K.G. Saur Verlag, München, 2004

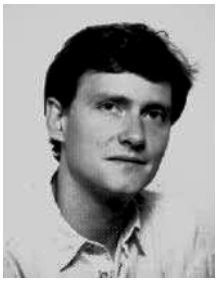
P. Laskov, C. Schäfer, I. Kotenko, and K.-R. Müller

Intrusion Detection in Unlabeled Data with Quarter-sphere Support Vector Machines

Pavel Laskov received the diploma in computer systems engineering from the Moscow Institute for Radioengineering, Electronics and Automatization in 1994, M.Sc. and Ph.D. in computer science from the University of Delaware in 1996 and 2001 respectively. Since 2001 he has been working as a research scientist at the intelligent data analysis (IDA) group of the Fraunhofer FIRST. His main research interests include design and analysis of machine learning algorithms and applications of machine learning in computer security.

Christin Schäfer received the diploma in statistics in 2001 from the University of Dortmund, Germany. She conducted studies in statistical learning theory with focus on the application of the Support Vector Machine (SVM) to dependent biomedical time series. Since 2002 she is with the intelligent data analysis (IDA) group at the Fraunhofer FIRST. Her scientific interests are in the fields of machine learning and statistics.

Igor Kotenko graduated with honors St. Petersburg Military Academy of Space Engineering (Department of Telecommunication and Computer-Aided Systems) and St. Petersburg Military Signal Academy (Department of Computer-Aided systems). He obtained the Ph.D. degree in 1990 and the National degree of Doctor of Engineering Science in 1999. He is Professor of computer science and Leading researcher of St. Petersburg Institute for Informatics and Automation. Scientific interests comprise artificial intelligence, including multi-agent systems, network security and information warfare, computer-aided decision support and telecommunication systems.



Klaus-Robert Müller received the diploma in mathematical physics 1989 and the Ph.D. in theoretical computer science in 1992, both from University of Karlsruhe, Germany. From 1992 to 1994 he worked as a Postdoctoral fellow at GMD FIRST in Berlin where he started to build up the intelligent data analysis (IDA) group. From 1994 to 1995 he was a European Community STP Research Fellow at University of Tokyo in Prof. Amari's Lab. From 1995 on he is the department head of the IDA group at GMD FIRST (since 2001 Fraunhofer FIRST) in Berlin and since 1999 he holds a joint Associate Professor position of the GMD and the University of Potsdam. In 2003 he became the full professor at the University of Potsdam. He has been lecturing at the Humboldt University, the Technical University of Berlin and the University of Potsdam. In 1999 he received the annual national prize for pattern recognition (Olympus Prize) awarded by the German pattern recognition society DAGM. He serves in the editorial board of the journals "Computational Statistics", "IEEE Transactions on Biomedical Engineering" and in the program and organization committees of various international conferences. His research areas include statistical physics and statistical learning theory for neural networks, support vector machines and ensemble learning techniques. He contributed to the field of signal processing working on time-series analysis, statistical denoising methods and blind source separation. His present application interests are expanded to the analysis of biomedical data, most recently to brain computer interfacing and genomic data analysis.

ABSTRACT

The anomaly detection methods are receiving growing attention in the intrusion detection community. The two main reasons for this are their ability to handle large volumes of unlabeled data and to detect previously unknown attacks. In this contribution we investigate the application of a modern machine learning technique – one-class Support Vector Machines (SVM) – for anomaly detection in unlabeled data. We propose a novel formulation of this technique which is particularly suited for the data typical for intrusion detection systems. Our evaluation on the well-known KDDCup dataset demonstrates a significant improvement over previous formulations of the one-class SVM.

1 INTRODUCTION

The majority of current intrusion detection methods can be classified as either misuse detection or anomaly detection [NWY02]. The former identify patterns of known illegitimate activity; the latter focus on unusual activity. Both groups of methods have their advantages and disadvantages. Misuse detection methods are generally more accurate but are fundamentally limited to known attacks. Anomaly detection methods are usually less accurate than misuse detection methods – in particular, their false alarm rates are hardly acceptable in practice – however, they are at least in principle capable of detecting novel attacks. This feature makes anomaly detection methods a topic of active research.

In some early approaches, e.g. [DR90, LV92], it was attempted to describe normal behavior by means of some high-level rules. This turned out to be quite a difficult task. More successful was the idea of collecting data from normal operation of a system and computing, based on this data, features describing normality. Deviation of such features was considered an anomaly. This approach is known as "supervised anomaly detection". Different techniques have been proposed for characterizing the concept of normality, most notably statistical techniques, e.g. [De87, JLA+93, PN97, WFP99] and data mining techniques, e.g. [BCJ+01, VS00]. In practice, however, it is difficult to obtain clean data to implement these approaches. Verifying that no attacks are present in the training data may be an extremely tedious task, and this is infeasible for the large amount of data. On the other hand, if the "contaminated" data is treated as clean, intrusions similar to the ones present in the training data will be accepted as normal patterns.

To overcome the difficulty in obtaining clean data, the idea of unsupervised anomaly detection has been recently proposed and investigated on several intrusion detection problems [PES01, EAP+02, LEK+03]. These methods compute some relevant features and use techniques of unsupervised learning to identify sparsely populated areas in feature space. The points – whether in the training or in the test data – that fall into such areas are treated as anomalies.

Two kinds of unsupervised learning methods have been investigated: clustering methods and one-class Support Vector Machines (SVM). In this contribution we focus on the one-class SVM methods and investigate application of their underlying geometric ideas in the context of intrusion detection.

Before we explain in detail the workings of SVM in the next section, we would like to mention two simple geometric ideas used in previous one-class SVM approaches:

- a hyperplane separating the normal data from the origin [SPST+01], and
- a sphere encircling the normal data [TD99].

Our analysis of the typical data arising in intrusion detection systems brings us to the conclusion that neither of these two formulations is well-suited for such specific data. Based on this analysis we propose the novel formulation whose main idea is to fit a sphere *centered at the origin* to the data. This formulation, to be referred to as a quarter-sphere, is particularly suitable for the data that has one-sided distribution and is concentrated around the origin.

The rest of this article is organized as follows. In section 2 we present the basic notions of the learning theory which are necessary to understand the main ideas of Support Vector Machines. In section 3 we introduce one-class SVM and the two previous formulations. Section 4 presents the quarter-sphere SVM and some algorithmic issues needed to be resolved for its implementation. The experiments on the KDDCup dataset and their interpretation are presented in section 5. Conclusions and a brief description of future work are given in section 6. The technical details and some reference material on the KDDCup dataset are left for the appendices A-C.

2 LEARNING TO CLASSIFY – SOME THEORETICAL BACKGROUND

Assume for the moment we had some data – for example, records of network connections – for which we new exactly whether a given data point belongs to an attack or not. Based on this information we would like to develop a rule for the detection of connections containing attacks. Our special focus would lie on *the generalization ability* of the rule, i.e. that it works not only for the data we had used to build it but also on any unseen connections. The process of obtaining a rule for some concept from the exemplary data is called "learning from examples", and if labels are available for the examples, the learning process is called "supervised".

How can a concept be learned from the data? Should we strive for a simple or a complex representation of a concept? Does it matter if we have collected only a few data points or an extensive database? These issues are addressed by the statistical learning theory which develops the tools that help design efficient learning algorithms with good generalization. In the following we will highlight some elements of this theory illustrating its most important points.

Consider the example in Fig. 1 (left). Assume the filled points represent the attack records, and the hollow ones represent the normal data. We would like to build a decision function separating the two classes. Two possible choices are shown on the left picture: a very simple one (a line) and a more complicated one (a sinusoid). Which of these two functions (if our choice were limited to only these two) should one prefer? One might think that the sinusoid is a better choice because it correctly separates the training points whereas the linear function makes two errors. Notice, however, that a training dataset is usually limited and furthermore not error-free. Thus, the same training dataset could have been obtained from the concept with a sinusoid as a true function (middle figure, error free data) or from the concept with a line as a true function (right figure, error-prone data). In both cases we can see that if a decision is made for the "wrong" function at the training stage this results in significant error when we apply the learned function for classification.

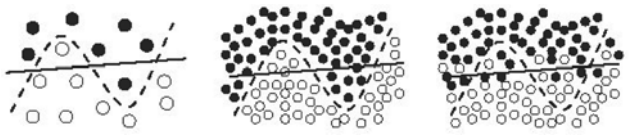


Fig. 1 Illustration of the overfitting dilemma: Given only a small sample (left) either, the solid or the dashed hypothesis might be true, the dashed one being more complex, but also having a smaller training error. Only with a large sample we are able to see which decision respects the true distribution more closely. If the dashed hypothesis is correct the solid would underfit (middle); if the solid were correct the dashed hypothesis would overfit (right). From [MMR+01].

What can be learned from the example above? The training error alone is not an adequate criterion for learning from examples (as it was done, for example in some classical learning with neural networks a decade ago). The complexity of the learned function needs to be under control. One can easily imagine a highly complicated function that perfectly separates the training data but would have nothing to do with the true concept. Thus, in order to achieve good generalization, a balance needs to be found between the training error and the complexity of a learned function. The process for attaining such balance is called "regularization".

SVM is one of the modern learning algorithms that has a built-in regularization mechanism. The main idea of this algorithm is to separate the data points with a hyperplane *with the largest possible margin*. A hyperplane is a generalization of the "line" in the previous example for high-dimensional spaces. The margin (see Fig. 2) is the distance between the two hyperplanes parallel to the separating hyperplane and adjacent to some training data points, the so-called "support vectors". The hyperplane is parameterized by the vector w which moves it around in the space resulting in different margins, the smaller w, the larger the margin. It is possible that for some values of w no margin is available at all, in which case some training points are erroneously classified and the margin is measured between the remaining, correctly classified points.

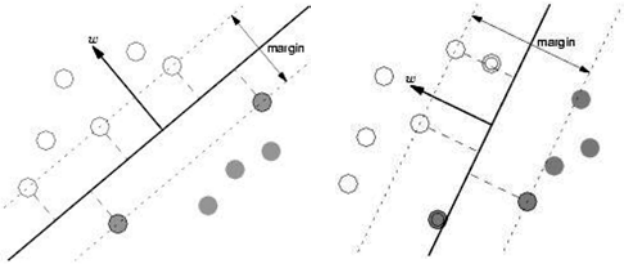


Fig. 2 Controlling the separation margin. On the left picture all training points are fully separated. On the right picture one training point is erroneously separated, but a larger margin (corresponding to the smaller value of the weight w) is attained. The points inside the margin are marked with double circles.

How does one train an SVM in practice? Let $X = \{x_1,\ldots,x_k\}$ be the training data vectors and $Y = \{y_1,\ldots,y_k\}$ be the labels, +1 for attacks and –1 for the normal points. Then training of an SVM amounts to solving the following optimization problem:

$$\min_{w,\xi,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i,$$

$$\text{subject to:} \quad y_i((w\cdot x_i)+b) \geq 1-\xi_i \ , \qquad (1)$$

$$\xi_i \geq 0 \ .$$

The problem (1) can be solved by any standard optimization software (e.g. CPLEX, [CP]) or by the special-purpose SVM software packages (e.g. SVM^{Light} [SL], LibSVM [LS]). The regularization parameter C that controls the tradeoff between the error and the margin is chosen by the following procedure called "cross-validation". Some subset is removed from the training data, and the SVM is trained on the remaining data and is validated on the held out subset of data. The procedure is repeated for several permutations of the data and for different values of the parameter C. The value for which the best validation error has been observed is taken for the optimal regularization parameter.

Although presented so far for the linear separating surfaces, the SVM can be applied also for nonlinear surfaces. This is achieved by means of the so-called "kernel trick". It is known from functional analysis that if we map our original data points x into another space by some mapping $\Phi(x)$, then the inner product $(\Phi(x)\cdot\Phi(x))$ can for many useful mappings be explicitly represented by a function $k((x_i\cdot x_j))$ of the inner product $(x_i\cdot x_j)$. This seemingly abstract mathematical property turns out to be extremely useful in practice. If a linear algorithm uses the data occur only through inner products $(x_i\cdot x_j)$, it can be extended to a non-linear one by merely replacing the inner products with the non-linear functions $k((x_i\cdot x_j))$ thereof. The function k is called the "kernel function". Some well-known examples of the kernel functions are $k(x_i,x_j) = ((x_i\cdot x_j)) + 1)^d$ for the polynomials of degree d, and $k(x_i,x_j) = \exp(-\|x_i-x_j\|^2/(2\sigma))$ for the radial basis functions (RBF). Both kernels have additional parameters: the degree d of the polynomial and the smoothness parameter σ of the RBF. These parameters introduce additional regularization in the algorithms: by choosing them we can select the simpler or the more complicated decision functions. In the experiments presented below we shall make use of the linear kernels (that is, direct inner products) as well as of the RBF kernels.

Further information about SVM, kernels and statistical learning theory can be found e.g. in [Vap95], [MMR+01], [SS02], [MSL+04].

3 ONE-CLASS SVM: PREVIOUS FORMULATIONS

Unlike the supervised learning considered in the previous section, in the unsupervised learning one cannot a-priori distinguish between the two classes of points. Therefore the geometric formulations have to be adjusted; however, the underlying statistical ideas and their implementation in the practical context remain largely intact.

The plane formulation. The original idea of the one-class SVM [SPST+ 01] was formulated as an "estimation of the support of a high-dimensional distribution". The essence of this approach is to map the data points x_i into the feature space by some nonlinear mapping $\Phi(x_i)$ – similarly to the classical SVM, and to separate the resulting image points *from the origin* with the largest possible margin by means of a hyperplane. The geometry of this idea is illustrated in Fig.3 (left).

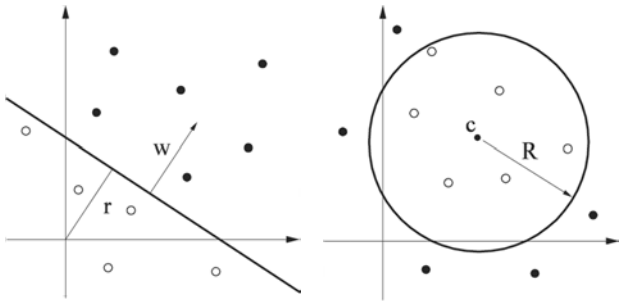


Fig. 3 (left) The geometry of the plane formulation and (right) of the sphere formulation of one-class SVM.

Due to nonlinearity of the feature space, maximization of the separation margin limits the volume occupied by the normal points to a relatively compact area in feature space (see appendix A for the mathematical details of the plane formulation).

The sphere formulation. Another, somewhat more intuitive geometric idea for the one-class SVM is realized in the sphere formulation [TD99], where the normal data is described by a sphere (in a feature space) encompassing the data, as shown in Fig. 3 (right) (see appendix B for mathematical details of the sphere formulation).

Analysis. When applying one-class SVM techniques to intrusion detection problems, the following observation turns out to be of crucial importance: *A typical distribution of the features used in IDS is non-negative and one-sided*. Several reasons contribute to this property. First, many IDS features are of temporal nature, and their distribution can be modeled using distri-

butions common in survival data analysis, for example by an exponential or a Weibull distribution. Second, a popular approach to attain coherent normalization of numerical attributes is the so-called "data-dependent normalization" [EAP+02]. Under this approach, described in more detail in section 5, the features are defined as the deviations from the mean, measured in the fraction of the standard deviation. This quantity can be seen as F-distributed. As a result, the overwhelming mass of data lies in the vicinity of the origin.

The consequences of the one-sidedness of the data distribution for the one-class SVM can be seen in Fig. 4. The one-sided distribution in the example is generated by taking the absolute values of the normally distributed points. The anomaly detection is shown for varying smoothness σ of the RBF (larger values of σ result in stronger regularization). The contours show the separation between the normal points and anomalies. One can see that even for the heavily regularized separation boundaries, as in the right picture, some points close to the origin are detected as anomalies. As the regularization is diminished, the one-class SVM produces a very ragged boundary and does not properly detect anomalies.

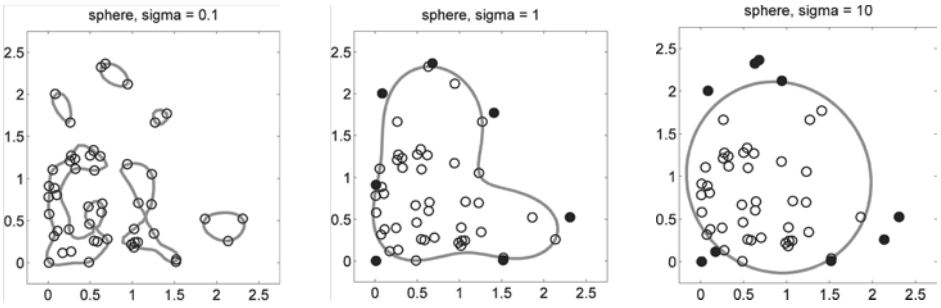


Fig. 4 Separation surfaces of the one-class SVM (sphere) for different values of the parameter σ.

The message that can be carried from this example is that, in order to account for the one-sidedness of the data distribution, one needs to use a geometric construction that is in some sense asymmetric. The new construction we propose here is the quarter-sphere one-class SVM described in the next section.

4 THE QUARTER-SPHERE FORMULATION OF ONE-CLASS SVM

A natural way to extend the ideas of one-class SVM to one-sided non-negative data is to require the center of the fitted sphere be fixed at the origin. The geometry of this approach is shown in Fig. 4. Repeating the derivation of the sphere formulation given in the Appendix B for $c = 0$, the following dual mathematical problem is obtained:

$$\min_{\alpha \in R^l} -\sum_{i=1}^{l} \alpha_i k(x_i, x_i),$$

$$\text{subject to: } \sum_{i=1}^{l} \alpha_i = 1, \qquad (2)$$

$$0 \le \alpha_i \le \frac{1}{\nu l}.$$

Note that, unlike the other two formulations, the dual problem of the quarter-sphere SVM amounts to a linear rather than a quadratic program. Herein lies the key to the significantly lower computational cost of our formulation.

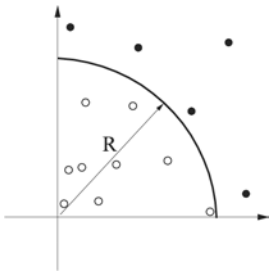


Fig. 5 The geometry of the quarter-sphere formulation of one-class SVM.

It may seem somewhat strange that the non-linear mapping affects the solution only through the norms $k(x_i, x_j)$ of the examples, i.e. that the geometric relations *between* the objects are ignored. This feature indeed poses a problem for the application of the quarter-sphere SVM with the distance-based kernels. In such case, the norms of the points are equal, and no meaningful solution to the dual problem can be found. This predicament, however, can be easily fixed by centering the images of the training points $\Phi(x_i)$ in feature space. In other words, the values of image points are re-computed in the local coordinate system anchored at the center of mass of the image points. This can be done by subtracting the mean from all image values:

$$\tilde{\Phi}(x_i) = \Phi(x_i) - \tfrac{1}{l}\sum_{i=1}^{l} \Phi(x_i).$$

Although this operation may not be directly computable in feature space, the impact of centering on the kernel values can be easily computed (e.g. [SSM98, SMB+99]):

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

where K is the $l \times l$ kernel matrix with the values $K_{ij} = k(x_i, x_j)$, and 1_l is an $l \times l$ matrix with all values equal to $1/l$. After centering in feature space, the norms of points in the local coordinate system are no longer all equal, and the dual problem of the quarter-sphere formulation can be easily solved.

Suitability of the quarter-sphere one-class SVM to non-negative, non-gaussian data can be clearly seen in Fig. 6. This figure shows the application of the quarter-sphere SVM to the same data used in the examples of Fig. 4, with the same values of the regularization parameter σ. One can see that in all three cases the surface separating the normal data from the anomalies correctly identifies the normal region as lying in the vicinity of the origin. The particular form of the separating surface – ranging from a more ragged to a circular surface as the regularization increases – depends on the value of the parameter σ.

5 EXPERIMENTS

To evaluate the proposed quarter-sphere formulation, experiments are carried out on the well-known KDDCup 1999 dataset. This dataset contains certain pre-defined features computed over the network traffic data collected in 1998 DARPA IDS evaluation. Each feature vector contains the total of 37 features and corresponds to one TCP/IP connection. The list of features and their brief descriptions can be found in Table 2 in the Appendix C.

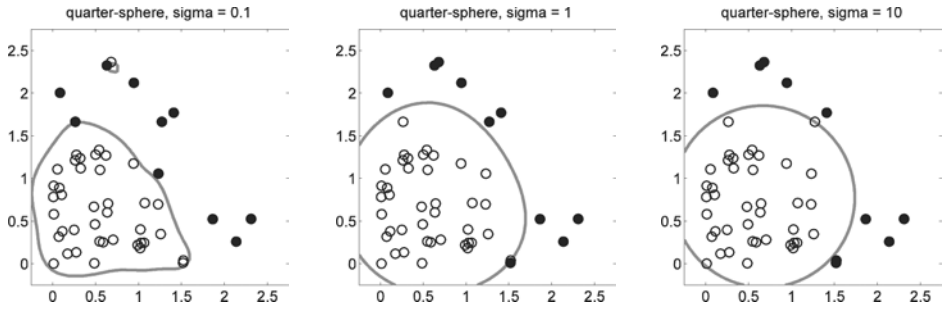


Fig. 6 Separation surfaces of the one-class SVM (quarter-sphere) for different values of the parameter σ

5.1 Comparison of one-class SVM formulations

We first compare the quarter-sphere one-class SVM with the other two algorithms using the RBF kernel. Since the sphere and the plane formulations are equivalent for the RBF kernels, identical results are produced for these two formulations (only sphere is shown).

Since the data comprises a variety of numerical and categorical attributes – each computed on its own scale – the first problem that needs to be taken care of is normalization, i.e. transforming the data into one consistent scale. This is accomplished, as proposed in [EAP+02], by the following procedure:

1. For each numerical attribute, the mean and the standard deviation are computed, and the values of the attributes are replaced by their distances from the mean divided by the standard deviation.
2. For each categorical attribute, the number c of the attribute values is computed, and the attribute is replaced by c numerical attributes. If the original categorical attribute has the value x_k, $k \cdot c$, the k-th numerical attribute is set to the value $1/c$, and the remaining ones are set to zero.

Note that a large proportion (about 75%) of the records in the KDDCup dataset represent the attacks; in particular, a large number of connections arise from the denial of service attacks and probes. Since it is implicitly assumed in (unsupervised) anomaly detection methods that anomalies constitute only a small fraction of the data, we perform sub-sampling of the full dataset in order to reduce the number of anomalies.[1] The main comparison results reported in section 5.1 are obtained on the data containing 2% of attacks. In section 5.3. the dependence of detection rate on the attack percentage is investigated. To obtain statistically significant results, 10 sub-sampled datasets are used in each experiment, the average rates reported. Standard deviations are reported in the more detailed presentation of results in section 5.2.

The following criteria are used for the evaluation:

– *Detection rate:* the ratio of correctly detected attacks to the total number of attacks.
– *False alarm rate:* the ratio of normal records detected as attacks ("false alarms") to the total number of normal records.
– *Detection cost:* the number of false alarms per correct detection.

A joint presentation of the detection rate and the false alarm rate can be made by means of the so called *Receiver Operating Characteristic* (ROC) curve, in which the detection rate is plotted as a function of the false alarm rate. In the results presented below the ROC curves are plotted for the false alarm rates less than or equal to 0.1, since larger false alarm rates are unacceptable for the amount of data that needs to be processed by intrusion detection systems.

The experiments are carried out for two different values of the parameter σ of the RBF kernel: 1 and 12 (the latter value used in [EAP+02]). These values correspond to low and medium regularization. The performance of a method is measured by a ROC curve. To produce such a curve, we use the output of the algorithm which returns a score for each incoming point measuring its anomality. Both one-class SVM considered here use the distance from the center as a score. A decision which points constitute anomalies is made by fixing a threshold and considering all points whose scores exceed the threshold as anomalies. We test every possible decision threshold and evaluate – for every fixed threshold – the values of the detection rate and the false alarm rate. Finally, we average both rates over the 10 different datasets. In the presented ROC curves the average detection rate is plotted against the average false alarm rate.

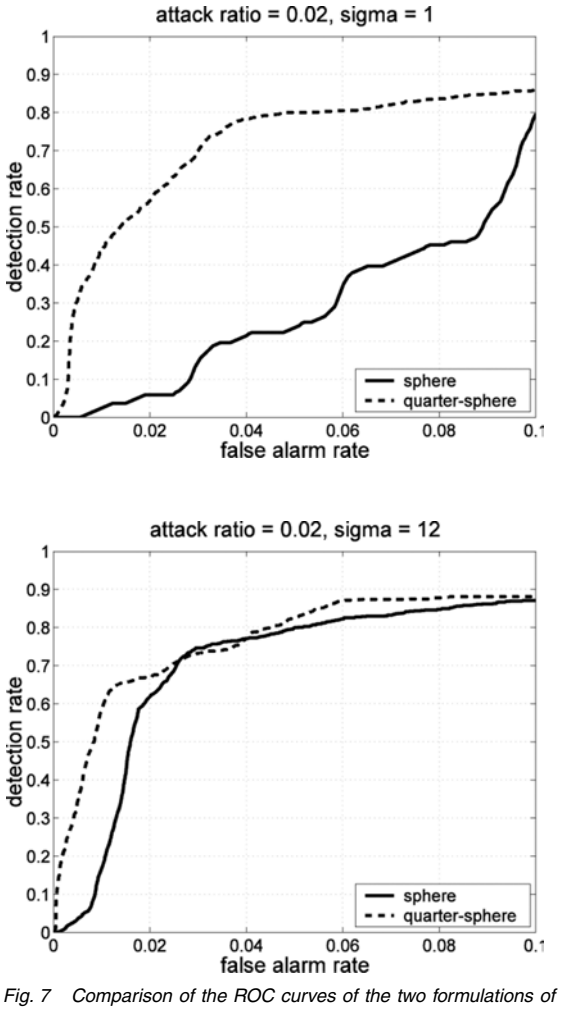


Fig. 7 Comparison of the ROC curves of the two formulations of one-class SVM.

1 Sub-sampling is also used in the previous work on unsupervised intrusion detection such as [PES01, EAP+02].

The comparison of ROC curves of the sphere and the quarter-sphere formulations of one-class SVM is shown in Fig. 7. It can be easily seen that the quarter-sphere formulation consistently outperforms the sphere formulation; especially for the low value of regularization parameter. The best overall results are achieved with the medium regularization with $\sigma = 12$, which has been most likely selected in [EAP+02] after careful experimentation. The advantage of the quarter-sphere in this case is not as dramatic as with low regularization but is nevertheless very significant in the low false alarm region.

5.2 Interpretation

The experiments presented above demonstrate the general advantage of the proposed quarter-sphere SVM formulation over the previous ones. To provide a more practical interpretation of our results, we investigate the distribution of detection and false alarm rates over the attack categories and the cost of detection in terms of the number of false alarms.

According to the well-known classification of Kendall [Ke99], the following *four types of attacks* are known:

- Denial of service (DOS) – attacks which prevent normal operation, such as causing the target host or server to crash, or blocking network traffic.
- Probing – surveillance and other probing, i.e. testing a potential target to gather information (e.g., port scanning).
- Remote to local (R2L) – unauthorized access from a remote machine, i.e. attacks in which an unauthorized user is able to bypass normal authentication and execute commands on the target (e.g. guessing password).
- User to root (U2R) – unauthorized access to local superuser (root) privileges.

Two attacks present in the KDDCup dataset cannot be classified into one of the four categories above and are defined as a fifth category "scenario", as they constitute specific exploits of known vulnerabilities. The summary of attacks present in the KDDCup dataset and their categorization is given in Table 3 in the Appendix C.

To perform the analysis of detection rates by attack category, we fix the set of false alarm rate values at 1%, 5% and 10%, and record the detection rate values for each attack category.[2] The mean values and the standard deviations of detection rates (over 10 sub-sampled experiments) are reported in Table 1 for the two formulation and the two values of the parameter σ of the RBF kernel. Additionally, the total detection rate (for all attacks) and the total detection cost, defined at the beginning of section 5, are reported for each case.

It can be seen from the Table 1 that both methods exhibit rather uniform detection rates over the 4 main categories of attacks, with probes and U2R attacks slightly better detected. The scenario attacks are detected noticeably worse, which can be explained by the fact that the features used in the KDDCup dataset are rather general and are not related to some particular exploits.

2 A single ROC curve consists of the piece-wise constant segments corresponding to the intervals on the false alarm axis. Therefore, one can always examine the detection rate for any given value of the false alarm rate by taking a respective interval. The results can be subsequently averaged over the number of experiments.

Table 1 Detection rates of the sphere and the quarter-sphere one-class SVM by attack category.

Attack category	mean	std	mean	std	mean	std
SPHERE, $\sigma = 1$						
	fa = 0.01		fa = 0.05		fa = 0.10	
dos	0.0269	0.0851	0.2291	0.3321	0.6826	0.3319
probe	0.0278	0.0878	0.2644	0.3716	0.7692	0.3091
R2L	0.0556	0.1757	0.2487	0.3347	0.6321	0.2509
U2R	0.0350	0.1107	0.2826	0.4174	0.7905	0.4177
scenario	0.0333	0.1054	0.2125	0.2789	0.5542	0.1735
Total detection rate	0.0370	0.1170	0.2500	0.3470	0.6940	0.3059
Total detection cost	N/A	N/A	N/A	N/A	14.0667	18.5988
SPHERE, $\sigma = 12$						
	fa = 0.01		fa = 0.05		fa = 0.10	
dos	0.2615	0.2811	0.8482	0.0503	0.8846	0.0334
probe	0.2194	0.2240	0.9531	0.0466	0.9944	0.0176
R2L	0.1030	0.1097	0.6581	0.0727	0.7876	0.0601
U2R	0.3665	0.3495	0.9457	0.0440	0.9660	0.0333
scenario	0.1306	0.1736	0.4778	0.1148	0.6361	0.0746
Total detection rate	0.2200	0.2196	0.8000	0.0350	0.8710	0.0179
Total detection cost	7.8102	8.1423	3.0680	0.1399	5.6279	0.1162
QUARTER-SPHERE, $\sigma = 1$						
	fa = 0.01		fa = 0.05		fa = 0.10	
dos	0.3811	0.1178	0.7688	0.0504	0.7770	0.0506
probe	0.3462	0.0649	0.9761	0.0310	0.9941	0.0186
R2L	0.4750	0.0928	0.6821	0.0421	0.8125	0.0387
U2R	0.6274	0.0842	0.9705	0.0255	1.0000	0.0000
scenario	0.2278	0.1316	0.5250	0.0909	0.6375	0.0829
Total detection rate	0.4390	0.0502	0.8000	0.0176	0.8580	0.0155
Total detection cost	1.1285	0.1205	3.0638	0.0680	5.7126	0.1033
QUARTER-SPHERE, $\sigma = 12$						
	fa = 0.01		fa = 0.05		fa = 0.10	
dos	0.6010	0.0295	0.8454	0.0578	0.8727	0.0431
probe	0.7296	0.1849	0.9819	0.0291	1.0000	0.0000
R2L	0.4418	0.0826	0.7111	0.0739	0.8269	0.0306
U2R	0.7258	0.0627	0.9512	0.0326	0.9902	0.0206
scenario	0.3417	0.1096	0.5347	0.1066	0.6028	0.0738
Total detection rate	0.5810	0.0561	0.8250	0.0438	0.8820	0.0199
Total detection cost	0.8514	0.0938	2.9776	0.1653	5.5582	0.1282

It can also be clearly seen that performance of the sphere one-class SVM on low false alarm rates is erratic and, in general, unsatisfactory. This is manifested in very high standard deviations of the detection rate, and very high detection costs (N/A is assigned to the case when at least in one experiment no attacks were detected for the given false alarm rate, which makes the denominator of the detection cost vanish). On the contrary, the quarter-sphere one-class SVM is quite reliable, even at low false alarm rates, and attains an astonishing detection cost of less than one false alarm per detection at fa = 0.01 and $\sigma = 12$. It can be also observed that, in general, attempting to increase detection rates by allowing higher false alarm rates leads to increasing detection cost; therefore, a solution should be sought in improving detection accuracy at low false alarm rates.

5.3 Dependency on the ratio of anomalies

The assumption that intrusions constitute a small fraction of the data may not be satisfied in a realistic situation. Some attacks, most notably the denial-of-service attacks, manifest themselves precisely in a large number of connections. Therefore, the problem of a large ratio of anomalies needs to be addressed.

In the experiments in this section we investigate the performance of the sphere and the quarter-sphere one-class SVM as a function of the attack ratio. It is known from the literature [TD99, SPST+01] that the parameter ν of the one-class SVM can be interpreted as an upper bound on the ratio of the anomalies in the data. The effect of this parameter on the quarter-sphere formulation is different: it specifies that *exactly ν fraction of points is expected to be the anomalies*. This is agreeably a more stringent assumption, and methods for the automatic determination of the anomaly ratio must be further investigated. Herein we perform a simple comparison of the algorithms under the assumption that ν matches the anomaly ratio; i.e. it is assumed that perfect information about the anomaly ratio is available.

One would expect that the parameter ν can tune both kinds of one-class SVM to the specific anomaly ratio. This, however, does not happen, as can be seen from Fig. 8. One can observe that the performance of both formulations noticeably degrades with the increasing anomaly ratio. We believe that the reason for this lies in the data-dependent normalization of the features: since the features are normalized with respect to the mean, having a larger anomaly ratio shifts the mean towards the anomalies, which leads to worse separability of the normal data and the anomalies.

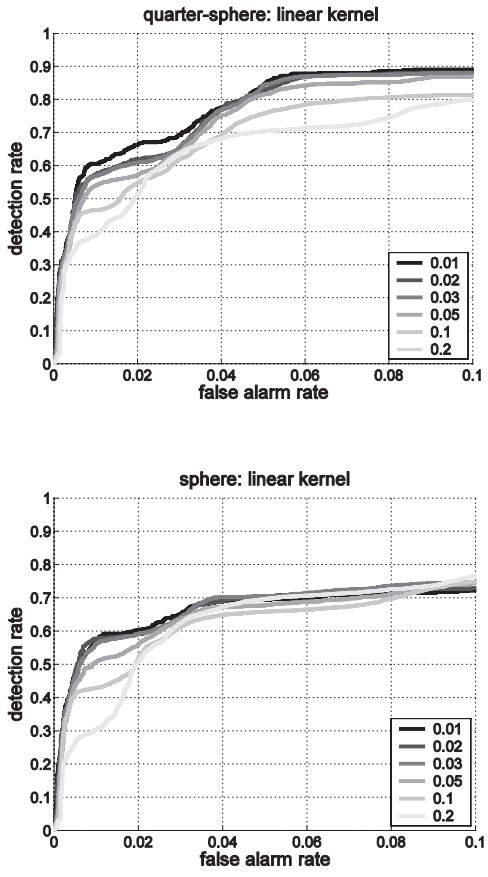


Fig. 8 Impact of the anomaly ratio on the accuracy of the sphere and quarter-sphere SVM.

6 CONCLUSIONS AND FUTURE WORK

We have presented a novel one-class SVM formulation, the quarter-sphere SVM, that is optimized for non-negative attributes with one-sided distribution. Such data is frequently used in intrusion detection systems. The classical one-class SVM formulations previously applied in the context of unsupervised anomaly detection do not account for non-negativity and one-sidedness; as a result, they can potentially detect very common patterns, their attributes close to the origin, as anomalies. Our new quarter-sphere SVM avoids this problem by simply aligning the center of the sphere fitted to the data with the "center of mass" of the data in feature space.

Our experiments conducted on the KDDCup 1999 dataset demonstrate significantly better accuracy of the quarter-sphere SVM in comparison with the previous, sphere or plane, formulations. Especially noteworthy are the strong advantages of the new algorithm at low false alarm rates, and the low cost of detection in terms of false alarm (in the best case, less than one false alarm per detection).

We have also investigated the behavior of one-class SVM as a function of attack rate. It is shown that the accuracy of all three formulations of one-class SVM considered here degrades with the growing percentage of attacks, contrary to the expectation that the parameter of one-class SVM, if properly set, should tune it to the required anomaly rate.

We have found that the performance degradation with the perfectly set tuning parameters is essentially the same as when the parameter is set to some arbitrary value. We believe that performance of anomaly detection algorithms on higher anomaly rates should be given special attention in the future work, especially with respect to the data normalization techniques.

A further line of research will consider anomaly detection using one-class boosting algorithms (cf. [RMSM02]) and ranking approaches.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge partial funding from the *Bundesministerium für Bildung und Forschung* under the project MIND (FKZ 01-SC40A) and the PASCAL Network of Excellence, EU # 506778. We also thank Stefan Harmeling and Rolf Schulz for valuable suggestions and discussions.

Appendix A: PLANE ONE-CLASS SVM FORMULATION

Mathematically, the problem of separating the data from the origin with the largest possible margin is formulated as follows [SPST+01]:

$$\min_{w\in F,\xi\in R^l,\rho\in R} \tfrac{1}{2}\|w\|^2 + \tfrac{1}{\nu l}\sum_{i=1}^{l}\xi_i - \rho,$$

$$\text{subject to: } (w\,\Phi\,(x_i)) \geq \rho - \xi_i, \qquad (3)$$

$$\xi_i = 0.$$

The weight vector w, characterizing the hyperplane, "lives" in the feature space F, and therefore is not directly accessible (as the feature space may be extremely high-dimensional). The

non-negative slack variables ξ_i allow for some points, the anomalies, to lie on the "wrong" side of the hyperplane. Instead of the primal problem (3), the following dual problem, in which all the variables have low dimensions, is solved in practice:

$$\min_{\alpha \in R^l} \sum_{ij=1}^{l} \alpha_i \alpha_j k(x_i, x_j),$$

$$\text{subject to: } \sum_{i=1}^{l} \alpha_i = 1, \tag{4}$$

$$0 \le \alpha_i \le \frac{1}{vl}.$$

Once the solution α is found, one can compute the threshold parameter $\rho = \sum_j \alpha_j k(x_i, x_j)$, for some example i such that a_i lies strictly between the bounds (such points are called *support vectors*). The decision, whether or not point x is normal, is computed as:

$$f(x) = \text{sgn}(\sum_i \alpha_i k(x_i, x) - \rho).$$

The points with $f(x) = -1$ are considered to be anomalies.

Appendix B: SPHERE ONE-CLASS SVM FORMULATION

Mathematically the problem of "soft-fitting" the sphere over the data is described as follows [TD99]:

$$\min_{r \in R, \xi \in R^l, c \in F} r^2 + \tfrac{1}{vl} \sum_{i=1}^{l} \xi_i,$$

$$\text{subject to: } \|\Phi(x_i) - c\| \le r^2 + \xi_i, \tag{5}$$

$$\xi_i = 0.$$

Similarly to the primal formulation (3) of the plane one-class SVM, one cannot directly solve the primal problem (5) of the sphere formulation, since the center c belongs to the possibly high-dimensional feature space. The same trick can be employed the solution is sought to the dual problem:

$$\min_{\alpha \in R^l} \sum_{ij=1}^{l} \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^{l} \alpha_i k(x_i, x_i),$$

$$\text{subject to: } \sum_{i=1}^{l} \alpha_i = 1, \tag{6}$$

$$0 \le \alpha_i \le \frac{1}{vl}.$$

The decision function can be computed as:

$$f(x) = \text{sgn}(R^2 - \sum_{ij=1}^{l} \alpha_i \alpha_j k(x_i, x_j) + 2\sum_{i=1}^{l} \alpha_i k(x_i, x) - k(x, x)).$$

The radius r^2 plays the role of a threshold, and, similarly to the plane formulation, it can be computed by equating the expression under the "sgn" to zero for any support vector.

The similarity between the plane and the sphere formulations goes beyond merely an analogy. As it was noted in [SPST+01], for kernels $k(x, y)$ which depend only on the difference $x - y$, the linear term in the objective function of the dual problem (9) is constant, and the solutions are equivalent.

Appendix C FEATURES AND ATTACKS IN THE KDDCUP DATASET

Table 2 Features of the KDDCup dataset.

Name	Description
duration	Duration of connection
protocol_type	Protocol type
service	Network service on the destination, (http, telnet)
flag	Normal or error status of the connection
src_bytes	Number of data bytes sent from source to destination
dst_bytes	Number of data bytes sent from destination to source
land	1 if the connection is from/to the same host/port; 0 otherwise
wrong_fragment	Number of "wrong" fragments
urgent	Number of urgent packets
hot	Number of "hot" indicators; count of access to system directories, creation and execution of programs, etc.
num_failed_logins	Number of failed login attempts
logged_in	Whether the user successfully logged in (using telnet, rsh, etc.)
num_compromized	Number of "compromised" conditions; count of file/path "not found" errors and "jump to" instructions, etc.
root_shell	1 if root shell is obtained; 0 otherwise
su_attempted	Whether a 'su' command is issued
num_root	Number of root accesses
num_file_creations	Number of file creation operations
num_shells	Number of shell prompts
num_access_files	Number of write, delete, and create operations on access control files
num_outbound_cmds	Number of outbound commands in an ftp session
is_hot_login	Whether the login belongs to the "hot" list
is_guest_login	Whether the login belongs to the "guest" list
count	Number of connections in the past 2 sec with the same destination IP as the current connection
srv_count	Number of connections in the past 2 sec with the same service as the current connection
serror_rate	% of connections to the same destination in the past 2 sec with SYN errors
srv_serror_rate	% of connections with the same service in the past 2 sec with SYN errors
rerror_rate	% of connections to the same destination in the past 2 sec with REJ errors
srv_rerror_rate	% of connections with the same service in the past 2 sec with REJ errors
same_srv_rate	% of connections to the same destination in the past 2 sec with the same service as the current connection
diff_srv_rate	% of connections to the same destination in the past 2 sec with the different service from the current connection
srv_diff_host_rate	% of connections with the same service in the past 2 sec to the different destination as the current connection

Name	Description
dst_host_count	Number of connections among the previous 100 connections to the same host with the same destination IP as the current connection
dst_host_srv_count	Number of connections among the previous 100 connections to the same host with the same service as the current connection
dst_host_same_srv_rate	% of connections among the previous 100 connections to the same host with the same service as the current connection
dst_host_diff_srv_rate	% of connections among the previous 100 connections to the same host with the different service from the current connection
dst_host_same_src_port_rate	% of connections among the previous 100 connections to the same host originating from the same source port as the current connection
dst_host_src_diff_host_rate	% of connections among the previous 100 connections to the same host originating from a different source IP from the current connection
dst_host_serror_rate	% of connections among the previous 100 connections to the same host with SYN errors
dst_host_serror_rate	% of connections among the previous 100 connections with the same service with SYN errors
dst_host_rerror_rate	% of connections among the previous 100 connections to the same host with REJ errors
dst_host_rerror_rate	% of connections among the previous 100 connections with the same service with REJ errors

Table 3 Taxonomy of attacks in the KDDCup dataset.

Attack type	Attack name	Attack type	Attack name
DoS	smurf	R2L	phf
DoS	pod	R2L	imap
DoS	apache2	R2L	xsnoop
DoS	udpstorm	R2L	worm
DoS	processtable	R2L	sendmail
DoS	neptune	R2L	ftp_write
DoS	back	R2L	guess_passwd
DoS	mailbomb	R2L	snmpguess
DoS	teardrop	U2R	loadmodule
Probe	saint	U2R	xterm
Probe	portsweep	U2R	perl
Probe	satan	U2R	ps
Probe	mscan	U2R	buffer_overflow
Probe	ipsweep	U2R	rootkit
Probe	nmap	U2R	land
R2L	warezmaster	U2R	sqlattack
R2L	named	scenario	snmpgetattack
R2L	xlock	scenario	httptunnel

REFERENCES

[BCJ+01] Barbará, D.; Couto, J.; Jajodia, S.; Popyack, L.; Wu, N.: ADAM: Detecting intrusions by data mining. In: Proc. IEEE Workshop on Information Assurance and Security, pages 11-16. 2001.

[CP] CPLEX. http://www.ilog.com/products/cplex/

[De87] Denning, D.: An intrusion-detection model. In: IEEE Transactions on Software Engineering. 13:222-232. 1987.

[DR90] Dowell, C.; Ramstedt, P.: The ComputerWatch data reduction tool. In: Proc. 13th National Computer Security Conference, pages 99-108. 1990.

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

[JLA+93] Jagannathan, R.; Lunt, T.F.; Anderson, D.; Dodd, C.; Gilham, F.; Jalali, C.; Javitz, H.S.; Neumann, P.G.; Tamaru, A.; Valdes, A.: Next-generation intrusion detection expert system (NIDES). Technical report. Computer Science Laboratory, SRI International. 1993.

[Ke99] Kendall, K.: A database of computer attacks for the evaluationof intrusion detection systems. M.Sc. Thesis, MIT, 1999.

[LEK+03] Lazarevic, A.; Ertoz, L.; Kumar, V.; Ozgur, A.; Srivastava, J.: A comparative study of anomaly detection schemes in network intrusion detection. In: Proc. SIAM Conf. Data Mining. 2003.

[LS] LibSVM. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[LV92] Liepins, G.; Vaccaro, H.: Intrusion detection: its role and validation. In: Computers and Security. 11 (4):347-355. 1992.

[MMR+01] Müller, K.-R.; Mika, S.; Rätsch, G.; Tsuda, K.; Schölkopf, B.: An introduction to kernel-based learning algorithms. In: IEEE Transactions on Neural Networks. 12 (2):181-201. 2001.

[MSL+04] Mika, S.; Schäfer, C.; Laskov, P.; Tax, D.; Müller, K.-R.: Support Vector Machines. Chapter III.15. In: Gentle, J.; Härdle, W.; Mori, Y.: Handbook of Computational Statistics. Springer-Verlag, 2004.

[NWY02] Noel, S.; Wijesekera, D.; Youman, C.: Applications of Data Mining in Computer Security. Chapter Modern intrusion detection, data mining, and degrees of attack guilt. Kluwer. 2002.

[PES01] Portnoy, L.; Eskin, E.; Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: Proc. ACM CSS Workshop on Data Mining Applied to Security. 2001.

[PG90] Poggio, T.; Girosi, F.: Regularization algorithms for learning that are equivalent to multilayer networks. In: Science. 247: 978-982. 1990.

[PN97] Porras, P.A.; Neumann, P.G.: Emerald: event monitoring enabling responses to anomalous live disturbances. In: Proc. National Information Systems Security Conference, pages 353-365. 1997.

[RMSM02] Rätsch, G.; Mika, S.; Schölkopf, B.; Müller, K.-R.: Constructing boosting algorithms from SVMs: an application to one-class classification. In: IEEE PAMI. 24 (9):1184-1199. September 2002.

[SL] SVM^{Light}. http://svmlight.joachims.org/

[SMB+99] Schölkopf, B.; Mika, S.; Burges, C.; Knirsch, P.; Müller, K.-R.; Rätsch, G.; Smola, A.: Input space vs. feature space in kernel-based methods. In: IEEE Transactions on Neural Networks. 10 (5):1000-1017. September 1999.

[SPST+01] Schölkopf, B.; Platt, J.; Shawe-Taylor, J.; Smola, A.; Williamson, R.: Estimating the support of a high-dimensional distribution. In: Neural Computation. 13 (7):1443-1471. 2001.

[SS98] Smola, A.; Schölkopf, B.: On a kernel-based method for pattern recognition, regression, approximation and operator inversion. In: Algorithmica. 22: 211-231. 1998.

[SS02] Schölkopf, B.; Smola, A.: Learning with Kernels. MIT Press. Cambridge, MA. 2002.

[SSM98] Schölkopf, B.; Smola, A.; Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. In: Neural Computation. 10: 1299-1319. 1998.

[TD99] Tax, D.; Duin, R.: Data domain description by support vectors. In: Verleysen, M. (Hrsg.): Proc. ESANN. S. 251-256. Brussels. 1999. D. Facto Press.

[Va95] Vapnik, V.: The nature of statistical learning theory. Springer Verlag. New York. 1995.

[VS00] Valdes, A.; Skinner, K.: Adaptive, model-based monitoring for cyber attack detection. In: Proc. RAID 2000, pages 80-92. 2000.

[WFP99] Warrender, C.; Forrest, S.; Perlmutter, B.: Detecting intrusions using system calls: alternative data methods. In: Proc. IEEE Symposium on Security and Privacy., pages 133-145. 1999.