Poster: (Semi)-Supervised Machine Learning Approaches for Network Security in High-Dimensional Network Data

Pedro Casas (1)*, Alessandro D'Alconzo (1), Giuseppe Settanni (1), Pierdomenico Fiadino (2), Florian Skopik (1)
(1) AIT Austrian Institute of Technology, (2) Eurecat Technology Centre of Catalonia, * Corresponding Author
(1) name.surname@ait.ac.at, (2) pierdomenico.fiadino@eurecat.org

ABSTRACT

Network security represents a keystone to ISPs, who need to cope with an increasing number of network attacks that put the network's integrity at risk. The high-dimensionality of network data provided by current network monitoring systems opens the door to the massive application of machine learning approaches to improve the detection and classification of network attacks. In this paper we devise a novel attacks detection and classification technique based on semi-supervised Machine Learning (ML) algorithms to automatically detect and diagnose network attacks with minimal training, and compare its performance to that achieved by other well-known supervised learning detectors. The proposed solution is evaluated using real network measurements coming from the WIDE backbone network, using the well-known MAWILab dataset for attacks labeling.

Keywords

Network Attacks, Machine Learning, Clustering, MAWILab, High-Dimensional Data.

1. INTRODUCTION

Network security and anomaly detection has become a vital component of any network in today's Internet. Ranging from non-malicious unexpected events such as flash-crowds and failures, to network attacks and intrusions such as denials-of-service, network scans, worms propagation, botnets activity, etc., network traffic anomalies can have serious detrimental effects on the performance and integrity of the network. The principal challenge in automatically detecting and characterizing traffic anomalies is that these are moving targets. It is difficult to precisely and continuously define the set of possible anomalies that may arise, especially in the case of network attacks, because new attacks as well as new variants to already known attacks are continuously emerging. A general anomaly detection system should therefore be able to detect a wide range of anomalies with diverse structures, without relying exclusively on previous knowledge and information. In this paper we devise k-CDA, a novel attacks detection and classification technique based on semi-supervised Machine Learning (ML) algorithms to automatically detect and diagnose network attacks with minimal training data. We additionally investigate and compare its performance to that achieved by other well-known supervised learning classifiers. In Sec. 2 we describe k-CDA and briefly overview the tested supervised models, whereas data description and results are presented in Sec. 3.

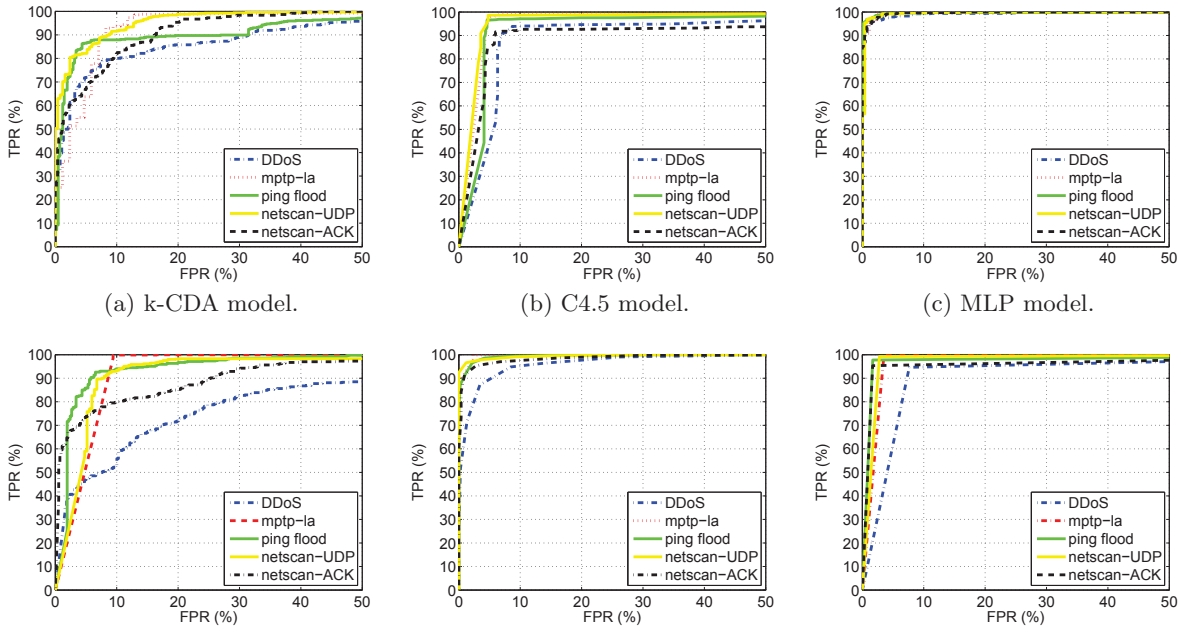
2. ML-BASED APPROACHES

In this section we describe the proposed detection and classification approach based on clustering, and briefly introduce five well-known, fully supervised based detection approaches used for testing and comparison purposes.

2.1 Clustering-based Analysis

The clustering-based approach introduced in this work, referred to as k-CDA (k-means Clustering-based Detector of Attacks) has the main advantage of being semi-supervised, which means that the amount of learning data which is required for training/calibration purposes if significantly less than that required by supervised approaches. Given a training set of m measurements consisting each of n features, our method uses first the well known K-means clustering algorithm [2] to partition the complete feature space $X \in \mathbb{R}^{m \times n}$ in a set of K clusters. The centroid of each of these K clusters is then computed, and a label is assigned to each of them, based on majority voting performed on a small sample of ground truth labels among the measurements belonging to each cluster. In particular, we decide the class y of a cluster based on the real label of only 5% of the samples within this cluster, randomly sampled. We have verified that this small fraction is good enough to provide proper detection and classification results. Once all clusters have been labeled, the standard approach to follow for clustering-based classification is to assign, to every new sample, the class of the cluster with the closest centroid [4]. However, given that the real number of clusters K is not known in advance, following such an approach might be counterproductive and lead to less robust results [4].

Indeed, one well-known limitation of using K-means as clustering algorithm is that one needs to define in advance the number of K clusters to identify, which in principle is completely unknown, specially when no labeled data is used. Selecting a small value for K results in bigger and potentially less homogeneous clusters; having a big number of clusters has the advantage of resulting in more homogeneous clusters, but if this number is too big, the analysis and interpretation of results becomes more cumbersome and the advantages of grouping samples together is partially lost. To partially solve this issue, we resort to a simple heuristic, based once again on a majority voting approach. We set K to a value equal to 0.1% of the training sample size, i.e., $K = m/1000$, and then decide on the label of a new sample using a k-NN (Nearest Neighbors) algorithm, computing the distance of the new sample to all the K centroids, and using majority voting on the labels of the k closest centroids. By doing so,



(a) k-CDA model. (b) C4.5 model. (c) MLP model.

(d) NB model. (e) RF model. (f) SVM model.

Figure 1: Detection performance per type of attack and ML-based approach.

we obtain more homogeneous clusters, and limit the impacts of single centroid-based classification. The value of k clearly depends on the value of K: based on empirical testing, we set $k = K/3$ (naturally, all values are rounded to obtain integer numbers for both k and K).

The final ingredient of our approach is on the particular way we compute distances: instead of using a simple Euclidean distance, we compute the per-cluster normalized Mahalanobis distance between every new sample and the K labeled centroids. The Mahalanobis distance takes into account the correlation between samples, dividing the standard Euclidean distance by the variance of the samples belonging to each cluster. In a nutshell, if a cluster has a bigger variance on a certain direction (i.e., feature), then the Mahalanobis distance will make samples closer to this cluster than to other ones with smaller variance, making less compact clusters closer to samples.

2.2 Supervised-based Analysis

We select five fully-supervised models using C4.5 Decision Trees (C4.5), Random Forest (RF), Support Vector Machines (SVM), Naïve Bayes (NB) and Neural Networks (MLP). We selected these detectors for comparison based on the a-priori good performance shown by their application in previous work on anomaly detection [3] and traffic classification [5]. We use the well-known Weka Machine-Learning software tool to calibrate these ML-based algorithms and to perform the evaluations. We address the interested reader to the survey [5] and to the Weka documentation for additional information on the algorithms and their configuration parameters.

3. DATA DESCRIPTION AND RESULTS

We test the performance of the proposed approaches using real network traffic measurements coming from the WIDE backbone network, using the well-known MAWILab dataset for attacks labeling [1]. MAWILab is a public collection of 15-minute network traffic traces captured every day on a backbone link between Japan and the US since 2001. Building on this repository, the MAWILab project uses a combination of four traditional anomaly detectors (PCA, KL, Hough, and Gamma, see [1]) to partially label the collected traffic. The traffic studied in this paper spans 2 months in late 2015. From the labeled anomalies and attacks, we focus on a specific group which are detected simultaneously by the four MAWILab detectors, using in particular those events which are labeled as "anomalous" by MAWILab. As such, we highly increase the quality of the obtained labels, as we keep only those which have the highest concensus. We consdier in particular 5 types of attacks/anomalies: (i) DDoS attacks (DDoS), (2) HTTP flashcrowds (mptp-la), (3) Flooding attacks (Ping flood), and two different flavours of distributed network scans (netscan) using (4) UDP and (5) TCP probing traffic. We train the different models to detect each of these attack types separately, thus each detection approach consists of five different detectors which run in parallel on top of the data, each of them specialized in detecting one of the five aforementioned attacks types. As a result, each detection approach can not only detect the occurrence of an attack, but also classify its nature. Finally, we evaluate detection performance per anomaly type, considering a slotted, time-based evaluation. For doing so, we split the traffic traces in consecutive time slots of one second each, and compute a set of features describing the traffic in each of these slots. In addition, each slot i is assigned a label l_i, consisting of a binary vector $l_i \in \mathbb{R}^{5\times 1}$ which indicates at each position if anomaly of type $j = 1..5$ is present or not in current time slot.

For the sake of better detecting and diagnosing anomalies and attacks, we compute a large number n of features de-

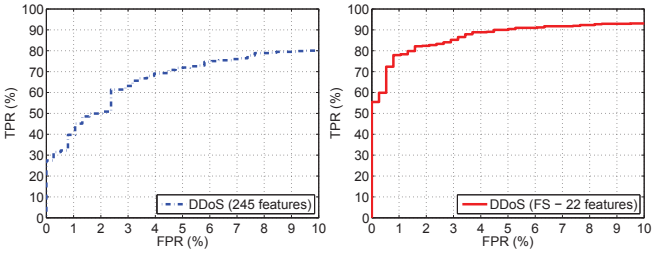
Field	Feature	Description
Tot. volume	#_pkts	num. packets
	#_bytes	num. bytes
PKT size	pkt_h	H(PKT)
	pkt_{min,avg,max,std}	min/max/std, $\overline{\text{PKT}}$
	pkt_p{1,2,5,...95,97,99}	percentiles
IP Proto	#_ip_protocols	num. diff. IP protos.
	ipp_h	H(IPP)
	ipp_{min,avg,max,std}	min/max/std, $\overline{\text{IPP}}$
	ipp_p{1,2,5,...95,97,99}	percentiles
	% icmp/tcp/udp	share of IP protos.
IP TTL	pkt_h	H(TTL)
	ttl_{min,avg,max,std}	min/max/std, $\overline{\text{TTL}}$
	ttl_p{1,2,5,...95,97,99}	percentiles
IPv4/IPv6	% IPv4/IPv6	share of IPv4/IPv6 pkts.
	#_IP_src/dst	num. unique IPs
	top_ip_src/dst	most used IPs
TCP/UDP ports	#_port_src/dst	num. unique ports
	top_port_src/dst	most used ports
	port_h	H(PORT)
	port_{min,avg,max,std}	min/max/std, $\overline{\text{PORT}}$
	port_p{1,2,5,...95,97,99}	percentiles
TCP flags (∀)	flags_h	H(TCPF)
	flags_{min,avg,max,std}	min/max/std, $\overline{\text{TCPF}}$
	flags_p{1,2,5,...95,97,99}	percentiles
TCP WDW size	wdw_h	H(WDW)
	wdw_{min,avg,max,std}	min/max/std, $\overline{\text{TCPF}}$
	wdw_p{1,2,5,...95,97,99}	percentiles

Table 1: Input features for the ML-based detectors. The full set consists of 245 different input features.

scribing a time slot, using traditional packet measurements including traffic throughput, packet sizes, IP addresses and ports, transport protocols, flags, etc. Tab. 1 describes the set of $n = 245$ features, which are computed for every time slot $i = 1..m$. Note that besides using traditional features such as min/avg/max values of some of the input measurements, we also consider the empirical distribution of some of them, sampling the empirical distribution at many different percentiles. This is not a common technique but provides as input much better information, as the complete distribution is taken into account. We also compute the empirical entropy of these distributions, reflecting the dispersion of the observed samples in the corresponding time slot.

3.1 Detection and Classification Performance

We test the detection/classification capabilities of k-CDA and the other supervised approaches by computing the True and False Positive Rates (TPR/FPR) for each of the attack types. Fig. 1 depicts the Receiver Operating Characteristic (ROC) curves obtained with each detector, for the proposed attack classes. To reduce over-fitting, all presented results correspond to 10-fold cross validation. Fig. 1(a) provides the results obtained for the k-CDA approach, whereas Figs. 1(b-f) show the comparative results obtained for the supervised detectors. k-CDA can correctly detect around 50% of the attacks without false alarms, and performs quite closely to the C4.5 model, which is fully supervised. Still, for some types of attacks, the achieved performance is poor, detecting about 80% of the DDoS attacks with a FPR below 10%. Both the MLP and the RF models achieve the best performance, detecting around 80% of the anomalies without false alarms. As a basic conclusion, we can claim that the k-CDA detection performance is comparable to that achieved by some of the fully supervised models, but using only 5% of labeled samples for training purposes, which is a major advantage in the practice.



(a) k-CDA, $n = 245$. (b) k-CDA, $n = 22$.

Figure 2: Improving k-CDA detection performance by feature selection.

3.2 Improving k-CDA by Feature Selection

While using a large set of input features can normally result in improved performance for some supervised approaches, it is not always the best strategy to follow, as it may negatively impact performance. Using more features increments the dimensionality of the feature space, normally introducing undesirable effects such as sparsity and training over-fitting. At the same time, using irrelevant or redundant features may diminish performance in the practice, specially for clustering approaches [6]. We show next that by carefully addressing the pre-filtering of input features by standard feature selection techniques, we can improve the detection/classification performance of k-CDA. In particular, we focus on improving the detection of DDoS attacks, which proved to be the worst detected by k-CDA in Fig. 1(a).

There are different search strategies and evaluation criteria to construct a sub-set of traffic features. We particularly apply a widely used evaluation criterion to construct a reduced sub-set of features: correlation-based evaluation. This approach basically selects sub-sets of features that are poorly correlated among each other, but highly correlated to the anomaly classes. As search strategy, we use Best-First (BF) search; BF is similar to a standard greedy exploration, but it has the ability to do backtracking, i.e., it basically keeps the previously evaluated sub-sets so as to avoid local maximum/minimum results when there is no local improvement. By running the proposed technique, we end up with a greatly reduced set of features, going from the initial 245 features to only 22. Fig. 2 shows the ROC curves for k-CDA in the detection of the DDoS attacks, using (a) the full set of features, and (b) the pruned set. Results show a clear improvement in the detection performance of DDoS attacks, partially compensating the initial performance issues observed in Fig. 1(a).

4. REFERENCES

[1] R. Fontugne et al., "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking", in *ACM CoNEXT*, 2010.

[2] A. K. Jain, "Data Clustering: 50 Years Beyond K-Means", in *Pattern Recognition Letters*, vol. 31 (8), 2010.

[3] V. Chandola et al., "Anomaly Detection: a Survey", in *Com. Sur.*, vol. 41 (3), 2009.

[4] P. Casas et al., "MINETRAC: Mining Flows for Unsupervised Analysis & Semi-Supervised Classification", in *ITC*, 2011.

[5] T. Nguyen et al., "A Survey of Techniques for Internet Traffic Classification using Machine Learning", in *IEEE Comm, Surv. & Tut.*, vol. 10 (4), pp. 56-76, 2008.

[6] P. Casas et al., "Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge", in *Com. Comm.*, vol. 35 (7), 2011.