Automatic Feature Selection for Anomaly Detection

Marius Kloft
Technical University of Berlin
Dept. of Computer Science
Berlin, Germany
kloft@cs.tu-berlin.de

Ulf Brefeld
Technical University of Berlin
Dept. of Computer Science
Berlin, Germany
brefeld@cs.tu-berlin.de

Patrick Düssel
Fraunhofer Institute FIRST
Intelligent Data Analysis
Berlin, Germany
patrick.duessel
@first.fraunhofer.de

Christian Gehl
Fraunhofer Institute FIRST
Intelligent Data Analysis
Berlin, Germany
christian.gehl
@first.fraunhofer.de

Pavel Laskov[*]
Fraunhofer Institute FIRST
Intelligent Data Analysis
Berlin, Germany
pavel.laskov
@first.fraunhofer.de

ABSTRACT

A frequent problem in anomaly detection is to decide among different feature sets to be used. For example, various features are known in network intrusion detection based on packet headers, content byte streams or application level protocol parsing. A method for automatic feature selection in anomaly detection is proposed which determines optimal mixture coefficients for various sets of features. The method generalizes the support vector data description (SVDD) and can be expressed as a semi-infinite linear program that can be solved with standard techniques. The case of a single feature set can be handled as a particular case of the proposed method. The experimental evaluation of the new method on unsanitized HTTP data demonstrates that detectors using automatically selected features attain competitive performance, while sparing practitioners from a priori decisions on feature sets to be used.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; I.2.6 [**Artificial Intelligence**]: Learning—*Parameter learning*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation, Feature evaluation and selection*

General Terms

Algorithms, Experimentation, Security

Keywords

Machine learning, anomaly detection, support vector data description, feature selection, multiple kernel learning, intrusion detection, network security

1. INTRODUCTION

The main merit of anomaly detection techniques is their ability to detect previously unknown attacks. One might think that the collective expertise amassed in the computer security community rules out major outbreaks of "genuinely novel" exploits. Unfortunately, a wide-scale deployment of efficient tools for obfuscation, polymorphic mutation and encryption results in an exploding variability of attacks. Although being only "marginally novel", such attacks quite successfully defeat signature-based detection tools. This reality brings anomaly detection back into the research focus of the security community.

The majority of anomaly detection methods use some form of machine learning techniques to devise a model of normality from observed normal traffic. They may vary in features being used but share the general idea of measuring anomality of new objects by their distance (in some metric space) from the learned model of normality, historically also known as "the sense of self" [2]. Apart from this theoretical observation, in practice the effectiveness of anomaly detection crucially depends on the choice of features. Various features have been deployed for network intrusion detection, such as raw values of IP and TCP protocol headers [6, 7], time and connection windows [5], byte histograms and n-grams [15, 14], and "bag-of-tokens" language models [10, 11]. While packet header based features have been shown to be effective against probes and scans (which many practitioners consider uninteresting anyway), other kinds of attacks, e.g. remote buffer overflows, require more advanced payload processing techniques. The right kind of features for a particular application has always been considered as the matter of a judicious choice (or trial and error).

But what if this decision is really difficult to make? Given the choice of several kinds of features, a poor a priori decision would lead to an inappropriate model of normality being learned. A better strategy is to have a *learning algorithm itself* decide which set of features is the best. The reason

[*]Pavel Laskov is also affiliated with University of Tübingen, Wilhelm-Schickard-Institute, Germany.

AISec'08, October 27,2008, Alexandria, Virginia, USA.
Copyright 2008 ACM 978-1-60558-291-7/08/10 ...$5.00.

for that is that learning algorithms find models with optimal generalization properties, i.e. the ones that are valid not only for observed data but also for the data to be dealt with in the future. The a priori choice of features may bias the learning process and lead to worse detection performance. By leaving this choice to the learning algorithm, the possibility of such bias is eliminated.

The problem of automatic feature selection has been well studied in the machine learning community in the context of classification, i.e. choosing among two or more labels to be assigned to events [4; 8; 16; 3, e.g.]. The classification setup, however, is hardly appropriate for anomaly detection since the training data contains examples of only one class, the normal traffic. To enable automatic feature selection for anomaly detection, we derive an appropriate formulation for one-class-classification, a particular kind of anomaly detection using support vector data description (SVDD) [13]. Our approach generalizes the vanilla SVDD that is contained as a special case when only a single feature vector is used. The solution to our feature selection problem is a sparse linear combination of features that realizes a minimal-volume description of the data. The underlying optimization can be phrased as a semi-infinite linear program and solved by standard techniques. A further advantage of the proposed method is that it allows training on contaminated data by limiting the impact of single events on the learned model. To emphasize this feature, we have carried out experiments on *unsanitized* training data obtained "from the wire".

Our paper is structured as follows. Section 2 reviews the problem setting of classical one-class anomaly detection with only a single feature mapping. We derive our feature selection SVDD in Section 3 where we also state the final optimization problem. Section 4 reports on empirical results and Section 5 concludes.

2. ONE-CLASS ANOMALY DETECTION

In this section, we briefly review the classical support vector data description (SVDD) [13]. We are given a set of n *normal* inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and a function $\phi : \mathcal{X} \to \mathcal{F}$ extracting features out of the inputs. For instance, $\mathbf{x}_i$ may refer to the i-th recorded request and $\phi(\mathbf{x}_i)$ may encode the vector of bigrams occurring in $\mathbf{x}_i$.

The goal in anomaly detection is to find a description of the normal data such that anomalous data can be easily identified as outliers. In our one-class scenario, this translates to finding a minimal enclosing hypersphere (i.e., center $\vec{w}$ and radius R) that contains the normal input data [13]. Given the function

$$f(\mathbf{x}) = \|\phi(\mathbf{x}) - \vec{w}\|^2 - R^2,$$

the boundary of the ball is described by the set $\{\mathbf{x} : f(\mathbf{x}) = 0 \wedge \mathbf{x} \in \mathcal{X}\}$. That is, the parameters of f are to be chosen such that $f(\mathbf{x}) < 0$ for normal data and $f(\mathbf{x}) > 0$ for anomalous points. The center $\vec{w}$ and the radius R can be computed accordingly by solving the following optimization problem [13]

$$\begin{aligned} \min_{\vec{w},R,\vec{\xi}} \quad & R^2 + \eta \sum_i \xi_i \\ \text{s.t.} \quad & \forall_{i=1}^n : \ \|\phi(\mathbf{x}_i) - \vec{w}\|^2 \leq R^2 + \xi_i \\ & \forall_{i=1}^n : \ \xi_i \geq 0. \end{aligned}$$

The trade-off parameter $\eta > 0$ adjusts point-wise violations of the hypersphere. That is, a concise description of the data might benefit from omitting some data points in the computation of the solution. Discarded data points induce slack that is absorbed by variables ξ_i. Thus, in the limit $\eta \to \infty$, the hypersphere will contain all input examples irrespectively of their utility for the model and $\eta \to 0$ implies $R \to 0$ and the center $\vec{w}$ reduces to the centroid of the data. In general, model selection strategies such as cross-validation are necessary not only to find optimal user-defined parameters such as the trade-off η, but also to choose an appropriate feature representation ϕ. In the next section, we detail an approach to automatically select the optimal linear combination of several feature mappings.

3. AUTOMATIC FEATURE SELECTION FOR ANOMALY DETECTION

In this section, we present our approach to automatic feature selection for anomaly detection. Our approach generalizes the support vector data description (SVDD) [13] that is obtained as a special case when only a single feature mapping is given. In contrast to the previous section we are now given k feature mappings $\phi_1, \ldots, \phi_k$ with $\phi_j : \mathcal{X} \to \mathcal{F}_j$, $1 \leq j \leq k$, in addition to the n input instances $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$. For instance, $\mathbf{x}_i$ may refer to the i-th recorded request and $\phi_j(\mathbf{x}_i)$ may encode the j-gram feature vector of $\mathbf{x}_i$.

Besides finding a center and radius, the operational goal is now to learn a linear combination of the given feature mappings to realize the minimal model. This can be expressed equivalently as an embedding of $\phi_1, \ldots, \phi_k$ with mixture coefficients $\beta_1, \ldots, \beta_k$. That is, the model f is now given by

$$f(\mathbf{x}) = \left\| \begin{pmatrix} \sqrt{\beta_1}\phi_1(\mathbf{x}) \\ \vdots \\ \sqrt{\beta_k}\phi_k(\mathbf{x}) \end{pmatrix} - \begin{pmatrix} \vec{w}_1 \\ \vdots \\ \vec{w}_k \end{pmatrix} \right\|^2 - R^2$$

and the above SVDD optimization problem can be generalized accordingly to multiple feature mappings. In the following we write $\vec{w} = (\vec{w}_1, \ldots, \vec{w}_k)^\mathsf{T}$ to avoid cluttering the notation unnecessarily. We are now ready to state the *primal* optimization problem for one-class anomaly detection with multiple feature mappings.

Optimization Problem 1 (primal). *Given n instances $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, k feature mappings $\phi_1, \ldots, \phi_k$ with $\phi_j : \mathcal{X} \to \mathcal{F}_j$, and $\eta > 0$. The primal feature selection SVDD optimization problem is given by*

$$\begin{aligned} \min_{\vec{w},R,\vec{\xi},\vec{\beta}} \quad & R^2 + \eta \sum_{i=1}^n \xi_i \\ \textit{s.t.} \quad & \forall_{i=1}^n : \quad \|\psi_\beta(\mathbf{x}_i) - \vec{w}\|^2 \leq R^2 + \xi_i \qquad (1) \\ & \forall_{i=1}^n : \ \xi_i \geq 0 \\ & \forall_{j=1}^k : \ \beta_j \geq 0 \\ & \sum_{i=1}^k \beta_j = 1, \end{aligned}$$

where $\psi_\beta(\mathbf{x}_i) = (\sqrt{\beta_1}\phi_1(\mathbf{x}_i), \ldots, \sqrt{\beta_k}\phi_k(\mathbf{x}_i))^\mathsf{T}$.

The last constraint in Optimization Problem 1 requires the mixing coefficients to sum to one which corresponds to an L_1 regularization. We thus promote sparsity and aim at selecting subsets of the k feature mappings. From a geometrical

point-of-view, Optimization Problem 1 can be understood as follows: Redundant and deceptive feature mappings lead to arbitrary and widespread data representations and thus render concise spherical descriptions impossible. Such inappropriate embeddings will be penalized by vanishing mixing coefficients β_j. On the contrary, useful feature mappings are promoted during the optimization, hence enforcing concise data descriptions.
Unfortunately, we cannot solve Optimization Problem 1 directly since it is not convex due to non-linear dependencies between $\vec{\beta}$ and $\vec{w}$, which we see by expanding the term

$$\begin{aligned} \|\psi_\beta(\mathbf{x}_i) - \vec{w}\|^2 &= \sum_{j=1}^{k} \beta_j \langle \phi_j(\mathbf{x}_i), \phi_j(\mathbf{x}_i) \rangle && \text{(2a)} \\ &\quad - 2 \sum_{j=1}^{k} \langle \sqrt{\beta_j} \phi_j(\mathbf{x}_i), \vec{w}_j \rangle + \langle \vec{w}, \vec{w} \rangle. && \text{(2b)} \end{aligned}$$

Moreover, setting $\phi_j(\mathbf{x}) = \vec{0}$ for $1 \le j \le k$, Equation (1) can be solved for the radius R which can be expressed in terms of the center $\vec{w}$ and a non-negative offset ϵ^2,

$$\|\vec{0} - \vec{w}\|^2 \le R^2 + \xi_i \quad \Rightarrow \quad R^2 = \langle \vec{w}, \vec{w} \rangle + \epsilon^2. \qquad (3)$$

A remedy to the nonlinearity in $\vec{w}$ and $\vec{\beta}$ is a variable substitution by $\vec{v}_j = \sqrt{\beta_j} \vec{w}_j$. Together with Equations (2) and (3) we obtain a convex analogue of the optimization problem (1) given by

$$\begin{aligned} \min_{\vec{v}, \epsilon, \vec{\xi}, \vec{\beta}} \quad & \epsilon^2 + \sum_{j=1}^{k} \frac{1}{\beta_j} \langle \vec{v}_j, \vec{v}_j \rangle + \eta \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \forall_{i=1}^{n} : \ \epsilon^2 + \xi_i \ge \sum_{j=1}^{k} \beta_j \langle \phi_j(\mathbf{x}_i), \phi_j(\mathbf{x}_i) \rangle \\ & \qquad\qquad\qquad - 2 \sum_{j=1}^{k} \langle \phi_j(\mathbf{x}_i), \vec{v}_j \rangle \\ & \forall_{i=1}^{n} : \xi_i \ge 0 \, ; \quad \forall_{j=1}^{k} : \beta_j \ge 0 \, ; \quad \sum_{j=1}^{k} \beta_j = 1. \end{aligned}$$

The above optimization problem is convex and has only linear constraints that can now be integrated into the objective by the Lagrange Theorem. For any valid $\vec{\beta} \in \{\vec{\beta}' : \sum_j \beta'_j = 1 \wedge \beta'_j \ge 0\}$ we obtain a partial Lagrangian by introducing nonnegative Lagrange multipliers $\vec{\alpha}, \vec{\mu} \ge 0$, leading to the Lagrangian L that needs to be minimized.

$$\begin{aligned} L(\vec{v}, \epsilon, \vec{\xi}, \vec{\beta}, \vec{\alpha}, \vec{\mu}) = \epsilon^2 + \sum_{j=1}^{k} \frac{1}{\beta_j} \langle \vec{v}_j, \vec{v}_j \rangle + \eta \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \mu_i \xi_i \\ - \sum_{i=1}^{n} \alpha_i \Bigg(- \sum_{j=1}^{k} \beta_j \langle \phi_j(\mathbf{x}_i), \phi_j(\mathbf{x}_i) \rangle \\ + 2 \sum_{j=1}^{k} \langle \phi_j(\mathbf{x}_i), \vec{v}_j \rangle + \epsilon^2 + \xi_i \Bigg). \end{aligned}$$

The Lagrangian reaches its minimal value when it is minimized with respect to the primal variables $\vec{v}, \epsilon, \vec{\beta}, \vec{\xi}$ and maximized with respect to the Lagrange multipliers; hence, the optimum is found at a saddle-point. Setting the partial derivatives with respect to the primal variables $\epsilon, \vec{v}$, and $\vec{\xi}$ to zero yields

$$\begin{aligned} \frac{\delta L}{\delta \epsilon} &\overset{!}{=} 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i = 1 && \text{(4a)} \\ \frac{\delta L}{\delta \vec{v}} &\overset{!}{=} 0 \quad \Rightarrow \quad \vec{v}_j = \beta_j \sum_{i=1}^{n} \alpha_i \phi_j(\mathbf{x}_i), \quad 1 \le j \le k && \text{(4b)} \\ \frac{\delta L}{\delta \vec{\xi}} &\overset{!}{=} 0 \quad \Rightarrow \quad \eta - \mu_i - \alpha_i = 0, \qquad 1 \le i \le n. && \text{(4c)} \end{aligned}$$

Equation (4c) together with the nonnegativity constraints on α_i and μ_i leads to the so-called box-constraints $0 \le \alpha_i \le \eta$. Resubstitution of Equations (4) into the primal Lagrangian removes its dependence on the primal variables:

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{k} \beta_j K_j(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,\ell=1}^{n} \alpha_i \alpha_\ell \sum_{j=1}^{k} \beta_j K_j(\mathbf{x}_i, \mathbf{x}_\ell).$$

Together with the minimization over $\vec{\beta}$ we resolve the following min-max problem

$$\begin{aligned} \min_{\vec{\beta}} \ \max_{\vec{\alpha}} \quad & L(\alpha) && \text{(5)} \\ \text{s.t.} \quad & \forall_{i=1}^{n} : \ 0 \le \alpha_i \le \eta \, ; \quad \sum_{i=1}^{n} \alpha_i = 1 \\ & \forall_{j=1}^{k} : \ \beta_j \ge 0 \, ; \quad \sum_{j=1}^{k} \beta_j = 1, \end{aligned}$$

where we introduce kernel $K_j(\mathbf{x}, \mathbf{x}') = \langle \phi_j(\mathbf{x}), \phi_j(\mathbf{x}') \rangle$ for $1 \le j \le k$. To efficiently optimize the above optimization problem, we translate it into an equivalent semi-linear infinite program (SILP). The idea behind this transformation is as follows: Let $\Omega(\vec{\alpha}, \vec{\beta})$ be the objective function in Equation (5) and suppose $\vec{\alpha}^*$ is chosen optimally. Then it holds $\Omega(\vec{\alpha}^*, \vec{\beta}) \ge \Omega(\vec{\alpha}, \vec{\beta})$ for all $\vec{\alpha}$ and $\vec{\beta}$. Hence we can equivalently minimize an upper bound θ on the optimal value and by doing so we arrive at the final Optimization Problem 2.

Optimization Problem 2 (SILP). *Given n instances $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, either k feature mappings $\phi_1, \ldots, \phi_k$ or alternatively k kernel functions $K_1, \ldots, K_k : \mathcal{X} \times \mathcal{X} \to \Re$ with $K_j(\mathbf{x}, \mathbf{x}') = \langle \phi_j(\mathbf{x}), \phi_j(\mathbf{x}') \rangle$, and $\eta > 0$. The SILP formulation of the feature selection SVDD is given by*

$$\begin{aligned} \min_{\vec{\beta}, \theta} \quad & \theta \\ s.t. \quad & \theta \ge \sum_{j=1}^{k} \beta_j \left(\sum_{i=1}^{n} \alpha_i K_j(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,\ell=1}^{n} \alpha_i \alpha_\ell K_j(\mathbf{x}_i, \mathbf{x}_\ell) \right) \\ & \forall \, \vec{\alpha} \in \Re^n : \quad 0 \le \alpha_i \le \eta \, , \quad \sum_{i=1}^{n} \alpha_i = 1; \\ & \forall_{j=1}^{k} : \ \beta_j \ge 0 \, ; \quad \sum_{j=1}^{k} \beta_j = 1. \end{aligned}$$

Optimization Problem 2 is equivalent to the primal Optimization Problem 1 and can be efficiently optimized by standard techniques [12].

4. EMPIRICAL EVALUATION

In this section we empirically evaluate the proposed feature selection SVDD on real HTTP network traffic recorded at

```
GET /openworx.php?key=malware+behavior HTTP/1.1\r\n
Host: www.first.fraunhofer.de\r\n
Connection: keep-alive\r\n
Keep-alive: 300\r\n
User-Agent: Mozilla/5.0 (Windows; Windows NT 5.1;
 en-US) Gecko/20070312 Firefox/1.5.0.11\r\n
Cookie: owx_ecrm_keks=b604613a489d40\r\n
Referer: http://www.first.fraunhofer.de/ida\r\n
Accept: image/png,*/*;q=0.5\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Accept-Encoding: gzip,deflate\r\n
```

Figure 1: An exemplary HTTP request.

the Fraunhofer FIRST institute. The unsanitized dataset contains a sample of 2500 normal HTTP requests drawn randomly from two months of incoming HTTP traffic. We injected 30 instances of 10 different attacks taken from recent exploits in the Metasploit framework[1] and a Nessus HTTP scan. All exploits (6 buffer overflow attacks and 4 PHP vulnerabilities) were normalized to match the tokens and frequent attributes of normal HTTP requests such that the malicous payload provides the only indicator for identifying the attacks.

4.1 Feature Extraction

We consider six different feature sets extracted from the raw data. Three of these feature sets are based on a *sequential representation* of byte streams comprising HTTP requests as depicted in Figure 1. The remaining three feature sets correspond to a *token representation* of the HTTP request. The latter is obtained by running requests through an HTTP protocol analyzer constructed with *binpac* [9], and collecting the analysis results in a token-attribute sequence. The tokens in this sequence correspond to keywords of the HTTP protocol whereas the attributes consist of byte sequences associated with these keywords. Figure 2 visualizes the corresponding token-attribute structure of the request in Figure 1. For each of the two representations, we extract the following features from the HTTP requests:

3-gram occurrence features
The feature functions ϕ_{occ}^{seq} and ϕ_{occ}^{tok} register the occurrence of particular byte 3-grams for the sequential and the token representation, respectively. Each feature function is a binary vector where the elements equal 1 if a certain 3-gram occurs in a sequence and 0 otherwise. For sequential representations, this measure is evaluated for the complete byte sequence of the requests. For the token representation, the measure is evaluated separately for all attributes of matching tokens and added up for all tokens.

3-gram frequency features
The computation of the frequency feature functions ϕ_{freq}^{seq} and ϕ_{freq}^{tok} is analogous to the 3-gram occurrence features. The only difference is that both vectors now contain the frequencies of the occurring 3-grams.

Expert features
The feature functions ϕ_{exp}^{seq} and ϕ_{seq}^{tok} exploit the expert knowledge about observed requests. We have chosen a somewhat eccentric set of features to show that even a wild guess may be well-suited for the automatic feature selection approach. Our feature set contains 16 features defined as follows. Positions 1 to 11 represent a coarse string length histogram. The range of observed string lengths up to l_{max}, the largest string length in a training corpus, is divided in 10 equally spaced bins. A binary feature is set, if the observed string length falls into the respective bin. Position 11 is reserved for strings exceeding the maximal training string length; this position is always 0 for the training data but may be set to one for the test data. Position 12 is set to one if the entropy of a string lies in the interval $[4.5, 5.5]$. Positions 13–15 flag the occurrence of the following character types in HTTP requests:

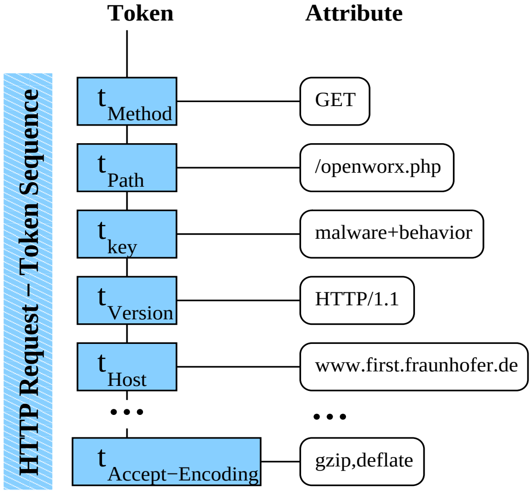

Figure 2: A protocol analyzer returns a set of token-attribute pairs for each HTTP request.

- non-printable characters: ANSI numbers 127-255,
- control characters: ASCII numbers 0-31 except for 10 and 13, and
- uncommon characters: \$, [,], {, }, |, \.

Position 16 is set if blacklisted words that are not supposed to appear in a request – in our case: *exec*, *home*, *passthru*, *root*, *CMD* and *SYSTEM* – are found in a string. The difference between sequential and token representations is the same as for the other feature sets.

4.2 Results

We compare the accuracy of the detector obtained by automatic feature selection using the proposed approach with the accuracy of individual detectors using each of the six features separately and a uniform mixture of the features. The respective optimization problems are solved with CPLEX. For the experiments, we randomly draw distinct training, validation, and test sets from the normal pool. The validation and test sets are each augmented by 15 randomly drawn attacks, where we make sure that attacks of the same class occur only in one of the two sets. For every $\eta \in [0, 250]$, each model in our discourse area is adapted to the training set and subsequently tested on the validation set for model selection. Models realizing the largest area under the ROC curve in the false-positive interval $[0, 0.01]$ ($\text{AUC}_{0.01}$) on the

[1] http://www.metasploit.com/

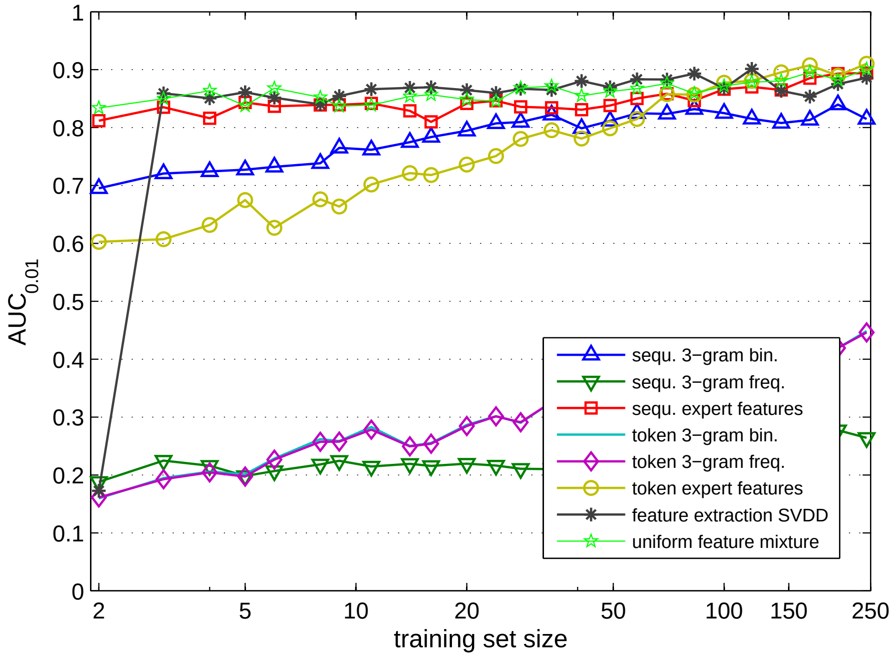

Figure 3: Average $AUC_{0.01}$ performances for varying training set size.

validation set are then evaluated on the independent test set.

We investigate the accuracy of learned detectors as a function of the training set size. The average $AUC_{0.01}$ values are reported in Figure 3 over $n = 100$ repetitions with randomly drawn training, validation, and test sets. The standard error (standard deviation divided by $\sqrt{n} = 10$) was observed to be less than 0.01 in our experiments and is not shown in the plots.

It can be clearly seen from Figure 3 that the accuracy of the detector with automatic feature selection dominates the accuracy of all individual classifiers for all training data set sizes (except the ridiculously small training set of size 2). Towards larger training set sizes, some of the features yield equally accurate detectors; the detector obtained by the proposed method remains among the winners.

The behavior of the automatic feature selection becomes clear from the analysis of the distribution of the mixture coefficients for different features shown in Figure 4. Recall that by definition of our problem these features add up to one. It can be seen that for smaller training set sizes, an optimal feature selection is non-sparse, i.e. more than one feature is needed for the best classification. This explains why a strict improvement of the detection accuracy is attained by our method. For larger training sets, the information contained in the data alone becomes sufficient to determine a "strict winner" among the features: in our case, the feature set ϕ^{seq}_{exp}. Although some other feature sets also exhibit good performance for these training set sizes, the choice is made for the feature set with the best overall performance. As a sanity check, we have repeated the experiment with the best feature set replaced by random features and have observed that the best alternative set of features is chosen by automatic feature selection (results not shown in the plots).

5. CONCLUSION

We have presented a novel generalization of the support vector data description (SVDD) that automatically selects the optimal feature combination. The optimization problem of the feature selection SVDD can be formulated as a semi-infinite linear program and solved with standard techniques. The vanilla SVDD is obtained as a special case for only a single feature function. Empirically, the automatic feature selection proved robust against noise in the training data: Fluctuations caused by small sample sizes are absorbed by appropriately chosen mixtures. The feature selection SVDD has consistently outperformed any baseline using only a single feature set.

The proposed method for feature selection for anomaly detection shows that multiple features sets, possibly resulting from various characterizations of the normal traffic, can be *automatically combined* to obtain optimal detectors. In this way a practitioner faced with the choice of alternative feature sets need not make an a priori choice by hand but can rely on the same learning algorithm used to derive the model of normal data.

The future work will focus on optimizing the run-time of the proposed method (currently our implementation uses standard optimization software not suitable to more than a few hundred examples, however for other types of machine learning these kinds of methods have been shown to scale to thousands of training examples [1, 12, 17]), as well as to extend the proposed method to other anomaly detection algorithms.

Acknowledgements

The authors wishes to thank Konrad Rieck and Alexander Zien for fruitful discussion and helpful comments. This work was supported in part by the German Bundesministerium für Bildung und Forschungt (BMBF) under the project RE-

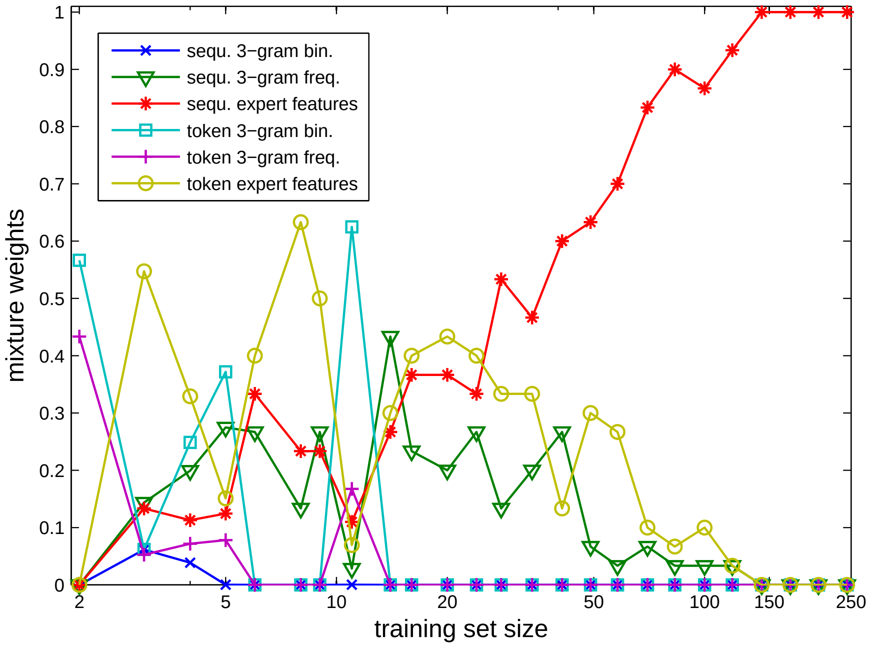

Figure 4: Averaged mixture coefficients $\vec{\beta}$ for varying training set size.

MIND (FKZ 01-IS07007A) and by the FP7-ICT Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-216886.

6. REFERENCES

[1] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[2] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff. A sense of self for unix processes. In *Proc. of IEEE Symposium on Security and Privacy*, pages 120–128, Oakland, CA, USA, 1996.

[3] A. Globerson and N. Tishby. Sufficient dimensionality reduction. *Journal of Machine Learning Research*, 3:1307 – 1331, 2003.

[4] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, 2003.

[5] W. Lee and S. Stolfo. A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information Systems Security*, 3:227–261, 2000.

[6] M. Mahoney and P. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 376–385, 2002.

[7] M. Mahoney and P. Chan. Learning rules for anomaly detection of hostile network traffic. In *Proc. of International Conference on Data Mining (ICDM)*, 2003.

[8] H.-N. Nguyen and S.-Y. Ohn. Drfe: Dynamic recursive feature elimination for gene identification based on random forest. In *Proceedings of the International Conference on Neural Information Processing*, 2006.

[9] R. Pang, V. Paxson, R. Sommer, and L. L. Peterson. binpac: a yacc for writing application protocol parsers. In *Proc. of ACM Internet Measurement Conference*, pages 289–300, 2006.

[10] K. Rieck and P. Laskov. Detecting unknown network attacks using language models. In *Detection of Intrusions and Malware, and Vulnerability Assessment, Proc. of 3rd DIMVA Conference*, LNCS, pages 74–90, July 2006.

[11] K. Rieck and P. Laskov. Language models for detection of unknown attacks in network traffic. *Journal in Computer Virology*, 2(4):243–256, 2007.

[12] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, July 2006.

[13] D. M. Tax and R. P. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004.

[14] K. Wang, J. Parekh, and S. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Adances in Intrusion Detection (RAID)*, pages 226–248, 2006.

[15] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In *Recent Adances in Intrusion Detection (RAID)*, pages 203–222, 2004.

[16] H.-L. Wei and S. A. Billings. Feature subset selection and ranking for data dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):162–166, 2007.

[17] A. Zien and C. S. Ong. Multiclass multiple kernel learning. In Z. Ghahramani, editor, *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 1191–1198. ACM, 2007.