Computer Networks 000 (2016) 1–9

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Learning combination of anomaly detectors for security domain

Martin Grill[a,b,*], Tomáš Pevný[a,b]

[a] Czech Technical University in Prague, Faculty of Electrical Engineering, Czech Republic
[b] Cisco Systems, Inc., United States

ARTICLE INFO

Article history:
Received 27 November 2015
Revised 10 May 2016
Accepted 29 May 2016
Available online xxx

Keywords:
Anomaly detection
Ensemble systems
Positive unlabeled data
Accuracy at top

ABSTRACT

This paper presents a novel technique of finding a convex combination of outputs of anomaly detectors maximizing the accuracy in τ-quantile of most anomalous samples. Such an approach better reflects the needs in the security domain in which subsequent analysis of alarms is costly and can be done only on a small number of alarms. An extensive experimental evaluation and comparison to prior art on real network data using sets of anomaly detectors of two existing intrusion detection systems shows that the proposed method not only outperforms prior art, it is also more robust to noise in training data labels, which is another important feature for deployment in practice.

1. Introduction

Increasing numbers of attacks against computing infrastructure and the critical importance of the infrastructure for enterprises drives the need to deploy progressively more sophisticated defense solutions to protect network assets. An essential component of the defense are Intrusion Detection Systems (IDS) [1] searching for evidence of ongoing malicious activities (network attacks) in network traffic crossing the defense perimeter.

Many intrusion detection systems are implemented as ensembles of relatively simple, yet heterogeneous detectors [2,3], where some of them can be specialized to particular types of intrusions, whereas others can be general anomaly detectors capable of detecting previously unseen attacks at the expense of higher false alarm rates. Such a setup has multiple advantages, including faster processing of the data stream, lower complexity of the detectors, and simpler inclusion of domain knowledge into the system. The main drawback is that combining outputs of individual detectors is a non-trivial problem. Although a vast prior art on the problem exists [4–6], we believe that peculiarities of the security domain, namely a highly imbalanced ratio of non-alarm and alarm samples in the data, lack of accurately labeled datasets, and the need of extremely low false positive rates, call for a tailored solution.

The rationale behind the above specifics is that from the user perspective each raised alarm needs to be thoroughly investigated, which is expensive and can be done only for a small number of them. Hence reporting high numbers of false positives renders any intrusion detection system useless (recall that most of the samples are legitimate). Note that using a supervised method to learn the combination may bring the expense of lower generalization, but according to our experience completely unsupervised approaches rarely have false positive rate low enough to be usable in practice. Moreover, anomaly detectors and their features are usually selected based on the experience of the designer, which is a kind of proxy for labels and surely not guaranteed to be complete.

Obtaining labeled data in security domains and in network intrusion detection especially can be difficult, time consuming, and expensive. Besides, labeled data frequently contains errors in labels of different sorts, for example some alerts might be missed and labeled as legitimate samples, or even worse, all samples of alerts of certain types might be missed and labeled as legitimate.

The above concerns motivated the main goals and contributions of this paper, which are a method of finding a convex combination of outputs of a fixed set of anomaly detectors maximizing the number of true alarms in τ-fraction of most anomalous connections (samples)[1] and an experimental study of the effect of different types of label noise in the training data on the accuracy of combinations obtained by different methods to better understand their advantages and drawbacks. Conducted experiments revealed that the proposed method is not only better than the state of the art, but also more robust with respect to various kinds of noise in labels we can expect in intrusion detection domains.

If the proposed method requires labeled data, one can ask why not use them to train a classifier and sidestep the use of anomaly

* Corresponding author.
E-mail addresses: magrill@cisco.com (M. Grill), tpevny@cisco.com (T. Pevný).

[1] Since the experimental evaluation is performed with network intrusion detection systems, the terms sample and connection are used interchangeably.

http://dx.doi.org/10.1016/j.comnet.2016.05.021
1389-1286/

detectors? The most important reason to favor anomaly detectors is that network traffic discussed in this paper is very non-stationary and anomaly detectors are good at coping with this aspect, as they can constantly update their models (see [7–9] for a review).

This paper is organized as follows: The next section formally defines the problem and presents the proposed solution. Section 3 reviews related work and algorithms that we evaluate in the experimental section. The experimental Section 4 compares the proposed solution with existing methods using sets of anomaly detectors from two different network intrusion detection systems operating on two different data sources.

2. Proposed method

Prior art in combining detectors and anomaly detectors in particular is vast [4,10], nevertheless we feel that security domains requires a tailored solution because of its prominent requirement of extremely low false positive rate. We assume that the network operator observers connections (samples) from an unknown distribution $P_o = \pi P_a + (1-\pi)P_b$ with P_a/P_b being distributions of malicious/background samples and $\pi \in [0, 1]$. The network operator uses set of m anomaly detectors on samples $\mathcal{H}_m = \{h_k : \mathcal{X} \mapsto [0,1]\}_{k=1}^m$ (w.l.o.g. it is assumed that zero means the sample is legitimate and one means the sample is malicious) and wishes to have a convex combination of anomaly detectors $\alpha = (\alpha_1, \ldots, \alpha_m)$ that would maximize the number of alarms in top τ quantile of the distribution of the combined anomaly scores. For purposes of this paper it is safe to assume that each connection (sample) is described by m-dimensional vector (an output of m anomaly detectors), which implies that distributions P_o, P_a, and P_b are defined on the m-dimensional Euclidean space.The requirements on detectors having their image in the interval [0, 1] and learning a convex combination instead of a linear one are to improve interpretability of the results as discussed in [11], but can be dropped. The same work also presents a general approach to scale the output of any anomaly detector to the interval [0, 1] reviewed in Appendix A.

With respect to the above, networks operator's goal can be written as

$$\arg\min_{\alpha\in\mathbb{R}^m} R(H_\alpha) = \underbrace{\mathbb{E}_{x\sim P_b}\left[\mathbb{1}(\alpha^{\mathrm{T}}h(x) \geq q_{\alpha,\tau})\right]}_{R^{\mathrm{fp}}(H_\alpha)} + \underbrace{\mathbb{E}_{x\sim P_a}\left[\mathbb{1}(\alpha^{\mathrm{T}}h(x) < q_{\alpha,\tau})\right]}_{R^{\mathrm{fn}}(H_\alpha)}, \tag{1}$$

subject to

$$\begin{aligned} H_\alpha(x) &= \sum_{k=1}^m \alpha_k h_k(x) = \alpha^{\mathrm{T}}h(x), \\ \mathbf{1}^{\mathrm{T}}\alpha &= 1, \\ \alpha_i &\geq 0,\ \forall i \in \{1,\ldots,m\}, \end{aligned} \tag{2}$$

where the first term in (1) is the false alarm rate, the second term is the false negative rate, and finally $q_{\alpha,\tau}$ is a τ-quantile of observed distribution of ensemble's output $\{\alpha^{\mathrm{T}}h(x)|x \in P_o\}$. The minimized term (1) captures the accuracy of a particular convex combination in top τ-quantile of its distribution, which is the goal.

In theory it would be sufficient if (1) minimizes either only the false positive rate R^{fp} or only the false negative rate R^{fn}, because each of them together with constraints (2) implies minimization of the other. But including both terms increases the robustness with respect to noise on labels, since the error and its gradient are estimated from larger number of samples implying their better estimates. This is demonstrated in Appendix B, where the combination of anomaly detectors was found by optimizing either only false positive rate or only false negative rate under constraints (2). The experiments have confirmed that optimizing the proposed (1) is indeed more robust to error in labels, which are almost inevitable in security domains. In the rest of this section we show, how to find a good solution in practice using adaptation of the method of Boyd et al. [12].

First, the true loss function (1) cannot be used in practice, since the true probability distributions P_a and P_b are not known. Therefore the expectations are replaced by their empirical estimates calculated from some labeled data used for learning the weight vector α. Below the $\mathcal{S} = \mathcal{S}_a \cup \mathcal{S}_b$ denotes the set of available samples with $\mathcal{S}_b$ being the set of background (legitimate) samples and $\mathcal{S}_a$ the set of malicious samples. The empirical estimate of (1) is therefore

$$\hat{R}(H_\alpha) = \frac{1}{|\mathcal{S}_b|}\sum_{x\in\mathcal{S}_b}\mathbb{1}\left[\alpha^{\mathrm{T}}h(x) \geq \hat{q}_{\alpha,\tau}\right] + \frac{1}{|\mathcal{S}_a|}\sum_{x\in\mathcal{S}_a}\mathbb{1}\left[\alpha^{\mathrm{T}}h(x) < \hat{q}_{\alpha,\tau}\right], \tag{3}$$

where $\hat{q}_{\alpha,\tau}$ is an empirical estimate of the true quantile $q_{\alpha,\tau}$ defined as

$$\hat{q}_{\alpha,\tau} = \arg\max_{\omega} \frac{1}{|\mathcal{S}|}\sum_{x\in\mathcal{S}}\left[\mathbb{1}(\alpha^{\mathrm{T}}h(x) \leq \omega)\right] \leq \tau. \tag{4}$$

Since the empirical loss function (3) is neither convex nor smooth, finding the optimal solution is an NP-complete problem. A usual approach is to replace indicator function $\mathbb{1}$ with a convex surrogate, for example an exponential used in this work.[2] This substitution leads to the following optimization problem

$$\begin{aligned} \arg\min_{\alpha} \quad & \frac{1}{|\mathcal{S}_b|}\sum_{x\in\mathcal{S}_b}\exp\left(\alpha^{\mathrm{T}}h(x) - \hat{q}_{\alpha,\tau}\right) \\ & + \frac{1}{|\mathcal{S}_a|}\sum_{x\in\mathcal{S}_a}\exp\left(\hat{q}_{\alpha,\tau} - \alpha^{\mathrm{T}}h(x)\right) \\ \text{subject to} \quad & \mathbf{1}^{\mathrm{T}}\alpha = 1, \\ & \alpha_i \geq 0,\ \forall i \in \{1,\ldots,l\}, \\ & \hat{q}_{\alpha,\tau} \text{ is a } \tau\text{-quantile defined in (4)}. \end{aligned} \tag{5}$$

where the optimized term (further denoted as $\hat{R}_{\exp}(H_{\boldsymbol{\alpha}})$) is an upper bound of the empirical loss function $\hat{R}(H_{\boldsymbol{\alpha}})$ defined in Eq. (3).

Nevertheless the last problem is still hard to solve, as it is not convex. Boyd et al. [12] showed how to find a good solution in polynomial time using series of convex problems. However his algorithm does not guarantee finding the global minimum, and the computational complexity prevents it from being used on problems with millions of samples. We therefore propose to solve (5) by a simple gradient algorithm summarized in Algorithm 1, which albeit not reaching the global minimum performs well, according to our experiments. In each step the current solution $\boldsymbol{\alpha_k}$ is updated by subtracting a small multiple of the gradient of (5), which is decreasing in each step to ensure convergence. The $\boldsymbol{\alpha_k}$ is then truncated to satisfy the constraints, and finally the estimate of the quantile $\hat{q}_{\alpha,\tau}$ is updated. The algorithm may find sub-optimal solutions but the experiments in Section 4 show that the solutions found are in most of the cases better than the ones of the state-of-the-art methods. Additionally, detailed discussion about the differences between the solution found by Boyd et al. and the one found by the proposed algorithm can be found in Appendix C.

The combination of detectors found by the above algorithm is optimized with respect to the *known* malware, by which we understand the malware whose samples are present in the training set

[2] The chosen convex surrogate does not have a significant impact on the solution and can be replaced by the reader's favorite choice, e.g. logistic, hinge, truncated square, etc.

Please cite this article as: M. Grill, T. Pevný, Learning combination of anomaly detectors for security domain, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.05.021

Algorithm 1: The algorithm used to solve the optimization problem (5).

Data: Set of labeled samples $x_1, \ldots, x_l \in \mathcal{S}$,
set of anomaly detectors $\mathcal{H}_m$
and δ_{min}.
Result: weights $\boldsymbol{\alpha} \in \mathbb{R}^m$
Start with equal weights $\boldsymbol{\alpha_1} = \mathbf{1}/m$;
repeat
  Set $q_{H_{\boldsymbol{\alpha}}}(\tau)$ to be τ-quantile of the distribution of H_{α_k};
  Update the step size as $\gamma_k = \frac{1}{\sqrt{k}}$;
  $\boldsymbol{\alpha_{k+1}} = \boldsymbol{\alpha_k} - \gamma_k \frac{\partial}{\partial \boldsymbol{\alpha}} \hat{R}_{\exp}(H_{\boldsymbol{\alpha_k}})$;
until $|R(H_{\boldsymbol{\alpha_k}}) - R(H_{\boldsymbol{\alpha_{k-1}}})| < \delta_{min}$;

and most of them are correctly labeled. We believe that it is very hard to draw any conclusions about the accuracy of the algorithm on malware that has never been observed. If the unknown malware is similar to the known one (e.g. using similar components or having similar behavior), then it is likely that the above optimization will help. In order to get insight to this phenomenon on real data, the experimental section compares accuracy of several algorithms on training sets with errors on labels of different types. We believe this study will help to select the right algorithms for practice.

3. Related work

There are two classes of prior art relevant to this work. The first are unsupervised methods combining outputs of anomaly detection algorithms. The second are supervised methods maximizing accuracy or some other type of loss in top τ-quantile of outputs using labeled samples. Both are briefly reviewed below (sorted from the least to the most important).

3.1. Unsupervised methods

The first explicit use of ensembles in anomaly detection [6] employed a feature bagging method to create a diverse set of anomaly detectors. Their output was fused either by summing anomaly scores of individual anomaly detectors for a given sample, which is equivalent to taking the *mean*, or by picking the k most anomalous samples from each detector (*breadth-first* strategy). In [3] authors have compared several static combination functions, namely *mean, median, minimum, maximum*, and *mean of maximum and mean* in network intrusion detection. According to their results, mean of maximum and mean [13] was the most effective.

A necessary condition to combine heterogeneous anomaly detectors is similar range of their output. This problem is tackled in [11] by using estimated cumulative distribution functions of detectors' output. The authors show that their approach outperforms other normalization strategies including HeDES [14], maximum rank [6] or sigmoid mean [15]. The experimental part of this work uses an adaptation of [11] described in Appendix A.

A hybrid solution proposed in [14] relies on artificial samples generated uniformly at random. First, several classifiers are trained to separate the artificial samples from the provided true ones, and then weights of classifiers in the combination function are determined according to their accuracy on artificial samples.

3.2. Supervised methods

Algorithms learning the combination of classifier outputs using labeled data do not differ much from general algorithms for supervised classification. But as already mentioned, for security applications the algorithms should be designed to handle large disproportions between numbers of samples in positive and negative classes, and achieve extremely low false positive rates. Such algorithms are also needed in information retrieval (although the requirement on low positive rates is not as strict), where most of the prior art comes from.

One class of relevant algorithms maximizes accuracy of ranking in top τ-quantile, which can be viewed as prioritizing the malicious samples over the legitimate ones. These algorithms (optimizing for example *Prec@k* [16] or Normalized Discounted Cumulative Gain [17]) frequently lead to non-convex optimization problems that are difficult to solve efficiently or lead to sub-optimal solutions [18] like ours. A notable exception is SVM-*perf* [19] method optimizing a convex upper bound on the number of errors among the top k items, but still the training is computationally intensive due to a large number of constraints of the quadratic program.

Another class of relevant algorithms like RankBoost [20] maximize area under ROC curve, which is equivalent to optimizing ranking. Since only top τ-quantile matters, Infinite Push [21] and Top Push [22] concentrate on the higher-ranked negatives and try to push them down.

From the above list of supervised methods RankBoost, SVM-*perf* and Top Push are compared to our method in the experimental section.

4. Experimental evaluation

The proposed combination technique was evaluated and compared to prior art using two existing network intrusion detection systems, both implemented as an ensemble of anomaly detectors with *mean* being the default combination function. The first one, described in Section 4.2, uses NetFlow [23] records, while the second one, described in Section 4.3, uses logs from HTTP proxy servers.

To compare algorithms, we use measures from information retrieval, namely *precision* and *recall*. Assuming that malware samples have positive labels, precision is the fraction of the number of malware samples classified as positive and the total number of samples classified as positives, and recall is the fraction of malware samples classified as positives and the total number of malware samples. To highlight that the detection threshold is set to 1% of the most anomalous samples, we abbreviate both measures as *Prec@1%* and *Rec@1%*. The use of precision and recall is preferred over the popular area under the Receiver Operating Characteristic curve (AUC ROC) [24], because the latter compares the algorithms in areas which are outside the region of the interest (top 1% anomalies). Moreover, precision and recall are better suited for problems with highly imbalanced classes [25].

The use of machine learning methods in security is frequently hindered by the lack of fully labeled dataset. While samples labeled as malicious are most of the time connected to some malicious behavior, it can frequently happen that some background samples are actually malicious, but the labelling oracle (analyst) has failed to recognize them. Experiments described below aim to simulate three types of noise in labels (and of course the noise-less case denoted as Non.) to investigate their effect on the learning of the combination function. The types of considered label noise are:

- The training data contains samples of all types of malicious activities, but 50% of the samples of each activity type were not recognized as malicious by the oracle (human), and therefore they are labeled as a background. This case is denoted below as anomaly label noise (*ALN*).
- Samples of some (50%) types of malicious activities are completely missing in the training data, but they are present in the testing data. Samples of remaining types of malicious activities are present in the training set, but as in the previous case the

Please cite this article as: M. Grill, T. Pevný, Learning combination of anomaly detectors for security domain, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.05.021

labeling oracle did not recognize 50% of such samples. This case is denoted as missing anomaly types (*MAT*).
- Samples of all types of malicious activities are present in the training data, but the oracle did not know 50% of types, and labeled them as background. On samples from the remaining 50% of types of malicious activities present in the training set the oracle again made a mistake and labeled them as background. This type of noise is further denoted as anomaly label noise with type mislabeling (*MLT*).

The testing set was always noiseless to allow for fair comparison and evaluate the effects.

Datasets for each intrusion detection engine are described in corresponding subsection. The available data were split so that 50% of samples were used to learn the combination of anomaly detectors and the rest for testing. This split has been repeated five times to account for the variance of the estimate.

4.1. Compared algorithms

The experimental comparison involves four unsupervised combination rules (*mean, max, rank BFS* [6], *mean rank* [6]), and four combination rules trained by supervised methods (SVM-*perf* [19], TopPush [22], RankBoost [20], and the proposed method). SVM-*perf* used L1-slack algorithm with constraint cache setting, so that 1% of positive examples was used as value of k for *Prec@k*. Regularization constant in TopPush was set to one. The proposed method (*Acc@Top*) was set to optimize the accuracy in top 1% of most anomalous samples, which means $\tau = 0.99$.

Algorithms chosen for comparison enabled comparing unsupervised methods among themselves (repeating the experiment in [3]), relevant supervised methods among themselves, and also the gain one can expect when using supervised methods even though the labels are not perfect.

4.2. Evaluation on NetFlow anomaly detection

The NetFlow anomaly detection engine [26,27] processes NetFlow [23] records exported by routers or other network traffic shaping devices. The anomaly detection engine identifies anomalous traffic using an ensemble of anomaly detection algorithms. Some of them are based on Principal component analysis [28–30], others detect abrupt changes in the behavior [31] or even use fixed rules [32]. Furthermore, there are detectors designed to detect specific type of unwanted behavior like network scans [33] or malware with domain generating algorithm [34]. In total the NetFlow anomaly detection engine uses 16 anomaly detectors. Thus the goal is to find a linear combination of these 16 anomaly detectors maximizing the accuracy in the top 1% quantile.

The evaluation used several datasets from traffic captured on the network of Czech Technical University (CTU) in Prague. The datasets and labels especially were created by three different approaches: manual labeling, infecting virtual machines, and performing real attacks against our computers within the network. In manual labeling, experienced network operator was able to successfully identify malicious activities that generated almost 10% of the total number of the connections (samples). In datasets with manually infected virtual machines[3] all their connections were labeled as malicious, whereas the rest was labeled as background. In the final dataset a network specialist run several attacks against one computer in the network. The attack vector consisted of a horizontal scan to discover open SSH ports, followed by SSH brute-force attack to break the password, and finished by SSH login and data download simulating data theft.

[3] Neeris, FastFlux and RBot were used to infect the machines [27].

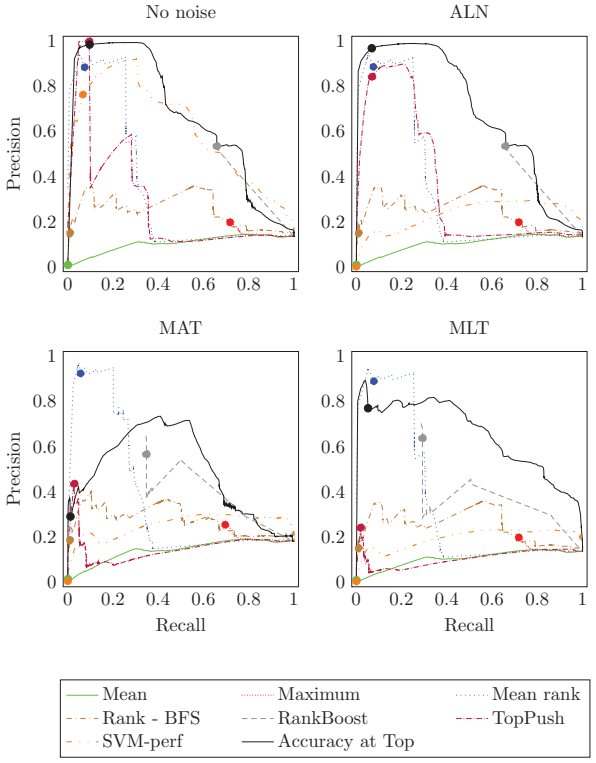

Fig. 1. PR curve comparison of algorithms with different types of label noise (described in Section 4) using the NetFlow anomaly detectors. Curves represent precision and recall values for all possible thresholds. The threshold corresponding to the 1% quantile is marked on each line with a filled circle.

Fig. 1 shows precision-recall curves for eight compared algorithms. The graphs demonstrate that the combination found by the proposed method (*Acc@Top*) most of the time dominates all other methods and fixed combination rules. Notable exceptions are cases when some types of malicious activities are completely missing in the training data (MAT) or they are incorrectly labeled (MLT). In these cases unsupervised *mean rank* combination is better on the lower recall part of the curve. This behavior suggests that different anomaly detectors detect different types of malicious activities and the supervised combination has slightly overfitted. In practical applications combining supervised and unsupervised combination rules should be used to ensure good accuracy on known malicious activities and simultaneously some generalization on unknown alerts, where the precision will be substantially smaller. Also notice that the proposed algorithm is the most robust with respect to noise from all supervised ones. SVM-*perf* is good in the noiseless case, but poor when noise of any kind is present. The TopPush is slightly more robust, but still it performed poorly with noise of MAT and MLT types, both of which are also the hardest cases. Unsupervised combination function **mean rank** performed the best among unsupervised combination functions and it was surprisingly close to supervised ones at low recall.

Precision and recall in top 1% quantile are shown in Table 1. It shows that the presented algorithm has the best or close to the best precision if we compare the supervised combination rules. As discussed above, the unsupervised *mean rank* is better in the presence of severe noise. The low recall of all algorithms except unsupervised *maximum* is caused by the high volume of malicious activities which have amounted up to 10% of the total volume of the traffic. This means that they cannot all fit into the top 1% quantile.

At the first sight RankBoost achieves the best recall of all algorithms, but a closer inspection reveals that it returns 20% of samples as those that belong in top 1%. This highly undesired

Table 1
Comparison of various combination techniques as applied to the NetFlow anomaly detectors described in Section 4.2. Each column represents precision or recall in percent for various types of noise defined at the beginning of Section 4. The best-scoring algorithm is boldfaced. Small numbers in braces below rates show the fraction of samples returned in top 1% quantile of anomaly scores. Technically this value should be equal to one, but if many samples have the same value, the algorithm returns all of them, which can results to values significantly higher than 1.0%.

	Prec@1%				*Rec@1%*			
Method	Non.	ALN	MAT	MLT	Non.	ALN	MAT	MLT
Mean	0.9 (1.0%)	0.9 (1.0%)	1.4 (1.0%)	0.9 (1.0%)	0.1 (1.0%)	0.1 (1.0%)	0.1 (1.0%)	0.1 (1.0%)
Maximum	19.8 (48.8%)	19.8 (48.7%)	25.4 (49.8%)	19.8 (48.8%)	**71.8** (48.8%)	**71.8** (48.7%)	**69.6** (49.8%)	**71.8** (48.8%)
Mean rank	88.3 (1.4%)	88.5 (1.5%)	**92.3** (1.4%)	**88.8** (1.5%)	7.4 (1.4%)	7.5 (1.5%)	5.7 (1.4%)	7.8 (1.5%)
Rank BFS	15.0 (1.0%)	15.0 (1.0%)	18.5 (1.0%)	15.0 (1.0%)	1.0 (1.0%)	1.0 (1.0%)	1.0 (1.0%)	1.0 (1.0%)
RankBoost	53.4 (16.6%)	53.5 (16.6%)	56.6 (9.8%)	63.7 (7.3%)	65.9 (16.6%)	65.9 (16.6%)	34.8 (9.8%)	29.2 (7.3%)
TopPush	**99.7** (1.3%)	84.1 (1.3%)	43.5 (1.4%)	24.2 (1.7%)	9.5 (1.3%)	6.9 (1.3%)	2.9 (1.4%)	1.9 (1.7%)
SVM-*perf*	76.2 (1.2%)	0.1 (1.0%)	0.4 (1.0%)	0.3 (1.0%)	6.7 (1.2%)	0.0 (1.0%)	0.0 (1.0%)	0.0 (1.0%)
Acc@Top	98.3 (1.0%)	**96.7** (1.0%)	29.1 (1.0%)	76.9 (1.0%)	9.7 (1.0%)	6.8 (1.0%)	1.2 (1.0%)	5.2 (1.0%)

behavior is caused by assigning the same score to multiple samples. The same phenomenon can be observed in the case of Maximum aggregation function.

4.3. Evaluation on HTTP network anomaly detection

The Cisco Cognitive Threat Analytics (CTA) [35] engine analyzes HTTP proxy logs (typically produced by proxy servers located on a network perimeter) to detect infected computers within the network. Although the logs do not contain all host traffic (only HTTP(S) requests), the information is richer than the NetFlow as each entry contains the following information extracted from HTTP headers: time of the request, source IP address, destination IP address, url, MIME type, downloaded and uploaded bytes, User-Agent identifier, etc. CTA contains more than 30 different anomaly detectors detecting anomalies according to empirical estimates of (conditional) probabilities such as P(country), P(domain|host), P(User-Agent|second level domain), etc.), time series analyses (models of user activity over time, detection of sudden changes in activity, identification of periodical requests, etc.), and HTTP specific detectors (e.g., discrepancy in HTTP User-Agent field [36]).

Evaluation data were collected from networks of 30 different companies of various sizes and types with collection period ranging from six days to two weeks. The data contains more than seven billion HTTP connections, in which Cisco analysts identified 2 666 infected users with 825 different families of malware. In total the number of HTTP connections created by the malware has reached more than 129 million. Malware connections usually represent less than 2% of the network total traffic, with a notable exception of networks with hosts infected by ZeroAccess malware [37]. ZeroAccess creates many HTTP connections that can easily reach 20% of the volume of network traffic. The other most present malware families were: Cycbot, QBot, SpyEye, BitCoinMiner, and Zbot. Malware connections were identified using multiple approaches starting with an analysis of the most anomalous HTTP logs as reported by the anomaly detection engine, malware reported by the individual network administrators, matching blacklists and other public feeds or third-party software. The rest of the logs remain unlabeled, though we are almost certain there are malware connections that have been missed.

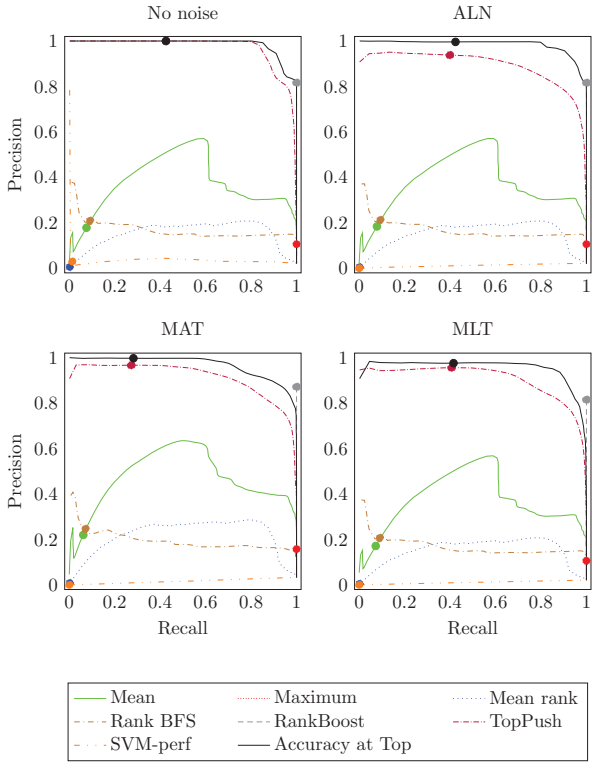

Fig. 2. PR curve comparison of various algorithms with various label noise using the HTTP anomaly detectors. Again, threshold corresponding to the 1% quantile is marked on each line with the dot.

As before, we show PR-curves of all evaluated detector combinations and types of noise in Fig. 2. We observe that the proposed *Acc@Top* method outperforms all other techniques in all cases of studied noise. Contrary to the above experiments with NetFlow analytic engine, noise does not have significant impact on supervised methods. This indicates that malicious behaviors of different types are similar in the space induced by the CTA HTTP(S) anomaly detectors. This is probably caused by the fact that all labeled malicious behaviors were in some sense connected to malware activity,

Table 2
Comparison of various combination techniques as applied to the HTTP anomaly detectors described in Section 4.3. The best-scoring algorithm is boldfaced. Small numbers in braces below rates show the fraction of samples returned in top 1% quantile of anomaly scores. Although this value should be equal to one, if many samples share the same value, the algorithm returns all of them, which can results to values significantly higher than 1.0%.

	Prec@1%				*Rec@1%*			
Method	Non.	ALN	MAT	MLT	Non.	ALN	MAT	MLT
Mean	17.9 (1.0%)	18.4 (1.0%)	22.0 (1.0%)	17.1 (1.0%)	7.6 (1.0%)	7.8 (1.0%)	6.2 (1.0%)	7.3 (1.0%)
Maximum	10.6 (20.0%)	10.6 (19.9%)	15.8 (20.9%)	10.6 (19.9%)	**100.0** (20.0%)	**100.0** (19.9%)	**100.0** (20.9%)	**100.0** (19.9%)
Mean rank	0.6 (1.0%)	0.5 (1.0%)	0.9 (1.0%)	0.5 (1.0%)	0.2 (1.0%)	0.2 (1.0%)	0.2 (1.0%)	0.2 (1.0%)
Rank BFS	20.9 (1.0%)	21.3 (1.0%)	24.7 (1.0%)	20.7 (1.0%)	9.1 (1.0%)	9.3 (1.0%)	7.1 (1.0%)	9.1 (1.0%)
RankBoost	81.7 (2.6%)	81.6 (2.6%)	87.2 (3.8%)	81.5 (2.6%)	**100.0** (2.6%)	**100.0** (2.6%)	**100.0** (3.8%)	**100.0** (2.6%)
TopPush	**100.0** (1.0%)	93.8 (1.0%)	96.7 (1.0%)	95.6 (1.0%)	42.6 (1.0%)	39.9 (1.0%)	27.3 (1.0%)	40.7 (1.0%)
SVM-*perf*	2.9 (1.0%)	0.0 (1.0%)	0.0 (1.0%)	0.0 (1.0%)	1.3 (1.0%)	0.0 (1.0%)	0.0 (1.0%)	0.0 (1.0%)
Acc@Top	**100.0** (1.0%)	**99.6** (1.0%)	**99.7** (1.0%)	**97.6** (1.0%)	42.6 (1.0%)	42.4 (1.0%)	28.2 (1.0%)	41.5 (1.0%)

for which CTA engine is designed. The curve of the *Acc@Top* suggest that even if less or more samples than 1% are requested by the operator, the precision will remain high in all scenarios. In contrast, the curve of the RankBoost method starts at high recall with lower precision suggesting that almost all malicious samples and around 20% legitimate were scored with the same maximal value. Also notice that the *mean rank* unsupervised combination function, dominating in the previous section, was in this experimental scenario superseded by simple *mean*.

Precision and recall at the top 1% are shown in Table 2, and as in the previous section RankBoost and *maximum* achieve the best recall. The causes are the same. RankBoost and maximum have returned 2.6% and 20% of samples, respectively, which is far away from the required 1%. This is again caused by assigning the same value to many samples. Contrary, the proposed *Acc@Top* meets the 1% requirement with really high precision. Its seemingly low recall is partially caused by the fraction of malicious samples being 2% of the total number of all samples. This means that the best achievable recall while meeting the requirements on returning 1% of the total number of sample is 50%.

5. Conclusion

This paper has proposed a new algorithm for finding a convex combination of anomaly detectors maximizing accuracy at τ-quantile of returned samples, which is a scenario frequently appearing in the security field. The algorithm assumes labeled data, which are difficult to obtain and rarely perfect in security domains. Therefore, an emphasis was put on the experimental study, involving two different types of intrusion detection systems, eight types of combination functions, 34 different network captures containing more than 20 million of samples of behavior of different algorithms under different types of noise.

The experimental results show that the proposed method is more accurate than prior art in finding a good combination of detectors with high accuracy in returned samples. The results also show that supervised methods can easily overfit if some type of malicious behavior is completely missing in the training data or is incorrectly labeled (mistake of labeling oracle). The severity of the overfitting depends on how much different types of malicious behavior are similar to each other. The comparison of unsupervised combination functions did not have a clear winner, since in one experimental setting *mean rank* was the best while in the second one it was *mean*.

The presented experimental results show that future efforts should be directed toward finding methods combining good properties of both supervised and unsupervised combination functions.

Appendix A. Scaling outputs of anomaly detectors

Generally, individual anomaly detectors need not generate anomaly scores of the same scale. This causes problems during the combination process, since one or more detectors could be inadvertently favored. Therefore, the anomaly scores of the individual detectors are normalized using the *gaussian scaling* proposed by Kriegel et al. [11]:

$$\tilde{h}(x) = \max\left\{0, erf\left(\frac{h(x) - \mu_h}{\sigma_h\sqrt{2}}\right)\right\}, \tag{A.1}$$

where the $\tilde{h}(x)$ is the normalization of the anomaly score $h(x)$ assigned to the observation $x \in \mathcal{X}$ by anomaly detector h. The used Gaussian Error Function $erf()$ is monotone and thus ranking stable. The μ_h and σ_h are the mean and the standard deviation of the anomaly scores returned by the anomaly detector h. This transforms the anomaly scores of individual anomaly detectors into probability estimates, where the probability of zero represent normal observation, aligned with the predictive model, whereas one indicates highly anomalous observation. These are therefore directly comparable and can be aggregated using a number of combination techniques [11].

Appendix B. Optimizing only false positives or false negatives

To demonstrate the advantage of minimizing both false positive and false negative rates in the objective function $\hat{R}_{\text{exp}}(H_\alpha)$ (5), we have evaluated two additional variants of the objective function with only the false negative part $\hat{R}^{fn}_{\text{exp}}$ (*Acc@Top-FN*) and false positive part $\hat{R}^{fp}_{\text{exp}}$ (*Acc@Top-FP*) using both NetFlow (Table B.3) and CTA (Table B.4) anomaly detection systems. As can be seen in Table B.3, using only one part of the criterion results in substantially decreased efficacy in the NetFlow scenario. Additionally, the false

Please cite this article as: M. Grill, T. Pevný, Learning combination of anomaly detectors for security domain, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.05.021

Table B3
Comparison of three variants of the proposed criterion, each used to train an ensemble for the NetFlow anomaly detection system. Small numbers in braces below rates show the fraction of samples returned in top 1% quantile of anomaly scores. Although this value should be equal to one, if many samples share the same value, the algorithm returns all of them, which can results to values significantly higher than 1.0%.

	Prec@1%				*Rec@1%*			
Method	Non.	ALN	MAT	MLT	Non.	ALN	MAT	MLT
Acc@Top	**98.3** (1.0%)	**96.7** (1.0%)	**29.1** (1.0%)	**76.9** (1.0%)	9.7 (1.0%)	6.8 (1.0%)	1.2 (1.0%)	5.2 (1.0%)
Acc@Top-FP	13.4 (100%)	13.4 (100%)	18.1 (100%)	13.4 (100%)	**100** (100%)	**100** (100%)	**100** (100%)	**100** (100%)
Acc@Top-FN	0.1 (1.9%)	0.1 (1.9%)	15.7 (1.4%)	21.7 (2.4%)	0.0 (1.9%)	0.0 (1.9%)	0.7 (1.4%)	1.6 (2.4%)

Table B4
Similarly to Table B.3, the table presents a comparison of three variants of the proposed criterion used on the CTA anomaly detection system.

	Prec@1%				*Rec@1%*			
Method	Non.	ALN	MAT	MLT	Non.	ALN	MAT	MLT
Acc@Top	**100** (1.0%)	**99.6** (1.0%)	**99.7** (1.0%)	**97.6** (1.0%)	**42.6** (1.0%)	**42.4** (1.0%)	28.2 (1.0%)	**41.5** (1.0%)
Acc@Top-FP	98.1 (1.0%)	1.2 (11.7%)	52.7 (7.8%)	0.6 (26.2%)	41.7 (1.0%)	11.2 (11.7%)	19.3 (7.8%)	25.4 (26.2%)
Acc@Top-FN	87.2 (1.0%)	87.5 (1.0%)	88.5 (1.2%)	86.0 (1.1%)	37.1 (1.0%)	37.2 (1.0%)	**29.2** (1.2%)	38.6 (1.1%)

positive variant (*Acc@Top-FP*) results in all the samples having the same, zero value anomaly score. The results on the CTA anomaly detection engine (Table B.4) are slightly better, but still the proposed *Acc@Top* outperforms both other variants in all label noise scenarios.

Appendix C. Comparison with state-of-the-art

Although the algorithm of Boyd et al. [12] is currently considered to be the state of the art for optimizing the accuracy at top, it is not guaranteed to find the global minimum. Its biggest advantage is in finding the solution by solving series of convex sub-problems; therefore it is bound to always find the same solution. But this limits the algorithm to be applicable only to small problems, since its complexity grows as of $O(n^4)$, where n is the number of training samples, and the solver of the convex sub-problems has the complexity of $O(n^3)$. In contrast, the proposed algorithm is essentially a stochastic descent algorithm, which has been proved to work well on problems with large number of samples.

In order to investigate how the solutions of both algorithms differ, we have created an artificial problem, which we think well models the application scenario of finding a convex combination of outputs of anomaly detectors. The problem was set to find the optimal combination of two anomaly detectors to optimize accuracy at the 20% quantile. The anomaly scores of the anomaly detectors for both, training and testing data, were generated using a set of normal distributions.[4]

The decision boundaries corresponding to solutions of both algorithms (shown in Fig. C.4) are very different, since Boyd et al.'s algorithm uses the output of one anomaly detector whereas ours uses both detectors. Corresponding PR-curves, shown in Fig. C.3, further reveal that on the training set (solid lines) Boyd et al.'s algorithm (*Boyd*) is better at the point of the interest, but the proposed method (*Acc@Top*) is better on a wider range of operating points, which suggests that it would behave better on unknown data. This is experimentally confirmed by PR-curves on the testing data (dashed lines), where the proposed algorithm dominates. Although, theoretically, this can be due to overfitting which can be solved by training on a larger training set, if available, this solution would be difficult in practice due to the prohibitive complexity of Boyd et al.'s algorithm.

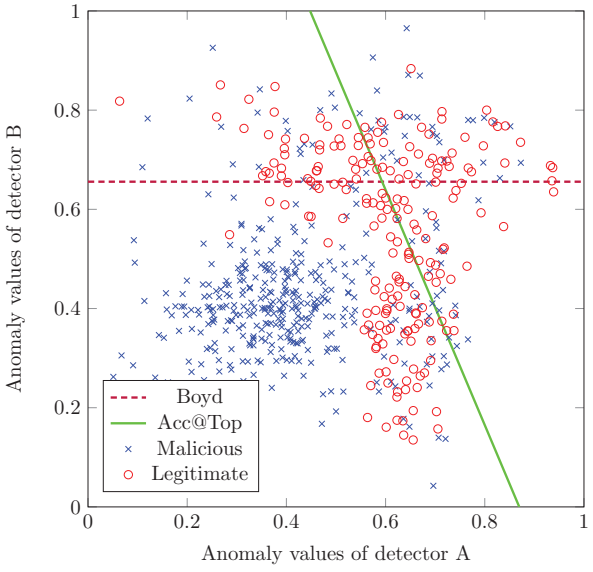

Fig. C3. Visualization of the artificial problem containing two anomaly detectors. The position of both anomalous and malicious samples is given by the anomaly score of the both A and B anomaly detectors. The solid and dashed lines represent decision boundaries of *Acc@Top* and *Boyd* algorithms respectively.

[4] The legitimate samples were drawn from uniform distribution (100 samples) to simulate noise of the anomaly detectors and $\mathcal{N}([0.4, 0.4], [0.01, 0.01]\mathbb{I}_2)$ (300 samples), where the $\mathbb{I}_2$ denotes the 2×2 identity matrix. The malicious samples were drawn from $\mathcal{N}([0.6, 0.7], [0.03, 0.003]\mathbb{I}_2)$ (100 samples) and $\mathcal{N}([0.7, 0.4], [0.003, 0.03]\mathbb{I}_2)$ (100 samples).

Please cite this article as: M. Grill, T. Pevný, Learning combination of anomaly detectors for security domain, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.05.021

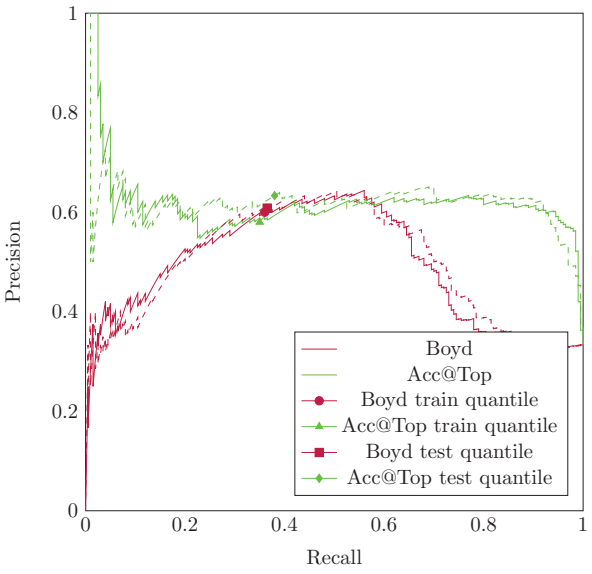

Fig. C4. PR-curves of the *Acc@Top* and the optimum found by the *Boyd* algorithm using the training (solid line) and testing (dashed line) data generated from the artificial problem. Threshold corresponding to the 20% quantile is marked on each curve.

References

[1] K. Scarfone, P. Mell, Guide to intrusion detection and prevention systems (IDPS), Technical Report 800-94, NIST, US Department of Commerce, 2007.
[2] G. Giacinto, F. Roli, Intrusion detection in computer networks by multiple classifier systems, in: Proceedings of the 16th International Conference on Pattern Recognition (ICPR), Volume 2, IEEE Press, 2002, pp. 390–393.
[3] S. Shanbhag, T. Wolf, Accurate anomaly detection through parallelism, Network, IEEE 23 (1) (2009) 22–28.
[4] L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, John Wiley, 2004.
[5] L.I. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 281–286.
[6] A. Lazarevic, V. Kumar, Feature bagging for outlier detection, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data mining, ACM, 2005, pp. 157–166.
[7] T. Pevný, Loda: lightweight on-line detector of anomalies, Mach Learn. (2015) 1–30.
[8] A. Coluccia, A. D'Alconzo, F. Ricciato, Distribution-based anomaly detection via generalized likelihood ratio test: a general maximum entropy approach, Computer Networks 57 (17) (2013) 3446–3462.
[9] C. Callegari, A. Coluccia, A. D'Alconzo, W. Ellens, S. Giordano, M. Mandjes, M. Pagano, T. Pepe, F. Ricciato, P. Zuraniewski, in: Datatraffic Monitoring and Analysis, Springer-Verlag, Berlin, Heidelberg, 2013, pp. 148–183.
[10] C.C. Aggarwal, Outlier ensembles: position paper, ACM SIGKDD Explor. Newslett. 14 (2) (2013) 49–58.
[11] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek, Interpreting and unifying outlier scores, in: 11th SIAM International Conference on Data Mining (SDM), Mesa, AZ, 42, SIAM, 2011.
[12] S. Boyd, C. Cortes, M. Mohri, A. Radovanovic, Accuracy at the top, in: Advances in Neural Information Processing Systems, 2012, pp. 953–961.
[13] P.F. Evangelista, M.J. Embrechts, B.K. Szymanski, Data fusion for outlier detection through pseudo-roc curves and rank distributions, in: Neural Networks, 2006. IJCNN'06. International Joint Conference on, IEEE, 2006, pp. 2166–2173.
[14] H.V. Nguyen, H.H. Ang, V. Gopalkrishnan, Mining outliers with ensemble of heterogeneous detectors on random subspaces, in: Database Systems for Advanced Applications, Springer, 2010, pp. 368–383.
[15] J. Gao, P.-N. Tan, Converting output scores from outlier detection algorithms into probability estimates, in: Data Mining, 2006. ICDM'06. Sixth International Conference on, IEEE, 2006, pp. 212–221.
[16] B. McFee, G.R. Lanckriet, Metric learning to rank, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 775–782.
[17] H. Valizadegan, R. Jin, R. Zhang, J. Mao, Learning to rank by optimizing ndcg measure, in: Advances in Neural Information Processing Systems, 2009, pp. 1883–1891.
[18] Q. Le, A. Smola, Direct optimization of ranking measures, arXiv preprint arXiv:0704.3359 (2007).
[19] T. Joachims, A support vector method for multivariate performance measures, in: Proceedings of the 22nd International Conference on Machine Learning, ACM, 2005, pp. 377–384.
[20] Y. Freund, R. Iyer, R.E. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences, J. Mach. Learning Res. 4 (2003) 933–969.
[21] A. Rakotomamonjy, Sparse support vector infinite push, arXiv preprint arXiv:1206.6432 (2012).
[22] N. Li, R. Jin, Z.-H. Zhou, Top rank optimization in linear time, in: Advances in Neural Information Processing Systems, 2014, pp. 1502–1510.
[23] E.B. Claise, in: Cisco Systems NetFlow Services export, Version 9, 2004.
[24] T. Fawcett, An introduction to roc analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.
[25] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 233–240.
[26] M. Rehák, M. Pěchouček, M. Grill, J. Stiborek, K. Bartoš, P. Čeleda, Adaptive multiagent system for network traffic monitoring, IEEE Intell. Syst. (3) (2009) 16–25.
[27] S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Comput. Secur. 45 (2014) 100–123.
[28] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, in: ACM SIGCOMM Computer Communication Review, 34, ACM, 2004, pp. 219–230.
[29] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, in: ACM SIGCOMM Computer Communication Review, 35, ACM, 2005, pp. 217–228.
[30] T. Pevný, M. Rehák, M. Grill, Detecting anomalous network hosts by means of pca, in: Information Forensics and Security (WIFS), 2012 IEEE International Workshop on, 2012, pp. 103–108.
[31] L. Ertoz, E. Eilertson, A. Lazarevic, P.-N. Tan, V. Kumar, J. Srivastava, P. Dokas, Minds - minnesota intrusion detection system, in: Next Generation Data Mining, MIT Press, 2004.
[32] K. Xu, Z.-L. Zhang, S. Bhattacharyya, Profiling internet backbone traffic: behavior models and applications, in: ACM SIGCOMM Computer Communication Review, 35, ACM, 2005, pp. 169–180.
[33] A. Sridharan, T. Ye, S. Bhattacharyya, Connectionless port scan detection on the backbone, in: Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International, IEEE, 2006, pp. 10–pp.
[34] M. Grill, I. Nikolaev, V. Valeros, M. Rehák, Detecting dga malware using netflow, in: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, 2015, pp. 1304–1309.
[35] Cisco Systems, in: CTA cisco cognitive threat analytics on cisco cloud web security, 2014–2015.(http://www.cisco.com/c/en/us/solutions/enterprise-networks/cognitive-threat-analytics).
[36] M. Grill, M. Rehák, Malware detection using http user-agent discrepancy identification, in: Information Forensics and Security (WIFS), 2014 IEEE International Workshop on, 2014, pp. 221–226.
[37] J. Wyke, The zeroaccess botnet–mining and fraud for massive financial gain, 2012.

Please cite this article as: M. Grill, T. Pevný, Learning combination of anomaly detectors for security domain, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.05.021

Martin Grill is a researcher at Cognitive Threat Analytics at Cisco. His research focuses mainly on network anomaly detection, ensemble systems and network event classification. Prior to Cisco, Martin was a researcher at Czech Technical University in Prague and CESNET, z.s.p.o developing a NetFlow based network behaviour anomaly detection system. Martin Grill holds master degree in Software development at the Faculty of Nuclear Sciences and Physical Engineering of the Czech Technical University in Prague. Currently he is pursuing his PhD at the Department of Computer Science of Czech Technical University in Prague.

Tomáš Pevný holds the position of researcher at Czech Technical University of Prague. He received his PhD in Computer Sciences from State University of New York in Binghamton in Computer Science at 2008 and MS in Computer Sciences from School of Nuclear Sciences and Physical Engineering at Czech Technical University in Prague in 2003. In 2008–2009, he did his post-doc at Gipsa-lab in Grenoble, France. His research interests are applications of non-parametric statistics (machine learning, data modeling) with focuses on steganography, steganalysis, and intrusion detection.

Please cite this article as: M. Grill, T. Pevný, Learning combination of anomaly detectors for security domain, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.05.021