Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 91–98

Flow–level Spam Modelling using separate data sources

Marcin Luckner
*Faculty of Mathematics and Information Science
Warsaw University of Technology
pl. Politechniki 1, 00-661 Warszawa, Poland
Email: mluckner@mini.pw.edu.pl

Robert Filasiak
†Orange Labs Poland
ul. Obrzeżna 7, 02-691 Warszawa, Poland
Email: Robert.Filasiak@orange.com

***Abstract*—Spam detection based on flow–level statistics is a new approach in anti–spam techniques. The approach reduces number of collected data but still can obtain relative good results in a spam detection task. The main problems in the approach are selection of flow–level features that describe spam and detection of discrimination rules. In this work, flow–level model of spam is presented. The model describes spam subclasses and brings information about major features of a spam detection task. The model is the base for decision trees that detect spam. The analysis of detectors, which was learned from data collected from different mail servers, results in the universal spam description consists of the most significant features. Flows described by selected features and collected on Broadband Remote Access Server were analysed by an ensemble of created classifiers. The ensemble detected major sources of spam among senders IP addresses.**

***Index Terms*—Spam detection, Flow analysis, Anomaly detection**

I. Introduction

RAPID development of the Internet and associated services induced growth of the desired bandwidth for their execution. Customers expect their Internet Service Providers (IS) to provide a flexible, fully secured access to the Internet. Requirements related to privacy and confidentiality increasingly important. The political environment inside European Union is discussing adjustment of law regulations to market needs. ISPs have to consider Quality of Service (QoS), security, Service Level Agreement (SLA) among others committed to privacy guarantee. This is one of the reasons for development of methods used for monitoring and analysis of traffic in the ISP's core network.

Two approaches are interesting from the perspective of multi–gigabit stream: packet header analysis and flow analysis. Both techniques do not use information contained in payload, which is very important from data reduction and privacy point of view. Additionally, the hash function can be executed on IP addresses to guarantee higher data protection. In some cases, data does not need to be stored. A good example is the statistical detection of Distributed Denial of Service (DDoS) attacks [1] and solutions developed on Field Programmable Gate Array (FPGA) cards [2]. Moreover, such limited data were successful used in an Internet traffic classification task [3].

The packet header analysis is focused processing of headers. The flow analysis is focused on sets of headers determined by a source, a destination IP, source and destination port, timestamps, etc. depending on parameters used to define a flow. The flow analysis enables more compact data reduction. IP Flow Information Export (IPFIX) [4] and NetFlow [5] are well known standards. The equivalent given by Juniper is named J-flow.

Regardless of what was an object of analysis (headers or flows), methods are based on a language that describes analysed events. Usually components of the language are values from packet headers, statistical values such as the average time, the maximal, or the minimal size of packets. Wide lists of defined primitives for flow analysis and honeypot detection are given in [6] and [7] respectively.

In all cases, the following schema is used:

1) select observed parameters $[(m_1, \ldots, m_n) \subset \{M\}]$ (metric),
2) capture values for the parameters $[m_1 \leftarrow .09, m_2 \leftarrow 11, \ldots, m_n \leftarrow false]$ (measurement),
3) calculate features $[f_1 \leftarrow m_1 * m_4, f_2 \leftarrow m_3 + m_5, \ldots]$(features),
4) determine logical relationships between features [if $(f_1 > f_2)$ then action$_1$ else action$_2$](decision).

One of the most important issues in the presented schema is a selection of features. The features that describe spam can be used to create its detection rules. In this paper, flow–level parameters $\{m\}$ selected by Žádník [8] as a subset of the set $\{M\}$ defined in [6] are used create a primary model of spam. In collected spam records, well–separate subclasses are detected (Section II-A). The comparison of subclasses defines important discriminants (Section II-B) that can be used to determine separation rules between spam and the rest of the flows (Section II-C).

Created model was trained and tested on separate data but collected from the same mail server. Therefore, a new data set was created and the most significant features were calculated once again on new data (Section III-A). Both sets of features were compared (Section III-B) and the comparison resulted in a universal set of features (Section III-C).

The final model was checked with Broadband Remote Access Server (BRAS) data (Section IV-A). Decision trees created on the base of both learning sets were used to detect

978-1-4673-4471-5/$25.00

spam among BRAS data (Section IV-B). The classification resulted in detection of main sources of spam (Section IV-C).

Conclusions from all tests are presented in Section V.

II. Spam model

The spam model is based on flows collected by Žádník and Michlovský [8]. The flows are defined by by NetFlow protocol that contains:

- source IP address,
- destination IP address,
- IP protocol,
- source port,
- destination port,
- IP type of service.

The NetFlow version 9 allows the user to collect additional features. The features collected in the flows are a subset of features presented in [6].

The authors collected data from the SMTP server hosting mailboxes the Liberouter project group. The data set contains over 58 thousand records described by 64 features and divided into several classes. Among classes, two describe spam. The first class *dnsbl* contains flows from IP address mentioned on DNS black lists. The second class *y_spam* consists of flows that have been successfully received and marked as spam by SpamAssassin.

In the case of the *dnsbl* class flows were labelled because of a source IP address. All flows send from the denied addresses are labelled as spam. For the second class *y_spam*, the labelling process is more complex. The flows are labelled in the off–line mode. The IP addresses and the time of arrival for the flow are compared with the SpamAssassin logs. If a mail with the same source IP address and the destination IP address was marked as spam then all flows with a similar time of arrival are labelled as spam.

The described division of spam is a consequence of used methodology and cannot be used as a framework of the model without any doubt. In the following section, the statistical methodology that creates spam subclasses is presented.

A. Detection of subclasses

Spam subclasses are created in two steps. Firstly, a clustering method is used to detect inner clusters. Next, a decision tree is created to find discrimination rules.

1) Clustering: The predefined spam subclasses *dnsbl* and *y_spam* are a consequence of used methodology. It should not be assumed that this division has a statistical base. Therefore, a new division is created using k–means clustering [9].

In the analysis, all members of *dnsbl* and *y_spam* are treated as a single class *spam*. The class consists of almost 54 thousand records.

A number of spam subclasses is unknown. Therefore, various values of k coefficient from the range $k \in [2, 25]$ are tested. The results of subsequent tests are compared in v–fold cross–validation process. In such test, random samples are drawn v times. Summary indices of the accuracy of the prediction are computed over the v replications. In the described case, the value v is fixed as 10.

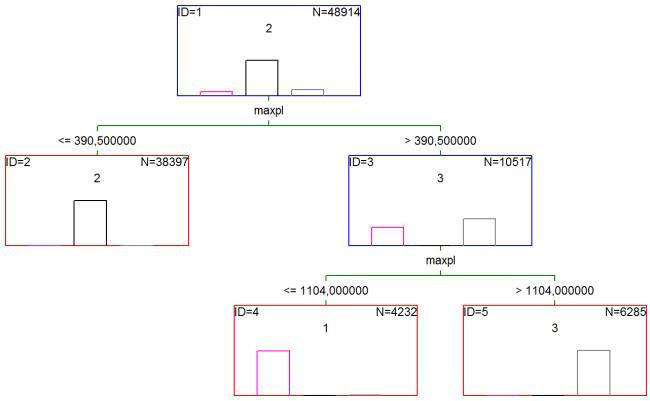


Fig. 1. The classification tree that divides spam into subclasses

A classification error calculated among cross–validation probes was smallest for division into four classes. In such case, the obtained error was 8 percent. However, one of the created classes has a relatively small cardinal number (632 records, which is about 1.3 percent of spam).

Significant differences between cardinal numbers of classes determine *a priori* probability used in a classification task. Members of classes with small *a priori* probability are classified as members of numerous classes to reduce the risk of misclassification.

To avoid future problems the number of classes was reduced to three. The cross–validation classification error increased over 8 percent, which is still an acceptable level. The smallest class was eliminated and the new distribution is more reasonable. The biggest class 2 contains 78.5 percent of record when classes 3 and 1 contain 12.9 and 8.6 percent respectively.

2) Separation rules: The second step of modelling creates discrimination rules between subclasses created during the clustering process. This task is done using a C&RT tree [10]. Such tree is not a very advanced classifier but creates clear decision rules.

The classification accuracy was over 99 percent. That proves a good division of spam into subclasses. Detailed information about classification errors is given in Table I.

TABLE I
THE MISCLASSIFICATION MATRIX FOR SUBCLASSES OF SPAM

	Observed 1	Observed 2	Observed 3
Predicted 1	4206	2	24
Predicted 2	1	38396	0
Predicted 3	6		6279

However, not the accuracy of classifier but the simplicity of created rules is the point to stress in this case. The classifier uses a single feature **maxpl**, which is the maximal length of a packet. The structure of classification tree is given in Figure 1.

Reasons for selection of **maxpl** as the most important discrimination factor are given in the next section.

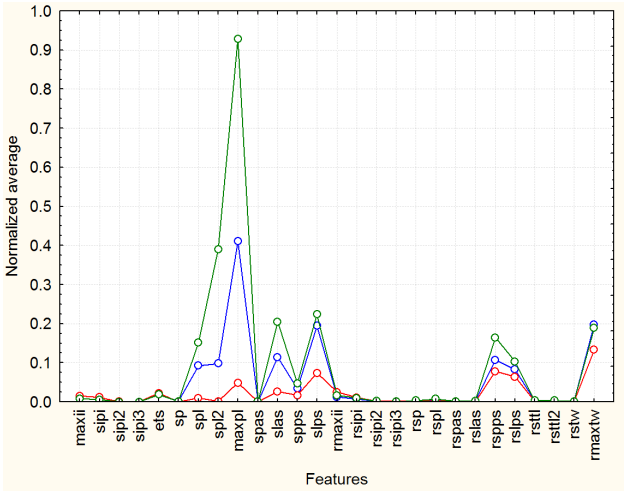


Fig. 2. Normalised average values of the most important spam features calculated separately for subclasses

B. Description of spam subclasses

A classification tree selects features because of the Gini coefficient [11] that is a measure of statistical dispersion. The feature significance is computed by summing over all nodes in the tree the drop in node impurity. The results are expressing relative to the largest sum found over all predictors where the largest sum gets 100 points. Details are given in [10].

The maximal length of package as the feature with major statistical dispersion should differ significantly among classes. It is proved by an analysis of normalised average of features for all three subclasses (presented in Figure 2). The differences between normalised average values calculated for separate subclasses are most significant for this feature. On the same basis, a set of important discriminants may be defined as:

spl Average package length,
spl2 Variation of package length,
maxpl Maximal package length,
slas Average length of package having the ACK flag,
slps Average length of package having the PUSH flag,
rspps Number of packages having the PUSH flag in response,
rslps Average length of package having the PUSH flag in response,
rmaxtw Maximal size of the response TCP window.

All mentioned features (except **rmaxtw**) take highest values for the subclass number 3 and the lowest for the subclass number 2. The subclass number 1 is somewhere between.

The presented analysis is limited to the normalised averages. More information is given by a probability density function. The function that was calculated for selected features (separately for all subclasses) is presented in Figure 3.

The presented normal distribution functions are determined by the following formula

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \qquad (1)$$

TABLE II
THE DISTRIBUTION OF *dnsbl* AND *y_spam* GROUPS AMONG SPAM SUBCLASSES

	dnsbl [%]	*y_spam* [%]
Subclass 1	2.9	97.1
Subclass 2	98.3	1.7
Subclass 3	0.0	100.0

where μ is the mean and σ is the standard deviation. Both values are estimated on the base of values of a feature in the given subclass.

The distribution for the feature **maxpl** presented in 3(b) shows once again that the maximal length of packets is a good discriminant of subclasses. The feature gives three well–separate distributions.

Finally, an analysis of distribution of groups *dnsbl* and *y_spam* among calculated subclasses is presented. The analysis shows that the subclass 2, which has the lowest values of analysed features, consists of spam sent by IP addresses from DNS black lists. In this case, the analysis confirms division of original data between spam detected by blacklists and a spam analyser. However, spam in the *y_spam* group, which contains spam detected by SpamAssassin, is not a solid group and its members split between subclasses 2 and 3. Details are given in Table II.

C. Spam subclasses in classification task

In section II-A, the decision tree was presented as a source of discrimination rules. The tree, which separates spam subclasses, brings the description of the most significant spam discriminants.

The features are used to create a similar tree that separate spam from the rest of network traffic. The tree separates known spam subclasses (1–3) from flows without spam. This new class, labelled as 0, consists of records accepted by SpamAssassin as well as outgoing traffic. The class consists of about 4 thousand of records, which is significant less than number of spam records (54 thousand).

The created tree is given in Figure 4. The tree uses the **rslps** feature that describes the average length of package having the PUSH flag in response.

TABLE III
THE CLASSIFICATION MATRIX FOR THE CLASSIFICATION TREE THAT SEPARATES SPAM SUBCLASSES FROM THE REST OF TRAFFIC

	Class 0	Class 1	Class 2	Class 3
Predicted 0	2920	2	0	282
Predicted 1	242	4169	2	0
Predicted 2	86	1	38396	0
Predicted 3	924	41	0	6021

The classes 1 to 3 are covered with a false positive error. A part of valid traffic is recognised as spam. The error is not equally distributed among classes. The error is minimal for the class 2 where it is only 0.2 percent. For classes 1 and 3

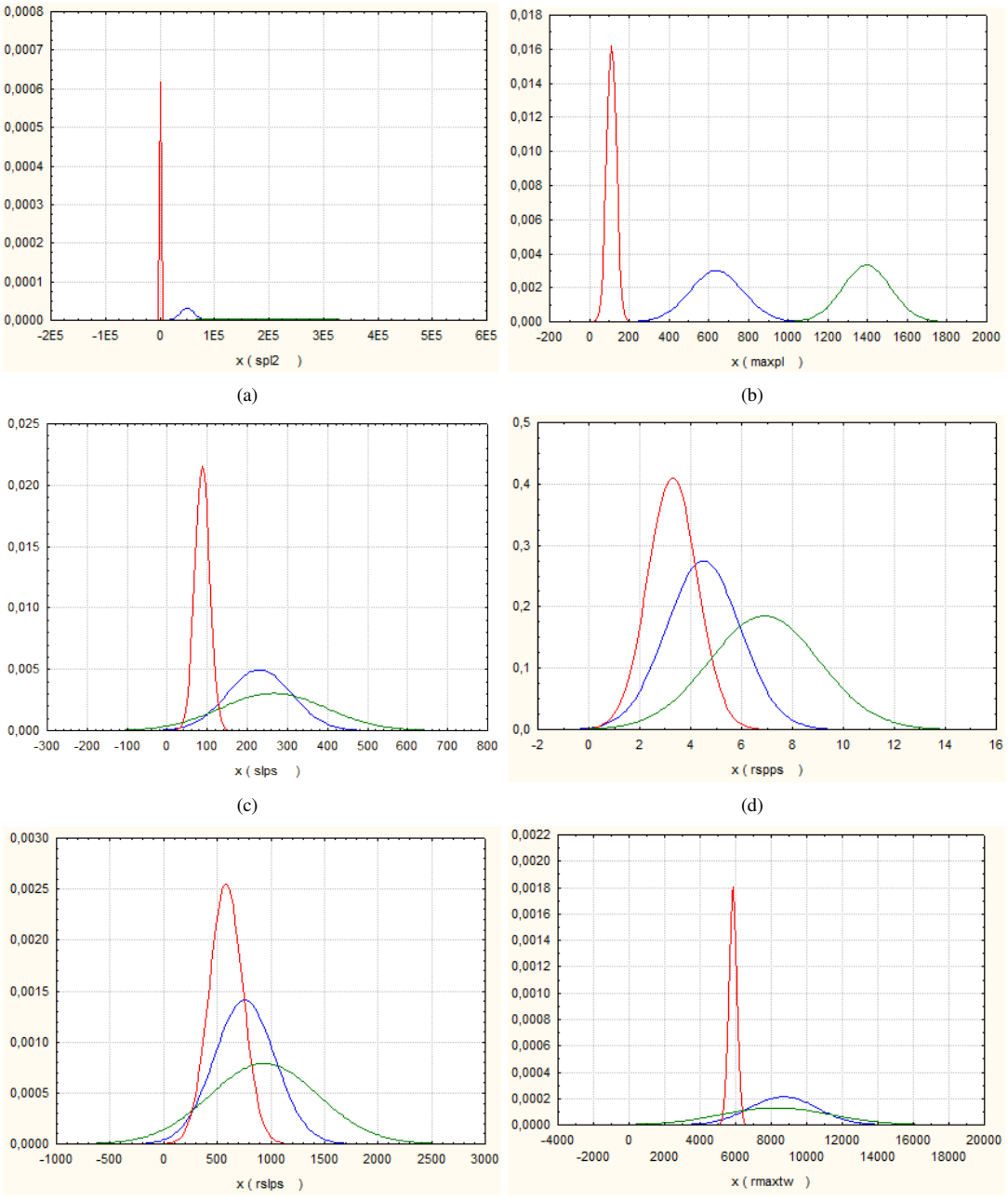


Fig. 3. Probability density calculated for the most characteristic spam features

it is 5.5 and 13.8 percent respectively. Details are given in Table III.

The total accuracy of the tree is 97 percent. When spam is treated as a single class, the accuracy is a slightly lower and the classification tree creates seven rules instead of three.

Similar results, for the same data set but described by different features, are presented in [12] (over 95 percent) where the Principal Component Analysis (PCA) was used to reduce number of features and in [8] (about 96 percent) where 64 features were used.

Although the accurate rate is higher than in cited works, a high error (about 30 percent) in classification of class 0 can be observed. This class is corresponding to the flows without spam. Obviously, this is the most important class since the classifier has to avoid misclassifying non–spam flows with spam flows. The high false positive ratio makes the binary classifier useless, but the classifier that recognises spam subclasses can be still used as a filter.

In the described example, there are three different cases. The first case concerns separation of valid traffic (class 0) and

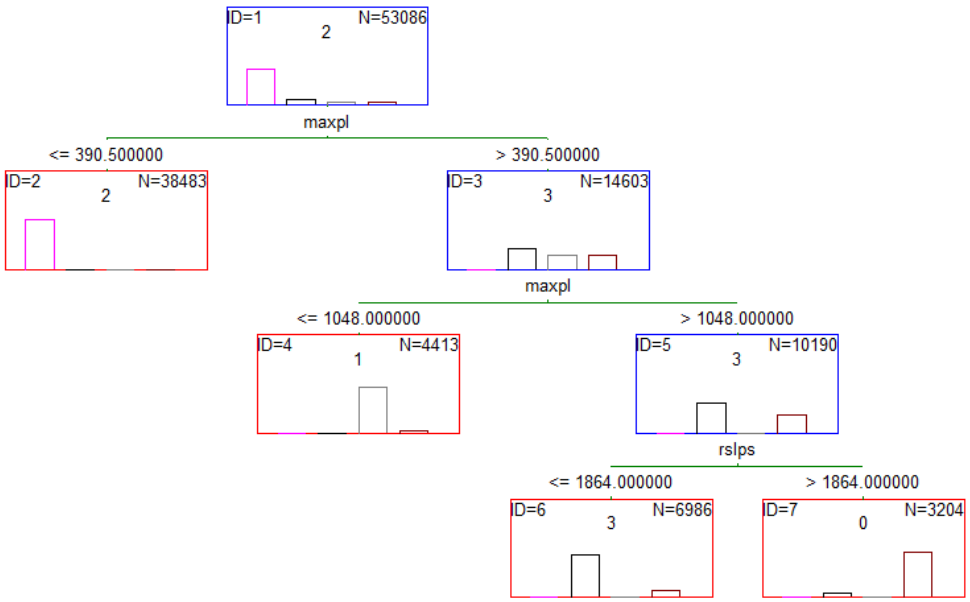


Fig. 4. The classification tree that separates spam subclasses from the rest of traffic

spam from blocked DNS addresses (class 2). In this case, the tree is an adequate tool to separate classes. The second case concerns the class 1. Nearly all members of this class are recognised correctly. However, some flows without spam are also classified as members. This class requires future analysis of false positive. The last case is focused on the class 3. Misclassifications are noticed between this class and the class 0. A more powerful classifier is needed to separate these classes.

III. Model verification

The results of analysis presented in Section II were verified with separate data. Significance of features determined on Žádník's data was validated by comparison with features calculated on Warsaw University of Technology mail server called Alpha.

A. Alpha

The described spam model was created on the base of NetFlow records collected by Žádník. For validation, a new set of NetFlow records was collected at Warsaw University of Technology. The set originates from the mail server called Alpha and consists of NetFlow records described by the same collection of features as Žádník's set. Data was collected through one working week. Over 42 thousand NetFlow records were collected. Among them 589 were labelled as spam.

This ratio between spam and non–spam data is completely different from the data discussed before. Here, the number of non–spam data is lower than the spam data. Therefore, the model was created on a dataset that is not very similar to the dataset collected from the Alpha.

Using the same method as for the Žádník's set (section II-C) a classifier based on a classification tree was created. The classifier separates spam from the rest of flows. The accuracy for both classifiers is very similar (about 97 percent). However, the disparate distribution of classes in the Alpha set results in much lower observed error (about 3 percent) in classification of class 0.

The created classification trees have different structures and splits are based on different features. Therefore, a set of features chose on the base of analysis of the Žádník's set should be verified.

B. Verification of features

Two decision trees that separate spam from the rest of traffic were created. The first one was trained on data from the Žádník's set, the second one on data collected from the Alpha server.

In the training process, the Gini coefficients were calculated. Therefore, the same approach that was used before to evaluate features describing spam can be applied once again to evaluate importance of features. The importance was calculated for all features proposed in [8].

An importance ranking on a 0–100 scale for each feature was created separately for each set. Figure 5 presents calculated significance of features.

The significant correlation between sets cannot be detected. There are several possible reasons for the difference between features selected by different classifiers.

The first reason may lie in the method. When a classification tree is created, the algorithm focuses on the most important features. Third–rate features may be different in various solutions.

The second reason lies in difference between the contents of discussed sets. As an example the feature **spas**, which describes the number of packages having ACK the flag set

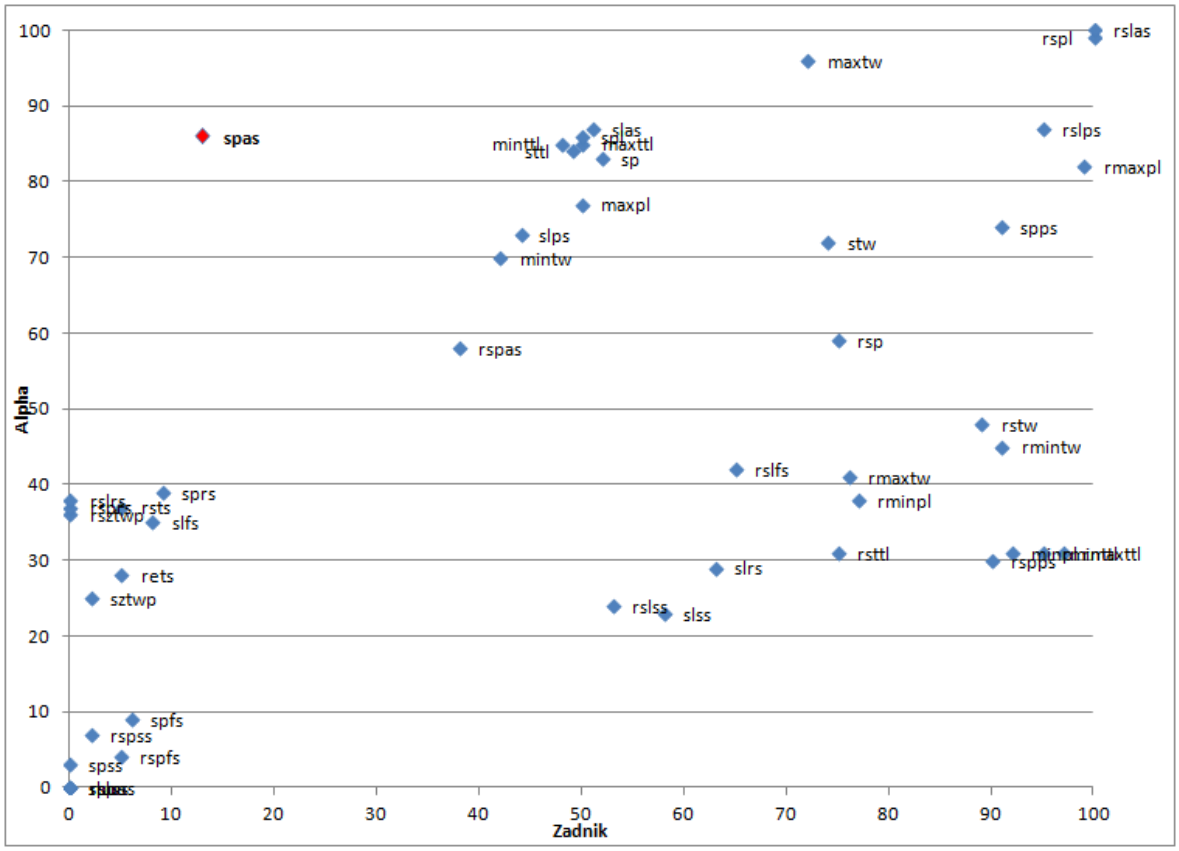


Fig. 5. Significance of features for both data sets: Žádník and Alpha

will be discussed. The feature is quite important for the classifier created on the Alpha set whereas of little importance for classifier created on the Žádník's set. The observation is highlighted in Figure 5.

The ACK flag is used to acknowledge the successful receipt of packets. Therefore, it should be turned on in almost all packages. The only exception is the first package sent in an exchange. In a transaction, the total number of packages with the ACK flag should be the total number of packages sent by a sender minus one. In practice, packages can be resend or missed and exceptions to the rule can be observed.

For discussed sets, the relation between the total number of packages (**sp**) and the total number of packages with the ACK flag (**spas**) in a flow was estimated using the method of least squares [13]. Results are presented in Figure 6.

For data collected on the Alpha server, (Figure 6(b)) the relation is similar to the ideal one:

$$\textbf{spas} = \ \textbf{sp} - 0.6. \tag{2}$$

Meanwhile, in the Žádník's set (Figure 6(a)), the relation is farther from the ideal:

$$\textbf{spas} = 0.9 \ \times \ \textbf{sp} - 9.4. \tag{3}$$

Such differences may influence features evaluation.

C. Universal features

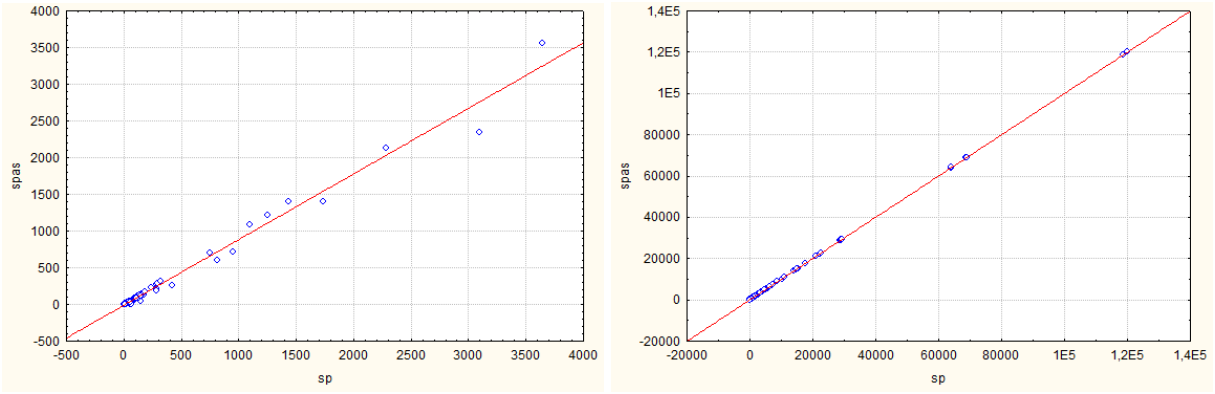
Despite the differences in importance of features evaluated on the base of data from different sources there is a small number of features such as **rslas** or **rspl** significant for both classifiers. However, their number is not enough to create a universal set of features. Therefore, the following method was used to create such set.

Each feature f calculated for a flow from source to designation has its equivalent rf calculated for a response. It is assumed, in the described method, that if a feature is added to the universal set then a response equivalent will be also added and vice versa.

The significance of features is evaluated on the base of Gini coefficients calculated during the decision tree creation. The evaluation functions g_A and g_Z are calculated from the Alpha and the Žádník's sets respectively. Each feature should be evaluated on the base of both evaluations. The presented assumptions result in the following evaluation function

$$g(f) = \frac{\max(g_A(f), g_A(rf)) + \max(g_Z(f), g_Z(rf))}{2}. \tag{4}$$

Because the range of evaluation functions g_A and g_Z is $[0, 100]$ it is reasonable to assume that significant features should at least achieve the level of 50 points.



(a) (b)

Fig. 6. The relation between the total number of packages and the total number of packages with the ACK flag calculated for the Žádník 6(a) and the Alpha 6(b) sets

TABLE IV
THE MOST SIGNIFICANT FEATURES DESCRIBING SPAM

Rank	Name	Description
100,0	slas	Average length of package having the ACK flag
99,5	spl	Average package length
91,0	slps	Average length of package having the PUSH flag
90,5	maxpl	Maximum package length
84,0	maxtw	Maximum TCP window size
82,5	spps	Count of packages having the PUSH flag
73,0	stw	Average TCP window size
68,0	mintw	Minimum TCP window size
67,5	maxttl	Maximum TTL
67,0	sp	Packages count
66,5	minttl	Minimum TTL
66,5	sttl	Average TTL

Among features, twelve have the evaluation result greater than 50 points. The most significant features are collected in Table IV. Information about direction of traffic is skipped. It is assumed that mentioned features should be calculated for both directions. That gives 24 features.

IV. Model application

The model was applied on data collected from a Broadband Remote Access Server (BRAS). On the base of learning data, an ensemble of classifiers was created and used to detect main sources of spam.

A. BRAS

Data was collected from a Broadband Remote Access Server (BRAS). Firstly, a probe that contains full headers was created. In eight seconds, almost 50 million PCAP packages were captured. From the packages payload was removed. The total size of remaining headers was 4.44 GB. Among all packages, 29.5 thousand were transferred by STMP protocol. That produced 407 NetFlow records. The ratio of collected records to the size of created file forced a different approach.

The second set was calculated in a reduced form. Because of huge size of collected PCAP headers NetFlows record were captured instead. Each record was described by 12 features including minimum, maximum, and average size of package, and binary information about flags occurrence. 176 thousand records were captured. The total size of collected NetFlow records was 18 MB.

Collected records was analysed by a spam detector to get out information about sources of spam.

B. Detector

The classifiers that detect spam were created on the base of two learning sets: Alpha and Žádník. Both sets were divided into learning and testing sets. The cardinal number of the training set was similar to the cardinal number of the learning set. Moreover, the proportion of spam to the rest data was similar in the learning and the testing sets, although the proportion of spam in the Žádník's set and the Alpha set are different.

It was mentioned before that a single tree should not be used as a detector. Instead, detector was projected as an ensemble of trees. Each tree was trained on the learning set and validated on the training set.

Each classifier from the ensemble recognises two classes: spam and background traffic. Before, the subclasses of spam were considered to determine the spam model. However, for the final user a determination of the spam subclass is not such important as a detection of spam. Therefore, the binary classification is performed.

The classes are labelled by 1 and -1 respectively. The classifier C_i that returns a decision y_i is described by the s_i coefficient that is the accuracy of discrimination between spam and the background. The final classification decision is the sign of a weighted sum given by the formula:

$$y = \operatorname{sgn}\sum_{i=1}^{n} \frac{s_i}{\sum_{j=1}^{n} s_j} y_i, \tag{5}$$

where n is the number of classifiers in the ensemble.

The detector can return 0 what means that the decision is uncertain.

C. Detection of spammers

The created detector was used to detect spam among flows collected from the BRAS. Over 60 trees trained on various subsets of the universal features set were used to create the detector defined by (5). In the result, 934 records from 176 thousand records were labelled as spam.

In the next step, sources of spam were localised. Records from the captured data came from 64088 unique IP addresses. Among them, 359 have sent at least one record labelled as spam. Most of them (211 addresses) sent just one record labelled as spam, but the record holder sent 221 spam records.

It is hard to assume that each sender from this set is a spammer. Therefore, a spammer was defined as a source of at least 10 spam records. This limitation results in seven main spam senders. Together they sent 46 percent of records labelled as spam. The main spammers can be easy blocked, which results in a significant reduction of spam.

V. Conclusions

In this work, the approach to detect main sources of spam in collected network traffic is presented.

Firstly, flow–level model of spam is created. The model describes spam subclasses and brings information about major features for a spam detection task. The presented model was verified on separate data. The verification resulted in a universal set of features.

The universal set consists of features that should be collected from a network in the form of NetFlow records. Among features are length of packages, information about window size, information about flags etc.

Selected features from the universal set were collected on Broadband Remote Access Server. Next, the detector, which was an ensemble of decision trees learned on various datasets, was created. The detector showed main source of spam among senders of collected flows. An elimination of detected spammers will reduce a number of spam by over 45 percent.

It should be noticed that the spam traffic properties will be changing over time and model will need to be retrained. The traffic properties can be also different in case of intensive spam attack. However, a regular collection of learning records from a network should resolve the first problem. Additionally, special learning sets that simulate intensive attacks can be used to improve model.

Gradually modifications of the model can be easy done by addition of new classifiers to the detector represented by equation (5) (although the total number of classifiers should be limited to avoid the drift learning problem).

Moreover, the detector based on decision trees can be implemented as a network probe. A software solution can be implemented as an nProbe [14] plug–in, but a hardware solution is also possible if the decision algorithm will be implemented on FPGA card.

References

[1] L. Limwiwatkul and A. Rungsawang, "Distributed denial of service detection using tcp/ip header and traffic measurement analysis," vol. 1, pp. 605 – 610 vol.1, oct. 2004.

[2] P. Kobiersky, J. Korenek, and L. Polcak, "Packet header analysis and field extraction for multigigabit networks," pp. 96 –101, april 2009.

[3] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 50–60, Jun. 2005. [Online]. Available: http://doi.acm.org/10.1145/1071690.1064220

[4] B. Claise, *Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information*, 2008.

[5] ——, *Cisco systems NetFlow services export version 9*, 2004.

[6] A. Moore, M. Crogan, A. W. Moore, Q. Mary, D. Zuev, D. Zuev, and M. L. Crogan, "Discriminators for use in flow-based classification," Tech. Rep., 2005.

[7] C. Leita and M. Dacier, "Sgnet: A worldwide deployable framework to support the analysis of malware threat models," pp. 99 –109, may 2008.

[8] M. Žádník and Z. Michlovský, "Is spam visible in flow-level statistics?" Tech. Rep., 2009.

[9] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979. [Online]. Available: http://dx.doi.org/10.2307/2346830

[10] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[11] G. Behera, "Privacy preserving c4.5 using gini index," in *Emerging Trends and Applications in Computer Science (NCETACS), 2011 2nd National Conference on*, march 2011, pp. 1 –4.

[12] M. Grzenda, "Towards the reduction of data used for the classification of network flows," in *Proceedings of the 7th international conference on Hybrid Artificial Intelligent Systems - Volume Part II*, ser. HAIS'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 68–77.

[13] D. Ruppert, S. J. Sheather, and M. P. Wand, "An effective bandwidth selector for local least squares regression (Corr: 96V91 p1380)," *Journal of the American Statistical Association*, vol. 90, pp. 1257–1270, 1995.

[14] L. Deri, "nprobe: an open source netflow probe for gigabit networks," in *In Proc. of Terena TNC 2003*, 2003.