Visualization of anomaly detection using prediction sensitivity

Pavel Laskov[1], Konrad Rieck[1], Christin Schäfer[1] and Klaus-Robert Müller[1,2]

[1]Fraunhofer-FIRST.IDA
Kekuléstr. 7
12489 Berlin, Germany

[2]University of Potsdam
Am Neuen Palais 10
14469 Potsdam, Germany

{laskov,rieck,christin,klaus}@first.fhg.de

Abstract: Visualization of learning-based intrusion detection methods is a challenging problem. In this paper we propose a novel method for visualization of anomaly detection and feature selection, based on prediction sensitivity. The method allows an expert to discover informative features for separation of normal and attack instances. Experiments performed on the KDD Cup dataset show that explanations provided by prediction sensitivity reveal the nature of attacks. Application of prediction sensitivity for feature selection yields a major improvement of detection accuracy.

1 Introduction

Transparency is an essential requirement for intrusion detection algorithms to be used in practice. It does not suffice that an algorithm tells – perhaps with a degree of uncertainty – if some attack (or a specific attack) is present; an algorithm must be able to provide a credible evidence to its prediction.

While such evidence is easy to produce for rule-based detection methods, whose rules are understandable to an expert, such credibility cannot be claimed by many approaches using learning-based methods, such as Neural Networks or Support Vector Machines [GSS99, MJS02]. The situation is somewhat better for misuse detection methods, for which several feature selection techniques are available, e.g. [WMC+01, GE03]. The problem is much graver for anomaly detection methods, for which almost no practical feature selection techniques are known to date.

In this contribution we propose a technique that enables one to visualize predictions of the quarter-sphere SVM, an anomaly detection technique proposed in [LSK04, LSKM04]. The technique is based on the notion of *prediction sensitivity* which measures the degree to which prediction is affected by adding weight to a particular feature. Using this technique we were able to gain interesting information about the predictions made by the quarter-sphere SVM on the KDD Cup dataset. The information we obtained is comparable but not identical to rules inferred by RIPPER, a classical rule-based method [Coh95].

By averaging prediction sensitivity over several datasets one can select the features that

are most important for anomaly detection. In our experiments on the KDD Cup dataset we have observed that reducing the set of features to the ones suggested by prediction sensitivity remarkably improves the accuracy of detection by the quarter-sphere SVM.

2 Approach: analysis of prediction sensitivity

The notion of *prediction sensitivity* expresses the degree to which prediction is affected by adding weight to individual features. Mathematically this can be described by the Jacobian matrix of the prediction function with respect to the input features. The derivation of the expression for this Jacobian matrix – which depends on a particular anomaly detection method, in our case, the quarter-sphere SVM – is rather technical, therefore, due to space limitations, only the main idea is presented in this section. The mathematical details will be subject of a forthcoming publication.

Let X be a $d \times l$ data matrix containing d features collected over l observations. We assume that an anomaly detection algorithm assigns the *anomaly score* $s(x_i)$ to every data point $x_i \in X$ (a column in the data matrix). The $l \times d$ Jacobian matrix is defined as the partial derivatives of s with respect to the components of x_i:

$$J_{(ik)} = \frac{\partial s(x_i)}{\partial x_k}, \quad 1 \leq i \leq l, \quad 1 \leq k \leq d. \tag{1}$$

For the sake of more intuitive visualization we will always consider the transposed Jacobian matrix J^T whose dimensions are identical to those of the initial matrix X. Thus, each column of the (transposed) Jacobian matrix can be seen as the sensitivity of the prediction $s(x_i)$ of the algorithm on the data point x_i with respect to the k-th feature of the data. The definition of $s(x_i)$ for the quarter-sphere SVM used in this paper is given in Eq. (4) in Sec. 3.

Further information can be gained by considering statistical properties of prediction sensitivity. To perform such analysis, randomly drawn data samples $X_1, \ldots, X_N$ are collected, in which the percentage of attacks is fixed. Once the data samples are collected, one computes the mean and the standard deviation of the respective Jacobian matrices over N samples. Based on this information, heuristic criteria can be defined (cf. Sec. 5) for selecting informative features for separating attacks and normal patterns.

3 Application: anomaly detection using quarter-sphere SVM

The quarter-sphere SVM [LSK04, LSKM04] is an anomaly detection technique based on the idea of fitting a sphere onto the center of mass of data. Once the center of the sphere is fixed, the distance of points from the center defines the anomaly score. Choosing a threshold for the attack scores determines the radius of the sphere encompassing the normal data points.

This geometric model can be extended for non-linear surfaces. We first apply some non-linear mapping Φ to the original features. Then, for each data point, the distance from the center of mass in the transformed space – which is our score function – is computed as:

$$s(x_i) = ||\Phi(x_i) - \frac{1}{l}\sum_{j=1}^{l}\Phi(x_j)||. \tag{2}$$

It remains to be shown how the score function (2) can be obtained without explicitly computing the mapping Φ, since the latter can map the data into a high- or even infinite-dimensional space.

It is well known in the machine learning literature (e.g. [MMR+01, SS02]), that, under some technical assumptions, inner products between images of data points under a non-linear transformation can be computed by an appropriate kernel function:

$$k(x_i, x_j) = \Phi(x_i)^T\Phi(x_j).$$

For many interesting transformation the kernel function is known in advance and is easy to compute. For example, for the space of radial-basis functions (RBF) the kernel function is computed as

$$k(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\gamma}}.$$

To compute the score function $s(x_i)$ using the kernel function, the following steps are needed:

1. Form the $l \times l$ kernel matrix K whose entries are the values of the kernel function $k(x_i, x_j)$ for all pairs of data points i and j.
2. Compute the centered kernel matrix [SSM98, SMB+99]:

$$\tilde{K} = K - 1_l K - K 1_l + 1_l K 1_l, \tag{3}$$

   where 1_l is an $l \times l$ matrix with all values equal to $\frac{1}{l}$.

3. The score function is given by the entries on the main diagonal of the centered kernel matrix:

$$s(x_i) = \sqrt{\tilde{K}_{(ii)}}. \tag{4}$$

4 Experimental setup

Before presenting the operation of our visualization technique a few remarks need to be made on data preprocessing. In our experiments we use the KDD Cup 1999 dataset [SWL+99], a standard dataset for the evaluation of data mining techniques. The set comprises a fixed set of connection-based features computed from the DARPA 1998 IDS evaluation [LCF+99] and contains 4898430 records of which 3925650 are attacks. A list of all features is provided in [LS01, LSKM04]. In-depth description of some features, e.g. the `hot` feature, is available in the Bro IDS documentation [Pax98, Pax04].

The distribution of attacks in the KDD Cup dataset is extremely unbalanced. Some attacks are represented with only a few examples, e.g. the `phf` and `ftp_write` attacks, whereas the `smurf` and `neptune` attacks cover millions of records. In general, the distribution of attacks is dominated by probes and denial-of-service attacks; the most interesting – and dangerous – attacks, such as compromises, are grossly under-represented.

In order to cope with the unbalanced attack distribution and to investigate the characteristic features of particular attacks, we construct separate datasets containing a fixed attack ratio of 5%. The desired ratio is achieved by combining two randomly drawn sub-samples. The first sub-sample is drawn from the attacks in question. If an attack is under-represented, i.e. there are too few samples to carry random sampling, all attack examples are drawn. The second sub-sample is drawn randomly from normal data matching the services used in the chosen attack. The number of examples in both sub-samples is chosen so as to attain the desired attack ratio.

In order to analyze the statistical properties of prediction sensitivity, as indicated in Sec. 2, 10 datasets of 1000 data points are generated for each attack. If the number of available attacks in the data is smaller than 50 (required to have 5% of attacks in datasets of size 1000), we reduce the dataset size to $L < 1000$, sufficient to accommodate all available attacks, and increase the number of generated datasets by the factor of $1000/L$.

After the sub-sampled datasets are generated, a data-dependent normalization [EAP+02] is computed, a quarter-sphere SVM is applied to each dataset and the corresponding Jacobian matrices (cf. Eq. (1)) are calculated.

5 Interpretation of anomaly detection on the KDD Cup dataset

The proposed prediction sensitivity criterion can be visualized by plotting the Jacobian matrix. If multiple training sets are available the mean and the standard deviation Jacobian matrices are plotted. The rows of the matrices correspond to features and the columns correspond to normal and attack instances.

An example of such visualization for the `land` attack is shown in Fig. 1. The following observations can be inferred from the prediction sensitivity matrices:

- Random sampling and averaging of prediction sensitivities emphasize the salient features of the data. As a result, instances corresponding to a particular attack are characterized by consistent regions in the mean Jacobian matrix, whereas the much more heterogeneous normal data exhibits random sensitivity.
- The consistency of prediction sensitivity for attack instances can be quantified by the standard deviation Jacobian matrix. Salient features exhibit low standard deviation. Thus one can suggest the following heuristic criterion for feature selection: *for attack instances, features must have high values in the mean and low values in the standard deviation Jacobian matrix*.

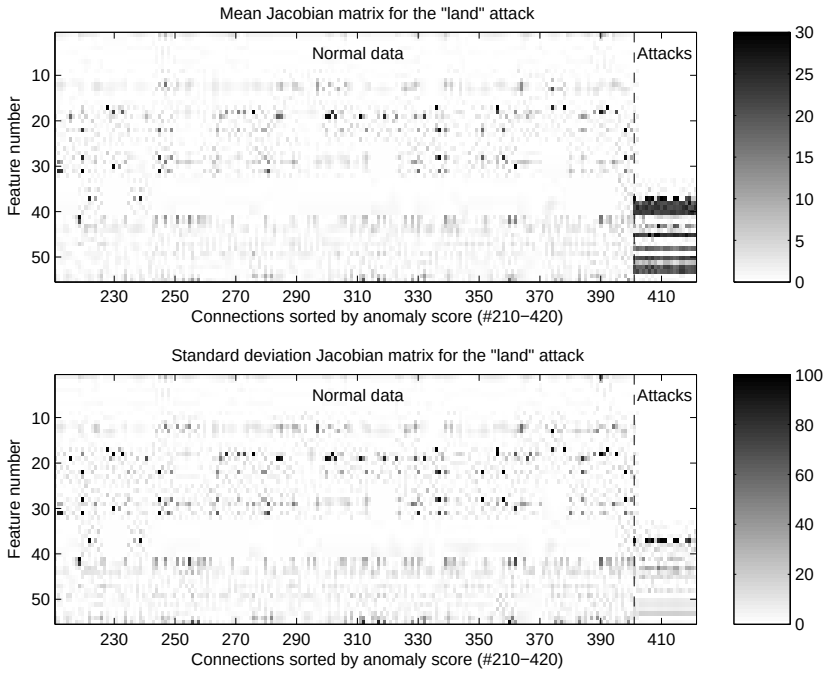

Figure 1: Visualization of prediction sensitivity. The mean and the standard deviation Jacobian matrices for the `land` attack exhibit different patterns for attack and normal data, as well as different impact of particular features on prediction. The grey-scale bars to the right of the figure illustrate the range of matrix values.

In order to illustrate feature selection based on the proposed criterion, we calculate the mean and the standard deviation of the mean Jacobian matrix for the attack instances only. These quantities computed for the `land` attack are shown in Fig. 2. One can see that the numerical characteristics of prediction sensitivity provide substantial information for identifying candidate features. According to the "high mean/low variance" criterion, most prominent for this example are the features 38, 39, 40, 45. Their names and brief descriptions are shown in Table 1. These features are indeed meaningful for the `land` attack. This attack is manifested in transmission of single TCP packets (with SYN set) that crash a server without eliciting an ACK reply; as a result high SYN error rates are observed. The features 48, 50, 52, 53 may also be added as second-choice candidates.

Number	Name	Description
38	`srv_count`	Number of connections to service
39	`serror_rate`	SYN error rate
40	`srv_serror_rate`	SYN error rate for service
45	`srv_diff_host_rate`	SYN error rate for service on multiple hosts

Table 1: Feature subset selected using prediction sensitivity.

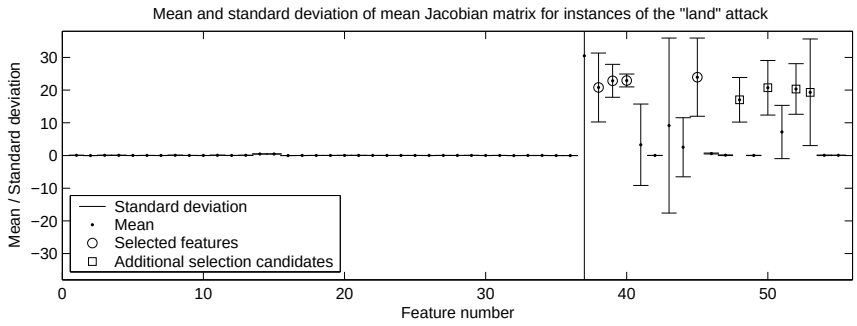

Figure 2: Mean and standard deviation of mean Jacobian matrix for instances of the `land` attack. According to the "high mean/low variance" criterion a subset of features and additional candidate features have been selected.

We have performed the interpretation and analysis of all 21 attacks present in the KDD Cup dataset. Due to space constraints we cannot present the detailed analysis here; so we restrict ourselves to 5 characteristic attacks which demonstrate the strengths as well as the limitations of the proposed visualization technique.

For each of the attack classes *remote-to-local* (R2L), *user-to-root* (U2R) and *probe* one attack was arbitrarily selected. For the class of *denial-of-service* (DoS) attacks we decided to interpret two attacks which differ in activity. The following attacks were chosen:

- the `phf` (R2L) attack exploits a security flaw in the input handling of CGI scripts which allows the execution of local commands on a remote web server,
- the `loadmodule` (U2R) attack exploits an improper boundary check in the program `loadmodule` of the Solaris operating system and allows a local attacker to gain super-user privileges,
- the `portsweep` (probe) attack discovers active services on remote hosts by systematically requesting connections to multiple TCP ports,
- the `pod` (DoS) attack crashes or reboots remote systems by sending a single, oversized IP datagram corrupting the host's packet reassembly,
- the `smurf` (DoS) attack uses misconfigured broadcast hosts to flood a victim host with spoofed ICMP datagrams.

In order to qualitatively compare the proposed feature selection method with alternative techniques, we applied the RIPPER classifier to our datasets, in a similar way as it was previously used in [LSM99, LS01] for feature analysis and generation of detection rules.

Table 2 lists the selected features based on prediction sensitivity and corresponding RIPPER rule sets for the five example attacks.

`phf`	Feature selection based on prediction sensitivity:
	`hot, num_access_files, duration`
	RIPPER rule set:
	`phf :- root_shell>=1, src_bytes<=51.`

`loadmodule`	Feature selection based on prediction sensitivity:
	`dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate`
	RIPPER rule set:
	`loadmodule :- dst_host_count<=6, src_bytes<=0, count>=2.`
	`loadmodule :- dst_host_count<=6, num_file_creations>=1, duration<=103.`

`portsweep`	Feature selection based on prediction sensitivity:
	`rerror_rate, srv_rerror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate`
	RIPPER rule set:
	`portsweep :- dst_host_srv_rerror_rate>=1, dst_host_same_srv_rate<=0.01, dst_host_same_src_port_rate>=0.02.`
	`portsweep :- src_bytes<=1, dst_host_same_srv_rate<=0.02, dst_host_same_src_port_rate>=0.03.`
	`portsweep :- rerror_rate>=0.19, dst_host_same_srv_rate<=0.8, dst_host_same_src_port_rate>=0.08, dst_host_count>=78, protocol_type=tcp.`
	`portsweep :- src_bytes<=0, service=private.`
	`portsweep :- src_bytes<=8, protocol_type=icmp.`
	`portsweep :- src_bytes<=0, service=ftp_data, dst_bytes<=0.`
	`portsweep :- duration>=42908.`
	`portsweep :- dst_host_rerror_rate>=0.95, dst_host_diff_srv_rate>=0.47.`
	`portsweep :- flag=OTH, service=smtp.`

`pod`	Feature selection based on prediction sensitivity:
	`src_bytes, wrong_fragment`
	RIPPER rule set:
	`pod :- src_bytes>=564.`

`smurf`	Feature selection based on prediction sensitivity:
	`count, src_count, src_bytes`
	RIPPER rule set:
	`normal :- src_bytes<=64.`

Table 2: Feature selection based on prediction sensitivity and RIPPER rule sets for selected attacks

Two questions arise from Table 2: How are the selected features related to the nature of attacks and why do features extracted by RIPPER and prediction sensitivity differ?

- For the `phf` attack the selected features indicate malicious activity accessing system files, e.g. `/etc/passwd`, and an anomal connection duration. These features match the typical application pattern of the `phf` attack, in which system files are retrieved by a short HTTP GET request. The corresponding RIPPER rule set reveals the problem of *overfitting*. The rules match specific properties of the training sets, but do not identify the general properties of the attack in question.
- The `loadmodule` attack belongs to the class of U2R attacks and thus evidence of the attack is only present in *content-based* features. The selected features and the RIPPER rules mainly contain *traffic-based* features. Both methods fail to select the relevant features because no content-based features clearly reflect the presence of the `loadmodule` attack.
- For the `portsweep` probe the prediction sensitivity reveals features related to rejection errors, e.g. `rerror_rate`. A side effect of vanilla portscans, as in case of `portsweep`, is a very high number of rejected connection requests because only few services are present on most network hosts. The RIPPER rule set is too complex for realistic application. Furthermore most rules involve the service and protocol feature which are not inherent properties of the `portsweep` attack.
- The selected features for the `pod` attack indicate an influence of the number of transmitted bytes and the presence of wrong fragments, which are very characteristic for the ping-of-death (`pod`) attack. The RIPPER rule is, however, too specific: there is no reason to believe that 564 bytes is a good threshold between normal data and the `pod` attack.
- The `smurf` attack is represented by traffic-based features, such as `count` and `srv_count`. The attack involves tremendous traffic from various spoofed sources. The selected feature set matches the `smurf` attack, but also contains generalization which applies to successor attacks, e.g. `fraggle`. The RIPPER rule exhibits similar overfitting as for the `pod` attack.

One can see that, provided relevant features are present in the data, both RIPPER and prediction sensitivity succeed in selecting an informative subset of features. However the RIPPER classifier is prone to overfitting, and the inferred rules often lack necessary generality, which undermines the main advantage of rule-based learning: understandable rules. Feature selection based on prediction sensitivity is more accurate and exhibits good generalization ability. Another difference between the two techniques is that prediction sensitivity determines a threshold for a combination of rather than for single features.

6 Improvement of detection by feature selection

As it was shown in the previous section, the prediction sensitivity criterion allows one to select an informative subset of features characterizing single attacks. Although we used labels for feature selection, the underlying concept beyond the notion of prediction sensitivity is anomaly detection in unlabeled data. In this section we demonstrate that the feature selection based on our criterion improves the accuracy of the quarter-sphere SVM, an unsupervised anomaly detection algorithm.

The experiments presented below were carried out under two scenarios. First we selected features for *single attacks* and applied a quarter-sphere SVM on the reduced feature sets. In the second experiment, the datasets – for feature selection as well as for anomaly detection – were composed of multiple attacks (ab)using the same service: FTP, HTTP and SMTP. The objective of both experiments is to investigate whether pre-selection of features improves detection accuracy compared to the full set of features. In all experiments, *unseen data* was used for the evaluation of feature selection in order to ensure that the selection generalizes beyond the particular datasets.

The impact of feature selection on the accuracy of anomaly detection by the quarter-sphere SVM is shown in Fig. 3. The evaluation criterion is the area under the ROC curve restricted to the low false-positive interval $[0, 0.1]$ ($\text{AUC}^{0.1}$). The area is multiplied by a factor of 10; this allows one to interpret $\text{AUC}^{0.1}$ as a percentage of the maximum attainable area on the desired interval of interest.

It can be seen from Fig. 3 that reducing the features according to prediction sensitivity provides a major improvement of the $\text{AUC}^{0.1}$ values. For no attack does the $\text{AUC}^{0.1}$ decrease after the feature selection. These results are very promising since detection accuracy at low false-positive rates is extremely important in IDS.

The full ROC curves for four attacks analyzed in Sec. 5 are shown in Fig. 4. The ROC curve for the `pod` attack was almost perfect before feature selection and thus is not shown in Fig. 4.

7 Discussion and conclusions

We have presented a new technique for visualization of anomaly detection based on prediction sensitivity. Its application enables an expert (a) to interpret the predictions made by anomaly detection and (b) to select informative features in order to improve detection accuracy.

Our experiments were conducted using the quarter-sphere SVM and the KDD Cup dataset. The features highlighted by prediction sensitivity reasonably reveal the nature of attacks present in this dataset, and, furthermore, exhibit more generality than the rules suggested by RIPPER, a rule-based learning method.

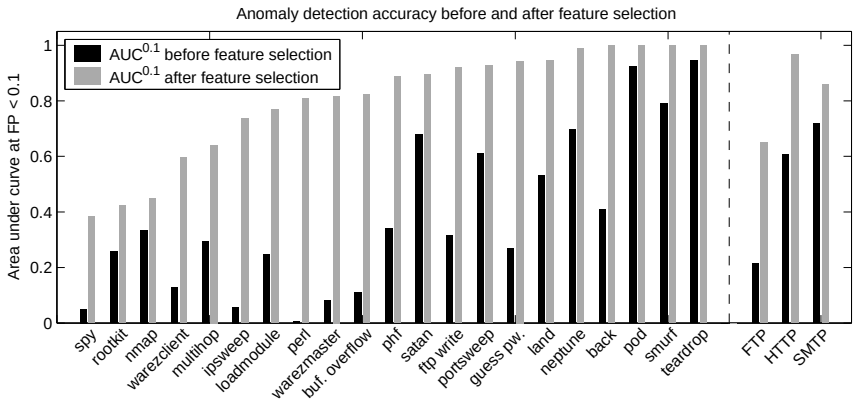

Figure 3: Anomaly detection accuracy before and after feature selection. The left part shows experiments with single attack datasets. The right part corresponds to experiments with FTP, HTTP and SMTP datasets.

The feature selection experiments showed major improvements of the accuracy for anomaly detection on a specific subset of features chosen by prediction sensitivity, which confirms the explanatory power of prediction sensitivity.

How can the proposed technique be useful in practice? It is true that the experimental setup presented in Sec. 4 is not fully unsupervised. One cannot, as one would like to, simply feed the data into the algorithm and obtain the explanations to predictions and the set of informative features. On the other hand, the label information is anyway needed for testing of intrusion detection systems: nobody would venture to deploy an IDS without ever wondering if it works right. At this point, using our technique, one can utilize the available label information to look beyond the bare accuracy metrics and obtain insights into *why* the anomaly detection produces the results it is producing and *what* can be done to improve it. Although labels are used for feature selection, no explicit training is required, and in this sense the procedure remains unsupervised. Furthermore, the explanatory information provided by prediction sensitivity can be particularly useful as a first guidance for development of signatures for unknown attacks.

8 Acknowledgements

The authors gratefully acknowledge the funding from *Bundesministerium für Bildung und Forschung* under the project MIND (FKZ 01-SC40A), and from *Deutsche Forschungsgemeinschaft* under the project MU 987/2-1. We would like to thank Sebastian Mika and Stefan Harmeling for fruitful discussions and anonymous reviewers whose suggestions helped to improve the quality of presentation.

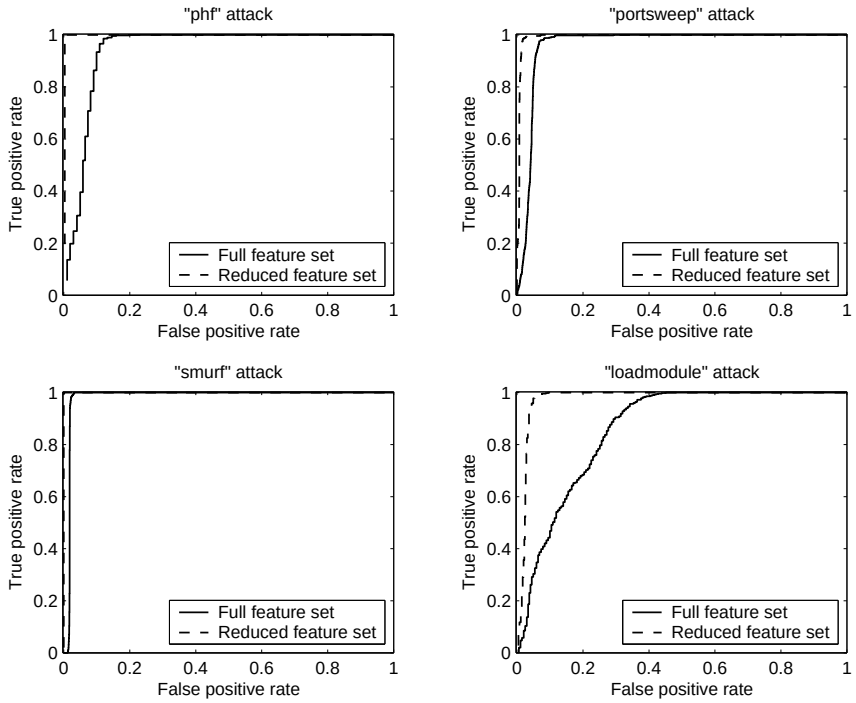

Figure 4: Full ROC curves for the attacks `phf`, `portsweep`, `smurf` and `loadmodule` after feature selection.

References

[Coh95] W. Cohen. Fast Effective Rule Induction. In *Proc. of the 12th International Conference on Machine Learning*, 1995. http://wcohen.com/postscript/ml-95-ripper.ps.

[EAP+02] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. *Applications of Data Mining in Computer Security*, chapter A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. Kluwer, 2002.

[GE03] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *JMLR*, 3:1157–1182, 2003.

[GSS99] A. K. Ghosh, A. Schwartzbard, and M. Schatz. Learning Program Behavior Profiles for Intrusion Detection. In *Proc. of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring*, pages 51–62, Santa Clara, USA, April 1999. http://www.cigital.com/papers/download/usenix_id99.pdf.

[LCF+99] R. Lippmann, R. K. Cunningham, D. J. Fried, K. R. Kendall, S. E. Webster, and M. A. Zissman. Results of the DARPA 1998 Offline Intrusion Detection Evaluation. In *Proc. RAID 1999*, 1999.

[LS01] W. Lee and S. Stolfo. A Framework for Constructing Features and Models for Intrusion Detection Systems. In *ACM Transactions on Information and System Security*, volume 3, pages 227–261, November 2001.

[LSK04] P. Laskov, C. Schäfer, and I. Kotenko. Intrusion detection in unlabeled data with quarter-sphere Support Vector Machines. In *Proc. DIMVA*, pages 71–82, 2004.

[LSKM04] P. Laskov, C. Schäfer, I. Kotenko, and K.-R. Müller. Intrusion detection in unlabeled data with quarter-sphere Support Vector Machines (Extended Version). *Praxis der Informationsverarbeitung und Kommunikation*, 27:228–236, 2004.

[LSM99] W. Lee, S. Stolfo, and K. Mok. A data mining framework for building intrusion detection models. In *Proc. IEEE Symposium on Security and Privacy*, pages 120–132, 1999.

[MJS02] S. Mukkamala, G. Janoski, and A. Sung. Intrusion Detection using Neural Networks and Support Vector Machines. In *Proceedings of IEEE Internation Joint Conference on Neural Networks*, pages 1702–1707, May 2002.

[MMR+01] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An Introduction to Kernel-Based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[Pax98] V. Paxson. Bro: a system for detecting network intruders in real-time. In *Proc. USENIX Security Symposium*, pages 31–51, 1998.

[Pax04] V. Paxson. The Bro 0.8 User Manual. Lawrence Berkeley National Laboratroy and ICSI Center for Internet Research, 2004.

[SMB+99] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola. Input Space vs. Feature Space in Kernel-Based Methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, September 1999.

[SS02] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[SSM98] B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel Eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[SWL+99] S. J. Stolfo, F. Wei, W. Lee, A. Prodromidis, and P. K. Chan. KDD Cup - Knowledge Discovery and Data Mining Competition, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[WMC+01] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature Selection for SVMs. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press, 2001.