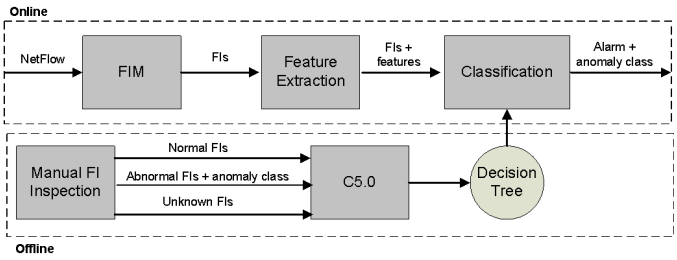

Fig. 1: System Overview

III. Anomaly Detection and Classification

A fundamental idea of our approach is that the entity that is classified as either normal or anomalous is a frequent item-set. In contrast, previous approaches to network traffic anomaly detection typically classify a time interval based on aggregate metrics derived from the traffic observed during this time interval. We also split time into intervals, e.g., 5-minute intervals, then for each time interval we find the corresponding frequent item-sets, and finally we classify frequent item-sets. We argue that it is much easier to classify a frequent item-set as normal or anomalous than to classify the entire traffic observed during a time interval since a frequent item-set conveniently groups flows related to anomalies. We describe how we exploit FIM in Section III-A.

Figure 1 shows an overview of our anomaly detection and classification scheme. The offline information flow in the bottom of the figure creates a decision tree starting from a semi-manually derived ground truth of normal, anomalous, or unknown frequent item-sets. The ground truth, as we discuss in detail further on, can be derived using the traffic inspection techniques developed in [14] and/or by leveraging information from external anomaly detectors. The normal, anomalous, and unknown frequent item-sets are fed to a learning algorithm, which produces a simple decision tree that is used for detecting anomalies and for classifying the type of a detected anomaly. On top of Figure 1, we illustrate the online information flow. Given a set of traffic flows observed, e.g., with NetFlow, during a time interval, we first apply FIM to extract frequent item-sets; then for each frequent item-set we compute a number of traffic features; and finally we use the decision tree to classify frequent item-sets based on their features. We describe our anomaly classification approach based on supervised learning in Section III-B.

A. Frequent Item-set Mining

Frequent item-set mining (FIM) is a well-known data mining technique that focuses on finding items that occur frequently together in a certain dataset. A set of items will be considered frequent if they appear together at least as many times as a given threshold, which is called *minimum support*. Applying FIM to network traffic allows us to discover groups of many flows sharing a certain combination of features (e.g., IPs or/and ports).

The extraction of item-sets allows us to represent traffic in a compact form, which more clearly reflects the characteristic

	Src IP	Dst IP	Src Port	Dst Port	Protocol
Item-set 1	*	1.1.1.1	*	5010	TCP
Item-set 2	2.2.2.2	*	*	80	TCP
Item-set 3	*	*	*	80	TCP

TABLE I: Example of item-sets

behavior and patterns of network anomalies. The set of items we select to perform the mining on is the well known 5-tuple flow (source/destination IP addresses, source/destination ports and protocol). The selection of these items makes our scheme compatible with NetFlow, which is widely deployed in operational environments.

Table I shows three example item-sets, one per row. Each of these item-sets summarizes a group of flows. Each field can have a specific value or a wild card, which means that there is no restriction for that field. For example, the second item-set summarizes all flows coming from one specific IP address that is scanning multiple destination IP addresses on port 80 using different source ports, while the first item-set resembles a distributed attack to a specific destination IP address and port. The learning method we present in Section III-B automatically identifies the correct anomaly class of an item-set. In our example, it would classify the first item-set as a *DDoS* and the second as a *Network Scan*.

In general, FIM algorithms report all frequent item-sets above the given *minimum support*. The downwards closure property [15] of FIM states that if an item-set is frequent, then all its subsets (less concrete item-sets, i.e., subset item-sets that have more wild cards) are also frequent. In the example of Table I, this means that if the second item-set is frequent, then the third item-set must be frequent as well (item-set 3 is a subset of item-set 2). Because there may exist many such frequent subset item-sets, the output of a standard FIM algorithm can be overwhelming.

In order to avoid the exponential growth in the output, we use *maximal item-set mining*. Maximal item-set mining finds only those frequent item-sets that do not have a frequent superset. In the example of Table I, maximal item-set mining would suppress the third item-set. We selected the FPmax* algorithm [10] as a miner, which according to [16] is one of the best performing methods for finding maximal item-sets. Nonetheless, note that our system is not tied to FPmax* and any other frequent item-set miner could be used instead.

We specify the *minimum support* in terms of flows, although as we will describe in Section III-B, we also have a separate data structure to deal with attacks that result in a large number of packets, but in a small number of flows, e.g., single *DoS*.

Figure 2 shows how the number of frequent item-sets changes with the *minimum support (ms)* parameter. The dashed line shows on the right y-axis the total number of frequent item-sets in the output. The solid line shows on the left y-axis the number of frequent item-sets we have manually identified as anomalous. We observe how these two variables change depending on the *ms*. We observe that although decreasing the *ms* increases the number of anomalous item-sets, this comes at a price: the total amount of found item-sets rises exponentially. Moreover, we observe that for any *ms*, the amount of reported frequent item-sets is always far higher than the number of anomalous item-sets. Our classifier distinguishes between normal and anomalous item-sets. As we describe in Section III-B, we create a ground truth of normal, anomalous, and unknown item-sets and learn their characteristics. This way, our classifier can automatically filter out those item-sets that correspond to normal traffic and reduce the number of item-sets reported.

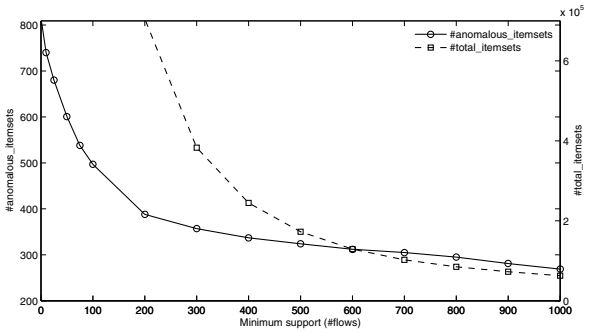

Fig. 2: Total number of reported frequent item-sets (right axis) vs. anomalous frequent item-sets (left axis) for varying *minimum support*

Compared to the anomaly extraction technique of [7], which applies FIM on a pre-filtered set of flows based on meta-data from an external anomaly detector to extract and summarize anomalous flows, our approach is substantially different. First, we also detect anomalies by classifying FIs using a very simple classifier. Second, we additionally classify detected anomalies. Our scheme can be used as stand-alone anomaly detector and classifier or in combination with the anomaly extraction technique of [7] to classify anomalies. In Section IV, we evaluate our system using real traces from two different networks and we show that it obtains high accuracy both with and without an external detector.

Selecting the correct *ms* parameter requires a small number (typically 2-3) of "trial and error" iterations. As reported in [17], actionable anomaly alarms require on average 60 minutes of investigation, which would correspond to 8 alarms per day assuming a full-time employ for analyzing alarms. Therefore, the *ms* parameter should be set high enough so that the output does not overwhelm the administrator of the system with too many alarms. Through extensive experimentation, we have found that a suitable *ms* parameter is typically in the range between 1% and 10% of the number of input flows. In Section IV we use this intuitive rule to guide the selection of the *ms*.

B. Anomaly Classification

In order to classify the item-sets extracted by FPmax* we use a machine learning (ML) technique. According to [18], the C4.5 algorithm [19] is one of the methods that offers the best trade-off between execution time and accuracy in traffic classification. We selected its evolution, the C5.0 algorithm.

	Value	
	Defined	**Undefined**
Defined Src IP/Dst IP	True	False
Src/Dst Port	Port Number	NaN
Protocol	Protocol Number	NaN
URG/ACK/PSH/RST/SYN/FIN	True	False
Bytes per Packet	$\#Bytes/\#Packets$	
Packet per Flow	$\#Packets/\#Flows$	

TABLE II: Frequent item-set classification features

In order to apply ML for anomaly classification, we first need a ground truth of pre-classified anomalies. Secondly, we have to decide what features will be used to perform the classification. Finally, we need to define a list of anomaly types.

The C5.0 algorithm uses a labeled data-set to create a model that predicts the class of an item-set. It selects one feature at each step to split the training dataset accordingly. The importance of a particular feature is given by the *gain ratio (GR)* metric, which is the entropy H for each anomaly class with and without the selected feature. The gain ratio is defined as follows:

$$GR(Class, Feature) = \frac{H(Class) - H(Class|Feature)}{H(Feature)}$$

Establishing a ground truth is one of the most critical phases of any machine learning process, because the entire classification process relies on its accuracy. We have used a set of anomalies detected in the GÉANT backbone network. These anomalies were first detected by three commercial anomaly detectors and then manually validated by the security experts of GÉANT [6]. An important aspect of our classification is that the decision tree needs to distinguish between legitimate and anomalous item-sets. Consequently, we extended the ground truth as described in Section IV with item-sets corresponding to normal traffic.

For each item-set we use the 13 features shown in Table II that can be trivially derived from NetFlow data. For the source and destination IP addresses, we use a boolean that states if an IP address is part of a frequent item-set. The source and destination port features take values between 1 and 65535. If a port is not defined, a special value is used to indicate that it is not a number (NaN). Similarly, the protocol feature takes a specific protocol value or the NaN value. The *Bytes per Packet (bpp)* and *Packets per Flow (ppf)* features are important to collect information about the bandwidth usage and to capture the used packet sizes, e.g., packets of minimal length are commonly used in attacks such as *Network Scans* or *Port Scans*. Finally, the TCP flags are encoded in six boolean features indicating if a specific flag is set or not.

Our system classifies the anomalies in the following classes and sub-classes:

- *DoS: Denial-of-service*. This class of anomalies corresponds to attempts to make a computer resource unavailable to its intended users. In its distributed variant (DDoS) the attack comes from multiple systems. The possible subclasses are *DDoS, SYN Floods, ACK Floods, UDP Floods* and *ICMP Floods*.
- *Port Scan*. A scanner sends client requests to several ports of a target system with the goal of finding active services. This anomaly is also known as *vertical scanning*. It has three subclasses: *SYN*, *ACK* and *UDP Scan*.
- *Network Scan*. The scanner probes multiple hosts, e.g., a whole subnet, to find systems running a specific service. This attack is also known as *horizontal scanning*. Its subclasses are *ICMP Scan* and *Other Network Scans*.
- *Unknown*. Traffic with strange behaviour that did not fit into any of the anomaly classes above.
- *Normal*. This class includes the legitimate traffic.

We created two different decision trees, one for item-sets with a large number of flows (*Main Tree*), and a second for subclasses of *DoS* with a small number of flows (*Secondary Tree*). We split the tree for performance reasons, since the *Secondary Tree* is much more compact than the *Main Tree*.

Finally, we ranked the features with the *gain ratio* metric, which is the discriminator used by C5.0, to understand the importance of each feature and to identify redundant ones. Figure 3 shows the that all the features were important except for the URG flag in the *Secondary Tree*.

IV. Evaluation and Deployment

To build the ground truth for our evaluation, we used a set of 994 anomalies that were first detected by three commercial anomaly detection systems (NetReflex [20], PeakFlow SP [21] and StealthWatch [22]) over a 14-day period and then were manually verified by a team of security experts in GÉANT [6].

We ran FPmax* on raw NetFlow data collected during the reported anomalous time intervals using a starting *minimum support (ms)* value of 1000. We manually examined the output to identify item-sets corresponding to the reported anomalies. When we could not find one, we progressively decreased *ms* until an anomalous item-set was found. In few cases, which we discarded, the *ms* reached almost zero without identifying an anomalous item-set. After this process, we ended up with 760 anomalous item-sets, which we used as ground truth. To classify an item-set as normal or anomalous, we manually examined a sample of the flows of an item-set, e.g., the first 20, using the methodology shown in [23]. We classified an item-set as anomalous, if its flows had suspicious feature values, e.g., destination port 22, and regular patterns pointing to an automated process, like a bot, that generated them. For example, one such regular pattern is the observation of a very large number of flows with a specific packet size, combination of TCP flags, source addresses, and flow inter-arrival time interval. In order to differentiate between malicious and legitimate traffic, we added normal item-sets to the training set. To find normal item-sets we applied FPmax* on random samples of 30-minute intervals for the same 14-day period during which the anomalies were found. We then manually inspected the flows of approximately 300 new item-sets and classified them as anomalous if they exhibited specific

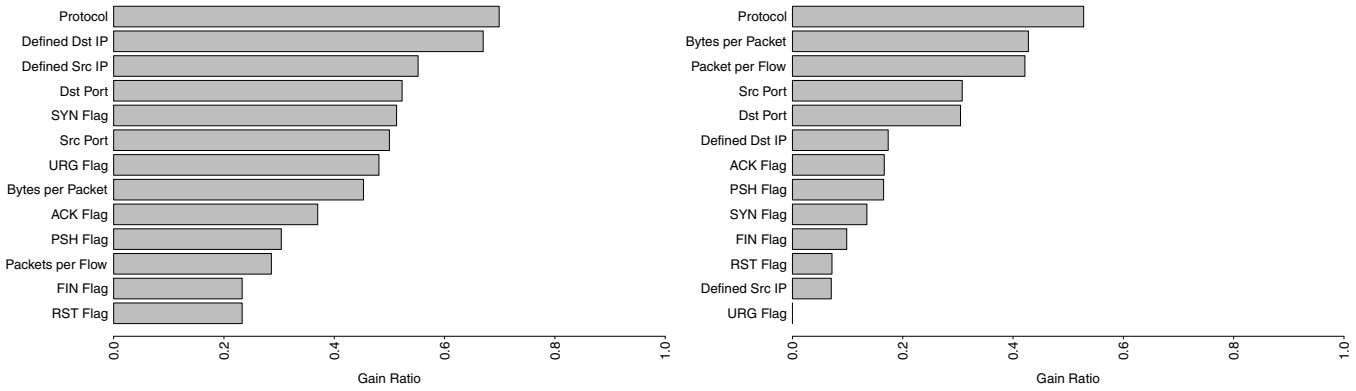

Fig. 3: Feature gain ratio for the main (left) and secondary (right) tree.

malicious patterns. Finally, since some *DoS* anomalies use a very small number of flows (mostly one or two) but lots of packets or bytes, we also added to the training set item-sets of anomalous and normal traffic with these properties, i.e., a small number of flows and a very large number of packets or bytes. The final ground truth used in the evaluation contains 1249 labeled item-sets.

We use three standard metrics to measure the performance of our classifier. While the first captures its overall accuracy, the second and third metrics evaluate its performance for specific anomaly classes.

$$overall\_accuracy = \frac{TruePositives}{TruePositives+FalsePositives}$$

$$precision(X) = \frac{TruePositives(X)}{TruePositives(X)+FalsePositives(X)}$$

$$recall(X) = \frac{TruePositives(X)}{TruePositives(X)+FalseNegatives(X)}$$

We implemented and evaluated our scheme using data from two different networks: 1) from the European backbone network of GÉANT (Section IV-A) and 2) from two 10Gb/s links interconnecting the Catalan with the Spanish NREN (Section IV-B).

A. Evaluation in GÉANT

In these experiments, we used our system solely for traffic classification. To validate the classification results, we used 10-fold cross-validation and report the average figures over 10 rounds. In Figure 4 we illustrate the precision and recall per class as well as the overall classification accuracy.

We observe that without balancing, even if the overall accuracy is greater than 95%, our classifier had a moderate performance for certain classes, i.e., note the low precision and recall (two left bars) for *ACK Port Scans* and *ICMP Floods*. Despite the poor performance for these two classes, the overall accuracy with the unbalanced training set was high because these two classes make a small fraction of the overall number of anomalies in the training data. To address this problem, we balanced the representativeness of anomaly

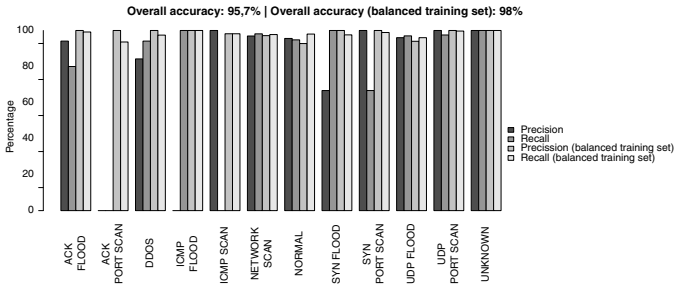

Fig. 4: GÉANT classification accuracy

classes in the training set by increasing the instances of under-represented types. We also balanced the class of normal traffic to increase the frequency of applications with few occurrences (e.g., DNS). With these modifications, our classifier had very good results for the most common anomalies and substantially increased its precision and recall also for the previously under-represented classes. The precision and recall for each anomaly type with the balanced training data is shown by the two right bars in Figure 4 and is substantially higher than with the unbalanced training data. The overall accuracy also increased from nearly 96% to approximately 98%.

B. Deployment in a Production Network Monitoring Platform

In our second scenario, we integrated our system into a production monitoring platform, called SMARTxAC [24], and deployed it on the Catalan NREN (Anella Científica) to monitor two peering links to the Spanish NREN (RedIris). In the experiments we describe below, we use our scheme for both anomaly detection and classification.

The *minimum support (ms)* is set to 3% of the total traffic, which triggers approximately 30 anomalies per day. To classify the item-sets, we used the two decision trees, which were trained using the labeled data derived from GÉANT. To validate the accuracy of our solution, we manually inspected all the detected anomalies during the first 10 days of August 2011. During this period, the system reported 310 different attacks of 7 different anomaly (sub)classes.

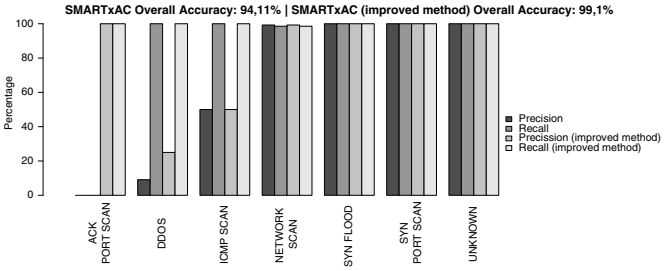

Fig. 5: SMARTxAC classification accuracy

The two leftmost bars of each block in Figure 5 show the classification results. The overall accuracy of the system was greater than 94%. For two anomaly classes, i.e., *DDoS* and *ACK Port Scans*, the precision rate was low. These classes made a very small fraction of the total number of anomalies (4,52% and 1,29% respectively). Except for these two cases, the system had a very good accuracy with *only 18 false positives out of 310 anomalies*.

The analysis of the low precision for *DDoS* and *ACK Port Scans* showed that 80% of these false positives were replies from *SYN Floods* and *Network Scans* misclassified as *ACK Port Scans* and *DDoS*, respectively. In order to correct this, we modified the system to look for these specific response patterns. The two rightmost columns of each group in Figure 5 show the precision and recall after applying the improvement to the system. As we can observe, the overall accuracy improved to 99,1%. The false positive detection ratio decreased to 1,3% (only 4 erroneously classified anomalies out of 310).

The results in this second scenario show that the decision trees, which were built by learning from the GÉANT network, can be effectively used in a different network. This indicates that the model and the selected features behave well independently of the network characteristics.

V. Conclusions

In this paper, we presented a novel scheme to detect and classify network traffic anomalies that combines frequent itemset mining with decision tree learning. Using real data from two different networks we showed that our solution has a very high classification accuracy, i.e., above 98%, and a low false positive rate, i.e., approximately 1%. In addition, with our proposal it is easy for network operators to understand and reason about detected anomalies. Based on our scheme, we have implemented an anomaly detection and classification system and deployed it in a production network, where it successfully monitors two 10 Gb/s links. Moreover, a particularly promising feature of our classifier is that it has been trained using traffic traces from the European backbone network of GÉANT and has been used successfully to detect and classify anomalies in a substantially different regional network.

Acknowledgments

We thank DANTE and CESCA for having provided us access to GÉANT and Anella Científica, respectively. This work was partially funded by the Spanish Ministry of Education under contract TEC2011-27474 and the Catalan Government under contract 2009SGR-1140.

References

[1] K.-K. R. Choo, "The cyber threat landscape: Challenges and future research directions," *Computers & Security*, vol. 30, no. 8, pp. 719–731, 2011.

[2] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of ACM SIGCOMM Workshop on Internet Measurment*, 2002.

[3] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proceedings of ACM SIGCOMM*, 2004.

[4] ——, "Characterization of network-wide anomalies in traffic flows," in *Proceedings of ACM SIGCOMM conference on Internet Measurement (IMC)*, 2004.

[5] A. Soule, K. Salamatian, and N. Taft, "Combining filtering and statistical methods for anomaly detection," in *Proceedings of ACM SIGCOMM conference on Internet Measurement (IMC)*, 2005.

[6] M. Molina, W. Routly, I. Paredes-Oliva, and A. Jain, "Anomaly Detection in Backbone Networks: Building a Security Service Upon an Innovative Tool." in *Proceedings of Terena Networking Conference (TNC)*, 2010.

[7] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian, "Anomaly extraction in backbone networks using association rules," in *Proceedings of ACM SIGCOMM conference on Internet Measurement (IMC)*, 2009.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[9] F. Silveira and C. Diot, "URCA: Pulling out anomalies by their root causes," in *Proceedings of IEEE INFOCOM*, 2010.

[10] G. Grahne and J. Zhu, "Efficiently using prefix-trees in mining frequent itemsets," in *Proceedings of Workshop on Frequent Itemset Mining Implementations (FIMI)*, 2003.

[11] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proceedings of ACM SIGCOMM*, 2005.

[12] B. Tellenbach, M. Burkhart, D. Schatzmann, D. Gugelmann, and D. Sornette, "Accurate network anomaly classification with generalized entropy metrics," *Computer Networks*, vol. 55, no. 15, pp. 3485–3502, Oct. 2011.

[13] H. Choi, H. Lee, and H. Kim, "Fast detection and visualization of network attacks on parallel coordinates," *Computers & Security*, vol. 28, no. 5, pp. 276–288, 2009.

[14] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," in *Proceedings of ACM SIGCOMM conference on Internet Measurement (IMC)*, 2006.

[15] R. Agrawal, T. Imieliski, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Record*, vol. Vol. 22, pp. 207–216, 1993.

[16] B. Goethals and M. Zaki, "FIMI03: Workshop on frequent itemset mining implementations," in *Proceedings of Workshop on Frequent Itemset Mining Implementations (FIMI)*, 2003.

[17] P. E. Proctor, "Marketscope for network behavior analysis," *Gartner Research Report G00144385*, 2006.

[18] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: Myths, caveats, and the best practices," in *Proceedings of ACM CoNEXT*, 2008.

[19] J. Quinlan, *C4. 5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[20] Guavus, "NetReflex," http://www.guavus.com.

[21] Arbor Networks, "PeakFlow SP," http://www.arbornetworks.com/.

[22] Lancope, "StealthWatch," http://www.lancope.com/.

[23] I. Paredes-Oliva, X. Dimitropoulos, M. Molina, P. Barlet-Ros, and D. Brauckhoff, "Automating root-cause analysis of network anomalies using frequent itemset mining (demo)," in *Proceedings of ACM SIGCOMM*, 2010.

[24] P. Barlet-Ros, J. Solé-Pareta, J. Barrantes, E. Codina, and J. Domingo-Pascual, "SMARTxAC: A passive monitoring and analysis system for high-speed networks." *Campus-Wide Information Systems*, vol. 23, no. 4, pp. 283–296, 2006.