POCAD: a Novel Payload-based One-Class Classifier for Anomaly Detection

Xuan Nam Nguyen, Dai Tho Nguyen*
University of Engineering and Technology
Vietnam National University, Hanoi
* UMI 209 UMMISCO IRD/UPMC
nguyendaitho@vnu.edu.vn

Long Hai Vu
IBM T.J. Watson Research Center
New York, United States
lhvu@us.ibm.com

***Abstract*— In this paper, we propose a novel Payload-based One-class Classifier for Anomaly Detection called POCAD, which combines a generalized 2_v-gram feature extractor and a one-class SVM classifier to effectively detect network intrusion attacks. We extensively evaluate POCAD with real-world datasets of HTTP-based attacks. Our experiment results show that POCAD can quickly detect malicious payload and achieves a high detection rate as well as a low false positive rate. The experiment results also show that POCAD outperforms state of the art payload-based detection schemes such as McPAD [4] and PAYL [8].**

I. INTRODUCTION

Intrusion Detection Systems (IDS) are powerful tools for the defense-in depth of computer networks. One of the leading approaches is anomaly detection, based on specification of normal or benign activities. This approach usually suffers from high false positive rate issues, but it is able to detect zero-day attacks. As it is very hard and expensive to achieve a labeled dataset for real network activities containing both normal and attack traffic, unsupervised or unlabeled learning approaches for network anomaly detection have been recently suggested. One-class classification algorithms pursue the concept of machine learning in absence of counter examples, and have been shown to be promising for network anomaly detection.

A. Payload-based Anomaly Detection

Many works have been carried out on unlabeled anomaly detection and focused on high speed classification using simple payload statistics [1, 5] (a payload is the actual data of a network packet). For instance, PAYL [8] extracts 256 features from the payload, each of which represents the occurrence frequency in the payload of one of 256 possible byte values. Although PAYL is based on simple statistics extracted from the payload, it has been shown to be quite effective [8]. Nonetheless, PAYL may suffer from a relatively high false positive rate [4]. A more generic *n*-grams version of PAYL has been proposed by the same authors [7]. A sliding window with length *n* is used to extract the occurrence frequency in the payload of all the possible *n*-grams. In this way, the payload is represented by a pattern vector in a 256^n-dimensional feature space. The obtained model is more precise than the simple byte frequency model. However, with the exponentially rising number of extracted features, the higher *n*, the more difficult it may be to construct an accurate model due to the curse of dimensionality and computational complexity. Perdisci et al. proposed McPAD, a multiple classifier system for anomaly detection with a 2_v-gram feature extraction scheme that counts the appearance frequency of any two byte values in all pairs of bytes that are *v* bytes apart in the packet payload [4]. This approach limits the number of dimensions to 256^2 and captures part of the structural information in the payload. However, it fails to capture the structural patterns within the *v* bytes of each pair separated by *v* bytes.

B. Our contribution

We propose a new Payload-based One-class Classifier for Anomaly Detection called POCAD that combines an improved version of the 2_v-gram scheme and a one-class SVM classifier to achieve high detection rate and low false positive rate at the same time. Instead of counting only the pairs of byte values that are *v* bytes apart in the payload like McPAD, POCAD takes into account all pairs of byte values separated by less than or equal to *v* bytes. Specifically, for each pair of byte values, POCAD first counts the appearance frequency of this pair separated by exactly *k* bytes for each *k* from 0 to *v*. Then, the counts are added up to give the feature value of the two byte values. Doing this, POCAD successfully captures the correlation and associations of close byte values in the packet payload. We use these features to train a one-

class SVM classifier and conduct extensive performance evaluations. We evaluate POCAD with different HTTP-based attacks, including Shell-code, CLET, Polymorphic Blending, etc. Our evaluation results show that POCAD not only quickly detects these attacks but also achieves a high detection rate and a low false positive rate. In summary, we make the following contributions:

- We propose POCAD, a novel payload-based anomaly detection scheme, which extracts correlated features from nearby bytes in the packet payload and constructs a one-class SVM classifier.
- We extensively evaluate POCAD with real-world datasets of different types of HTTP-based attacks. Our evaluation shows that POCAD outperforms state of the art schemes including McPAD [4] and PAYL [8], and achieves a very high detection rate at an extremely low false positive rate.

This paper is organized as follows. We discuss the background in Section II. Section III presents the details of the POCAD payload-based anomaly detection scheme and Section IV presents our evaluation results. Finally, we conclude the paper in Section V.

II. BACKGROUND

A. One-Class Classification

One-class classification techniques are specifically useful in case of two-class learning problems whereby one of the classes, referred to as the target class, is well-sampled, whereas the other one, referred to as the outlier class, is severely under-sampled. The small number of instances from the outlier class may be explained by the fact that it is too difficult or overpriced to acquire a significant number of training patterns of that class [6]. This is especially true in the network anomaly detection context since an extremely small number of network incidents is anomalous or malicious, meanwhile the rest of network activities is benign. So, one-class classifiers fit network anomaly detection naturally. One-class SVM has been shown to achieve very competitive performance in text classification problems [2, 7]. The payload anomaly detection problem using *n*-gram frequencies as features is analogous to text classification since both use the bag-of-words model in which a simple unweighted raw frequency vector representation is used [3]. Therefore, we choose to make use of a one-class SVM classifier in POCAD. In what follows, we use a feature vector $x_i = [x_{i_1}, x_{i_2}, \ldots, x_{i_l}]$ to represent the payload π_i in a l-dimensional feature space F. In Section III, we explain how x_i is extracted.

III. POCAD: A MULTIPLE 2_v-GRAM ONE-CLASS CLASSIFIER

A. Feature extraction

The PAYL's detection model relies on the counting of the number of appearances of *n*-grams (i.e., sequences of *n* successive bytes) in the payload [8]. The appearance frequencies of the *n*-grams are measured using a sliding window of length *n*. This window slides over the payload with one-byte steps and counts the appearance frequency for all 256^n possible *n*-grams. As a result, the payload is represented by a vector in the 256^n-dimensional feature space. There are two issues with this approach. On the one hand, if *n* is small ($n = 1$ or $n = 2$), little structural information is captured by this feature vector. So, the vector might not be useful in the detection of malicious patterns. On the other hand, for a larger *n* ($n > 2$), the number of dimensions increases exponentially and thus leads to the curse of dimensionality [7].

To solve the above problem, McPAD uses an ensemble of classifiers, each classifier corresponds to a 2_v-gram scheme that counts the appearance frequency of any two byte values in all pairs of bytes that are v bytes apart in the payload [4]. Note that, we have 256^2 combinations of two byte values in total. This approach limits the number of dimensions to 256^2 and captures part of the structural information in the payload. However, each classifier is not expressive enough to capture the structural patterns within the v bytes of a 2_v-gram. In other words, it only captures the information at the j^{th} byte and the $(j+v)^{th}$ byte but does not make use of bytes in the range $[j+1, j+v-1]$. This information might be important since patterns in practice may not always be at fixed places, and thus the distance between them might not be always a constant. For example, in a packet payload of an exploit, hackers may not always keep the command sequence of `OPEN`, `NOOP`, `JUMP` in all bytecode deliveries of the same sample. Instead, they may change this sequence and make a new command sequence of `OPEN`, `JUMP`, `NOOP` in some (and random) deliveries. The two executable programs run exactly the same once loaded, since `NOOP` command is ignored in program execution, but their actual bytecode contents are not the same.. This will break detectors that use a fixed 2_v-gram feature extraction scheme, like McPAD. Although McPAD combine many classifiers, but as each one is not good enough,

its overall efficiency is limited. To overcome the problem, for any two byte values, POCAD captures all structural information in the range [*j*, *j*+*v*] as follows. It first counts the appearance frequency of these two byte values that are *k* bytes apart for each *k* from 0 to v. Then, POCAD adds these frequencies up and use the addition sum as the appearance frequency of the given two byte values. As a result, POCAD still has 256^2 features but the feature values are more expressive than McPAD.

The main idea of POCAD is that two bytes *not too far apart* from each other in the payload may have some correlation or association. Each appearance of these two bytes together can be an additional evidence of the desired patterns we are looking for to detect abnormal contents, so it's important to count all of them. For example, assuming $v = 5$, we have six feature vectors extracted by six 2_k-gram feature extractors with *k* varying from 0 to 5 (2_0-gram, 2_1-gram, 2_2-gram, 2_3-gram, 2_4-gram, and 2_5-gram). By making the sum of these six vectors we have a new vector representing the number of appearances of each of the possible pairs of byte values that are distanced from each other by at most 5 bytes instead of exactly 0, 1, 2, 3, 4 or 5 bytes. For example, if one element *X* have 3 possible values *A*, *B*, and *C*, and we have a payload *AABCCABAA*, then the appearance frequency vector of 2_0-grams is as below:

AA	BB	CC	AB	BA	BC	CB	AC	CA
2	0	1	2	1	1	0	0	1

(The appearance frequency of pair AA is 2)

The appearance frequency vector of 2_1-grams:

AA	BB	CC	AB	BA	BC	CB	AC	CA
1	0	0	1	1	1	1	1	1

The sum of two appearance frequency vectors:

AA	BB	CC	AB	BA	BC	CB	AC	CA
3	0	1	3	2	2	1	1	2

Note that, we assume that each element has 3 possible values instead of 256 because a table of 256 columns is too big to be listed.

B. Payload classification

After obtaining the sum of appearance frequency vectors for all 256^2 features, we use a clustering algorithm to reduce dimensions as presented in [4]. Then, we construct a model of normal traffic by training a one-class SVM classifier using the same method as [4]. Fig. 1 represent our training phase. The one-class SVM classifier uses a predefined threshold τ to determine if an abnormal payload is found, that is if $P(\texttt{normal} \mid p) \leq \tau$. The threshold τ is used to control the tradeoff between the true and false positive rates of the trained model. Obviously, if τ is large, then we can detect most attacks but also have many false alarms. In contrast, a small value of τ leads to most normal packets being treated truly as normal, but the system can allow an attacker to go through with a high probability.

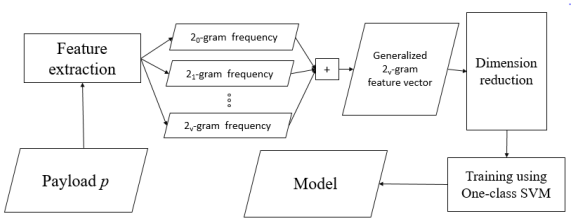

Fig. 1. Training phase

During the testing and production phases, with each payload *p* introduced as input, we compute the representation vector of *p*, using the same feature extraction and dimensionality reduction steps as in the training phase. Then, we classify this representation using the trained one-class SVM. Fig.2 shows the testing phase of POCAD.

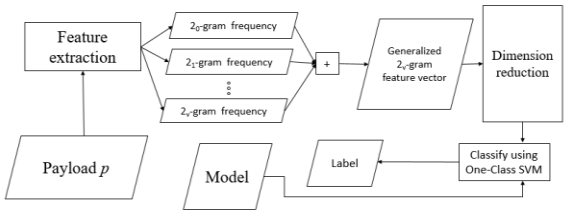

Fig. 2. Testing phase

IV. EVALUATION

A. POCAD implementation

We extend the McPAD open-source program[1] to implement POCAD. Specifically, we developed a new module for extraction of more generalized 2_v-gram features and extend the original packet classification module to implement our proposed packet classification technique. In our experiments, we vary the values of the threshold τ to thoroughly evaluate the performance of POCAD. The number of dimensions is reduced from 256^2 to 160 in the clustering algorithm, which is also the number of dimensions used for McPAD [4]. We let $v = 1$ in the feature extraction process, that is, 2_0-grams and 2_1-grams are combined together for the calculation of feature vectors. With McPAD, we limit the number of

[1] http://roberto.perdisci.googlepages.com/mcpad

classifiers in ensemble to 10, and use the maximum combination rule, which has been shown to produce the best performance for McPAD. The performances of POCAD are compared with PAYL and McPAD over different realistic datasets and types of real-world attacks.

B. Validation Metrics

We use two metrics in performance evaluation, including the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). The ROC curve provides a method to visually show the trade-off between the false positive rate and the true positive rate for different values of the detection threshold τ [3]. The AUC shows the classification productivity of the classifier in the entire range of the false positive rate. More precisely, a higher AUC implies a better performance of detecting attacks from normal packets. However, an IDS with a false positive rate of higher than 10% might be not be usable in practice. So, we focus on the AUC in the range of [0, 0.1] since it is a meaningful performance indicator in practice.

C. Datasets

We use two datasets in our experiments. We first use HTTP requests extracted from the first week of DARPA'99 dataset[2], and call it DARPA dataset. Similar to previous works PAYL [8] and McPAD [4], DARPA is used to train our model and evaluate the false positive rate. Second, we take HTTP-based attacks from McPAD site[1] and call this dataset ATTACKS, which consists of following data subsets:

- Generic Attacks: this subset consists of 66 HTTP attacks[3]. Among these, 11 are categorized as shell-code attacks that carry executable code in the payload. The remaining attack categories include Failure to handle exceptional conditions, File disclosure, Information leak, Input validation error, Poor memory management, Poor resource management, Signed interpretation of unsigned value, URL decoding error.
- Shell-code Attacks: this subset contains the 11 shell-code attacks from the Generic Attacks data subset above.
- CLET Attacks: this subset contains 96 polymorphic attacks generated using the polymorphic engine CLET [6].
- Polymorphic Blending Attacks (PBAs): this subset is created based on the well-known Code-Red worm. PBAs attacks mimic the normal traffic with the same distribution of *n*-grams as normal packets. By doing this, they try to evade detection by payload-based anomaly IDS using *n*-gram analysis.

[2]https://www.ll.mit.edu/ideval/data/1999/training/week1/index.html.

D. Experiment results

In this section, we first present the performance results of POCAD. We then compare its performance with those of PAYL [8] and McPAD [4] on various types of attacks.

1) Validation of POCAD

TABLE I. AUC OBTAINED BY POCAD

Type of Attack	AUC
Generic Attacks	0.91425
Shell-code	0.99912
CLET	0.99831

Table I shows that POCAD achieves AUC values very close to 1 for different types of attacks. This indicates that POCAD can be used in practice. Among these types of attacks, Generic attacks have the lowest AUC since Generic attacks include attacks such that Information leak and File disclosure, which usually do not contain executable code. Instead, they contain irregularities in the packet payload used to exploit target systems. However, their payloads are very similar to normal packet payloads in byte distribution and structure, and thus make it difficult for POCAD to detect. Meanwhile, packets with Shell-code attacks and CLET attacks contain executable code in their payloads, and thus are detected by POCAD with a higher precision.

2) Comparing POCAD with PAYL and McPAD

In this section, we compare the performance of POCAD with those of PAYL [8] and McPAD [4]. Specifically, we use 4 types of attacks: Generic, Shell-code, CLET, and PBAs.

Figures 1, 2, and 3 show that for Generic, Shell-code, and CLET attacks, the true positive rate (or detection rate) of PAYL rapidly decreases for a false positive rate of less than $5*10^{-3}$. Meanwhile, McPAD and POCAD can detect these attacks with a higher precision while only incurring a very low false positive rate of much smaller than 10^{-3}. More importantly, POCAD outperforms McPAD in all these three types of attacks. For example, Figure 1 shows that for the false positive rate of 10^{-5}, the detection rate

of McPAD is about 62% while that of POCAD is about 61%. Besides, for the false positive rate of 1%, POCAD has a true positive rate of 95%, while that of McPAD is only 89%. With Shell-code attacks and CLET attacks, POCAD also achieves a slightly better performance than McPAD.

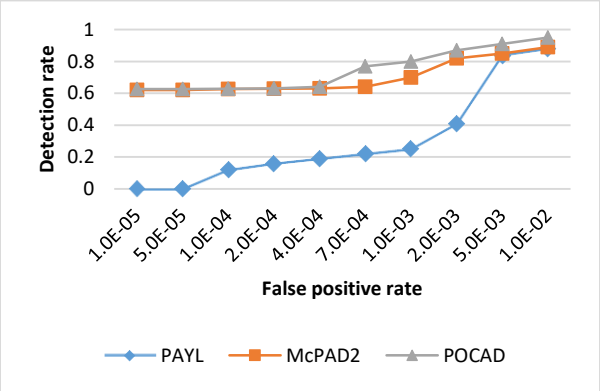

Fig. 3. Generic Attacks

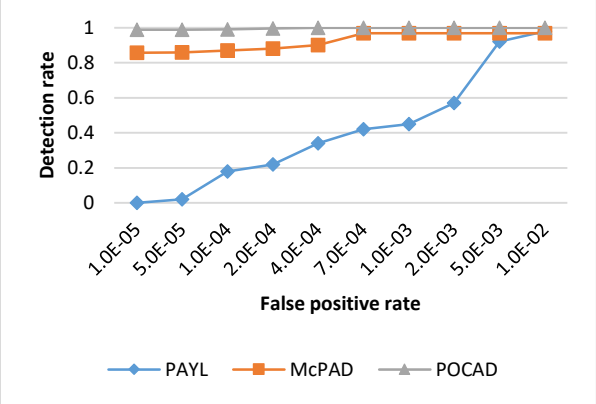

Fig. 4 Shell-code Attacks

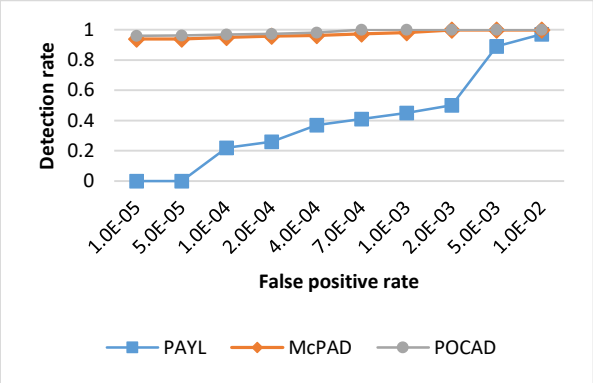

Fig. 5. CLET Attacks

Figures 4, 5, and 6 show the detection results on PBAs of PAYL, POCAD, and McPAD, respectively. For these experiments, we create PBAs that mimic the

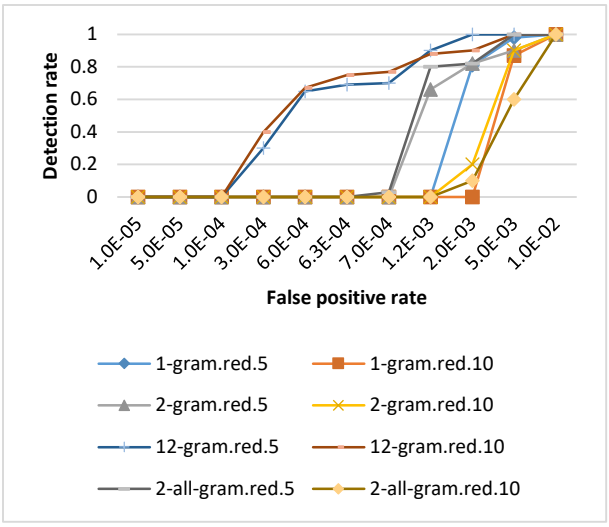

Fig. 6. ROC curves of PAYL for PBAs

statistical distribution of n-grams with n = 1, 2, 4, 12, and 2_v-grams with v = 1..10 spreading over either 5 or 10 overall attack packets as in [4]. Essentially, for a higher number of overall attack packets, PBAs mimic the distribution of normal traffic better, and thus they are more difficult to be detected. Figure 4 shows that PAYL achieves 0 detection rate for the false positive rate of roughly 10^{-3}. Meanwhile, Figures 5 and 6 show that POCAD and McPAD achieve a medium detection rates for 5-packet-length attacks. However, when attacks are spread over a larger number of packets (i.e., 10 packets), all three schemes fail for a very low false positive (i.e., 10^{-5}).

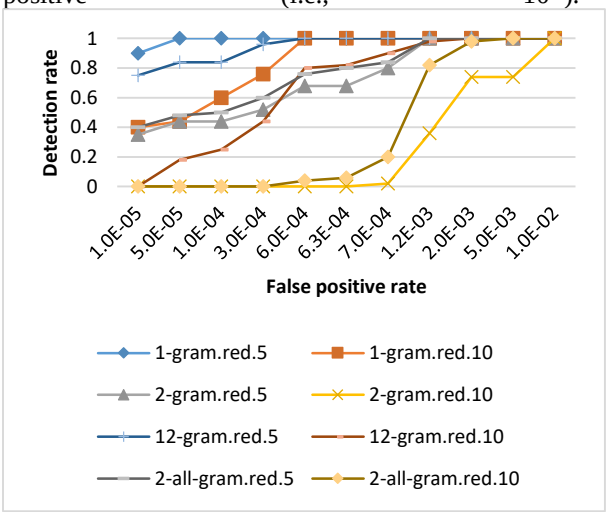

Fig. 7. ROC curves of POCAD for PBAs

Table II compare the average processing time per payload of PAYL, McPAD, and POCAD. While POCAD and PAYL are comparable, McPAD is significantly slower. This is because PAYL and

POCAD use only one classifier but McPAD uses ten classifiers and thus it takes much longer to run.

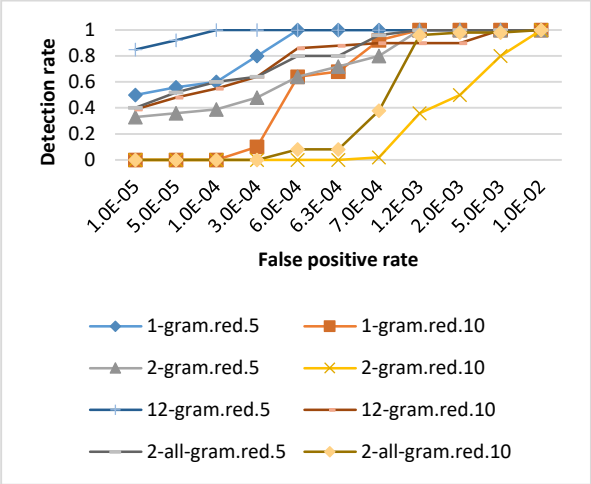

Fig. 8. ROC curves of McPAD for PBAs

TABLE II. AVERAGE PROCESSING TIME PER PAYLOAD

Detector	AVG processing time (ms)
PAYL	0.039
McPAD	13.39
POCAD	0.041

E. Discussion

Our experiment results show that POCAD is capable of detecting a variety of attacks even at very low false positive rates. Moreover, it can detect PBAs, a very challenging type of attacks. POCAD outperforms both PAYL and McPAD on all types of attacks, and achieves a significantly faster average detection time.

In [1], Axelsson showed that in order to have a good Bayesian detection rate we need to maintain a relatively high detection rate and meanwhile reduce the false positive rate to around 10^{-5}. As we show in Figures 2, 3, and 4, POCAD achieves this goal for different types of attacks. For example, it has a detection rate of around 98% for Shell-code attacks at a false positive rate of 10^{-5}. That means $P(\texttt{Alarm}|\texttt{Intrusion}) = 0.98$ and $P(\texttt{Alarm}|\texttt{Not Intrusion}) = 10^{-5}$. As in [1], we assume the probability $P(\texttt{Intrusion}) = 2\cdot10^{-5}$. Then, POCAD has the Bayesian detection rate $P(\texttt{Intrusion}|\texttt{Alarm}) = 0.98\cdot2\cdot10^{-5}/[0.98\cdot2\cdot10^{-5} + 10^{-5}\cdot(1-2\cdot10^{-5})] = 0.66$. Meanwhile, similar analysis of McPAD gives $P(\texttt{Intrusion}|\texttt{Alarm}) = 0.65$. For PAYL, the detection rate is zero at the false positive rate of 10^{-5}, and therefore the Bayesian detection rate is zero. All this confirms that POCAD outperforms both PAYL and McPAD in terms of effectiveness.

V. CONCLUSION

We design and implement POCAD, a novel payload-based anomaly detection scheme using an improved 2_v-gram feature extractor and a one-class SVM classifier to effectively detect network intrusion attacks. POCAD captures the correlation of nearby bytes in the packet payload and thus provides a promising method for payload-based anomaly detection. Our extensive performance evaluation of POCAD shows that it can quickly detect different types of HTTP-based attacks and consistently achieves a high detection rate as well as a very low false positive rate. POCAD also outperforms state of the art payload-based detection schemes such as McPAD [4] and PAYL [8]. We thus believe POCAD can be useful for payload-based intrusion detection in practice. Moving forward, we plan to study the optimal number of features to be extracted in the clustering step of Section II.B, and investigate the optimal value of v in the construction of generalized 2_v-gram features.

REFERENCE

[1] S. Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In CCS '99: Proceedings of the 6th ACM conference on Computer and communications security, pages 1–7, 1999

[2] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research, 3:1265–1287, 2003.

[3] E. Leopold and J. Kindermann. Text categorization with support vector machines. How to represent texts in input space? Machine Learning, 46:423–444, 2002.

[4] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, W. Lee. "McPAD : A Multiple Classifier System for Accurate Payload-based Anomaly Detection." Computer Networks, Special Issue on Traffic Classification and Its Applications to Modern Networks, 5(6), 2009, pp. 864-881.

[5] B. Sch¨olkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and RC Williamson. Estimating the support of a high-dimensional distribution. Neural Computation, 13:1443–1471, 2001.

[6] D. M. J. Tax. One-Class Classification, Concept Learning in the Absence of Counter Examples. PhD thesis, Delft University of Technology, Delft, Netherland, 2001.

[7] K. Wang and S. Stolfo. Anomalous payload-based worm detection and signature generation. In Recent Advances in Intrusion Detection (RAID), 2005.

[8] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In Recent Advances in Intrusion Detection (RAID), 2004.