Exploiting Transport-Level Characteristics of Spam

Robert Beverly
MIT CSAIL
rbeverly@mit.edu

Karen Sollins
MIT CSAIL
sollins@csail.mit.edu

Abstract

We present a novel spam detection technique that relies on neither content nor reputation analysis. This work investigates the discriminatory power of email *transport-layer* characteristics, i.e. the TCP packet stream. From a corpus of messages and corresponding packets, we extract per-email TCP features. While legitimate mail flows are well-behaved, we observe small congestion windows, frequent retransmissions, loss and large latencies in spam traffic. To learn and exploit these differences, we build "SpamFlow." Using machine learning feature selection, SpamFlow identifies the most selective flow properties, thereby adapting to different networks and users. In addition to greater than 90% classification accuracy, SpamFlow correctly identifies 78% of the false negatives from a popular content filter. By exploiting the need to source large quantities of spam on resource constrained hosts and networks, SpamFlow is not easily subvertible.

1 Introduction

By all estimates, unsolicited email (spam) is a pressing and continuing problem on the Internet. A consortium of service providers reports that across more than 500M monitored mailboxes, 75% of all received mail is spam, amounting to more than 390B spam messages over a single quarter (Messaging Anti-Abuse Working Group, 2007). Spam clogs mailboxes, slows servers and lowers productivity. Not only is spam annoying, it adversely affects the reliability and stability of the global email system (Afergan & Beverly, 2005).

Popular methods for mitigating spam include content analysis (Mason, 2002; Sahami et al., 1998), collaborative filtering (SpamCop, 2007; Prakash, 2007), reputation analysis (Spamhaus, 2007; SORBS, 2007), and authentication schemes (Allman et al., 2007; Wong & Schlitt, 2006). While effective, none of these methods offer a panacea; spam is an arms race where spammers quickly adapt to the latest prevention techniques.

We propose a fundamentally different approach to identifying spam that is based on two observations. First, spam's low penetration rate requires spammers to send extremely large volumes of mail to remain commercially viable. Second, spammers increasingly rely on zombie "botnets," (Cooke et al., 2005) large collections of compromised machines under common control, as unwitting participants in sourcing spam (IronPort, 2006). Botnet hosts are typically widely distributed with low, asymmetric bandwidth connections to the Internet. Combining these observations we make the following hypothesis: the network links and hosts which source spam are constrained. We ask whether *the transport-level characteristics of email flows provide sufficient statistical power to differentiate spam from legitimate mail (ham).*

In investigating this hypothesis, we gather a live data set of email messages and their corresponding TCP (Postel, 1981) packets. We extract and examine per-email flow characteristics in detail. Based on the statistical power of these flow features, we develop "SpamFlow," a spam classifier. In contrast to existing approaches, SpamFlow relies on neither content nor reputation analysis; Figure 1 shows this relation. Using machine learning feature selection, SpamFlow identifies the most selective flow properties, thereby allowing it to adapt to different users and network environments.

By examining email at the transport layer, we hope to exploit a fundamental weakness in sourcing spam, the requirement to send large quantities of mail on resource constrained links. As the volume of spam is unlikely to abate, SpamFlow represents a new defense against a significant source of unwanted mail. Our research thus makes the following primary contributions:

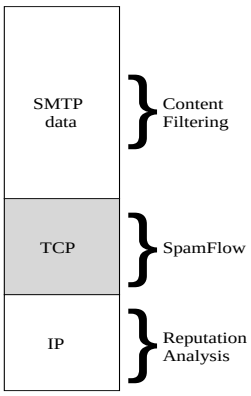


Figure 1: SpamFlow vs. existing schemes. SpamFlow learns and classifies on the TCP packet stream.

1. Identification of TCP flow features that exhibit significant probability differences between spam and ham.

2. SpamFlow, a classifier to learn and leverage these statistical differences for $> 90\%$ accuracy, precision and recall

3. Correct identification of 78% of the false negatives generated by SpamAssassin (Mason, 2002).

Consequently, we hope this paper serves to identify a new area of spam research and present a working system which sources of spam cannot easily evade.

2 Experimental Methodology

The intuition behind our scheme is simple. Because spammers must send large volumes of mail, they transmit mail continuously, asynchronously and in parallel. In addition, the sources of spam are frequently large compromised "botnets," which are resource constrained and typically connected to the Internet by links with asymmetric bandwidths, e.g. aDSL and cable modems.

Therefore the flows that comprise spam TCP traffic exhibit behavior consistent with traffic competing for link access. Thus, there is reason to believe that a spammer's traffic is more likely to exhibit TCP timeouts, retransmissions, resets and highly variable round trip time (RTT) estimates.

Is it reasonable to believe that a spammer's TCP/IP traffic characteristics are sufficiently different than traffic from Mail Transport Agents (MTAs) sending legitimate mail? To systematically understand large-scale behavior, we instrument an MTA to collect passive flow data for the email messages it receives.

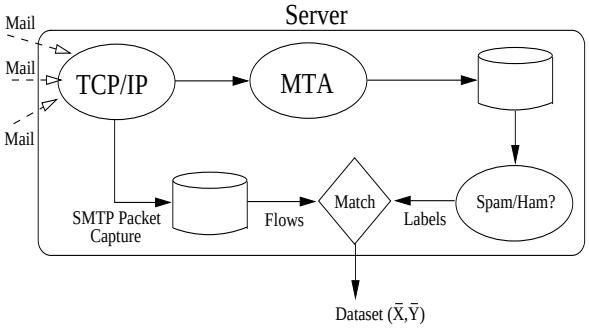


Figure 2: Data collection: incoming SMTP packets are captured and coalesced into flows. Each flow is matched with a binary spam/ham ground-truth label.

2.1 Data Collection

Figure 2 depicts our collection methodology. Our server has a dedicated, non-congested 100Mbps Ethernet connection to the local network which is in turn connected via multiple diverse Gigabit-speed links to the Internet. The server processes SMTP (Klensin, 2001) sessions and writes emails to disk. In the header of each email, the server adds the remote IP and TCP port number of the remote MTA that sent the mail. Simultaneously the server passively captures and timestamps all SMTP packets. Each email is then manually labeled as spam or ham to establish ground truth.

We coalesce the captured email packets into flows. Let our server's IP address be S. Define a flow $f_{IP:port}$ as all TCP packets $(IP{:}port) \rightarrow (S{:}25)$ and $(S{:}25) \rightarrow (IP{:}port)$[1]. Using the IP and TCP port number in the email headers, each email message is unambiguously matched with its corresponding SMTP flow. The port number is vital when receiving many, potentially simultaneous, emails from the same source.

Over the course of one week in January, 2008, we collect a total of 18,421 messages, 220 of which are legitimate while the remaining 18,201 are spam (98.8%). Of the ham messages, 39 are from unique mail domains.

2.2 Extracting Flow Features

We use the collected live data set to formalize a machine learning problem. Properties of each flow (f_i) provide the learning features $(\mathbf{x}_i)$. Currently we extract the features in Table 1. While our flows are undirected, particular features are directional, for instance received and sent packet counts, RSTs, FINs and retransmissions. Including directional features, we consider 13 features in total for each flow.

[1]Since our server's IP and SMTP port are fixed ($S{:}25$), these fields are not included in the flow tuple.

Table 1: Flow properties used as classification features

Feature	Description
Pkts	Packets
Rxmits	Retransmissions
RSTs	Packets with RST bit set
FINs	Packets with FIN bit set
Cwnd0	Times zero window advertised
CwndMin	Minimum window advertised
MaxIdle	Maximum idle time between packets
RTT	Initial round trip time estimate
JitterVar	Variance of interpacket delay

Each f_i corresponds to an email that is given a binary $y_i \in \{\pm 1\}$ label. Our data thus includes the input vector $\mathbf{x}_i \in \mathbb{R}^d, d = 13$ for flow f_i and label y_i. From these features, we wish to determine which provide the most discriminative power in picking out spam and how the number of training examples affects performance.

2.3 Transport Characteristics

In this subsection, we examine three of the flow properties in detail to illustrate the differences between spam and ham transport characteristics. Figure 3 compares the RTT, maximum idle time and FIN packet count between ham and spam in the entire data set. Here we define the RTT as the initial RTT estimate inferred by the three-way TCP handshake. Figure 3(a) shows the cumulative distribution of RTT times in our data. The difference between spam and ham is evident. While more than 20% of ham flows have an RTT less than or equal to 10ms, almost no spam flows have such a small initial RTT. The RTT of nearly all ham flows is 100ms or less. In contrast, 76% of spam flows have an RTT greater than 100ms.

A feature such as RTT can be used to provide a classifying discriminator, by taking the posterior probability of a message being a spam, given that the RTT of the message (rtt) is greater than r. Bayes' rule provides a convenient way to take the causal information and form a diagnosis:

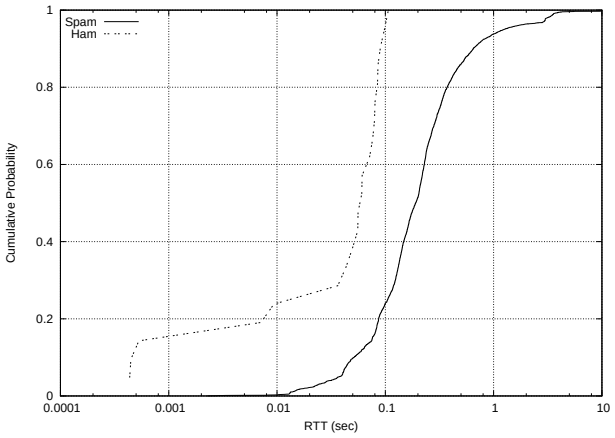
$$P(spam|rtt > r) = \frac{P(rtt > r|spam)P(spam)}{P(rtt > r)} \quad (1)$$

Figure 3(b) shows the conditional probability of a spam message across a continuous range of RTTs. We include the probability of a ham message in the figure as well; these probabilities sum to one, hence providing mirror images of each other. With an RTT less than 10ms, the probability is strongly biased toward being a ham message. In the range $[0.02, 0.1]$s, the probability estimate is relatively neutral without a strong bias toward either category. However, after 100ms, there is a strong tendency toward the message being spam. This conditional probability distribution corresponds exactly to the data in Figure 3(a).

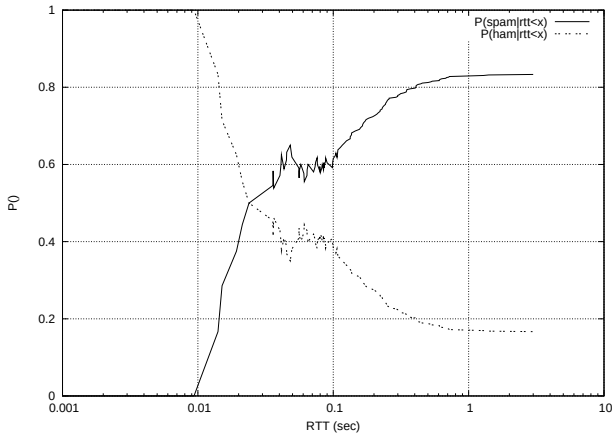
The differences in RTT raise several interesting points. For some classes of users, it is not unexpected that legitimate email originates from geographically nearby sources. Thus, it is prudent in many cases to take advantage of locality of interest. RTT may be less of a distinguishing characteristic though for users with frequent trans-continental conversations. However, approximately 50% of the spam messages have an RTT greater than 200ms, suggesting that the remote machines are quite remote, overloaded or reside on constrained links. Further, the $\sim$ 10% of flows with an RTT greater than one second cannot easily be explained by geographic distance and are more likely evidence of persistent congestion or end host behavior. We emphasize that RTT is just one potential feature. In instances where users receive legitimate email with large RTTs, the system may use a threshold strategy or simply lower the relative importance of RTT in favor of other flow features. Just as content filters are frequently customized per-user, the distinguishing flow characteristics can be unique to each user based on his or her receiving patterns.

As a second feature, consider maximum idle, the maximum time interval between two successive packets from the remote MTA. In some instances the maximum idle time directly reflects the initial RTT, but is often different. Figure 3(c) depicts the cumulative distribution of maximum idle times. Again, we see marked differences between the character of spam and ham flows. For instance, nearly 40% of spam flows have a maximum idle time greater than one second, events unlikely due to geographical locale. Figure 3(d) shows the conditional probability that the message is spam. After a maximum idle of 250ms, the probability tends strongly toward spam, as there are few legitimate messages with such a long idle time.

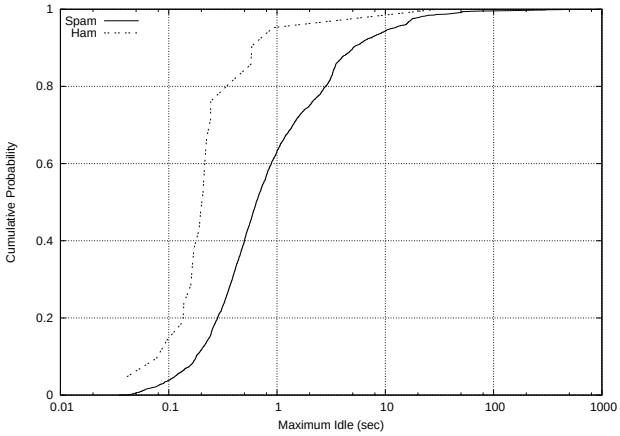
Finally, to emphasize that there are many potential features available in a flow (Table 1 enumerates features examined in this work), we examine TCP FIN segments. In a normal TCP session termination, each endpoint issues a finish (FIN) packet to reliably end the connection. Figure 3(e) shows that almost 45% of the spam email flows do not send a FIN compared to only 5% for ham. Finally, a small fraction of ham flows result in two FINs whereas only 0.7% of spam flows send more than one FIN. The resulting conditional probabilities are given in Figure 3(f).



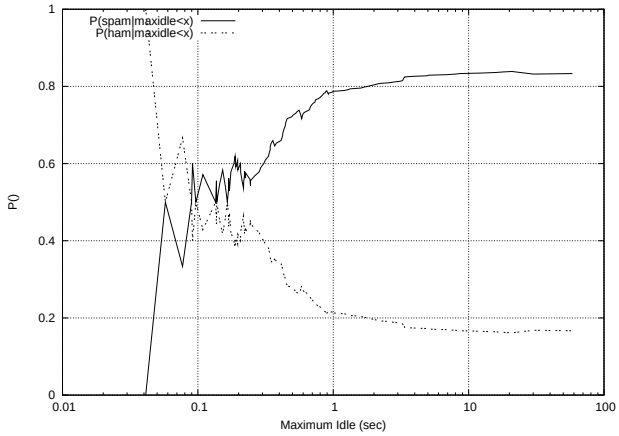
(a) RTT Cumulative Probability Distribution



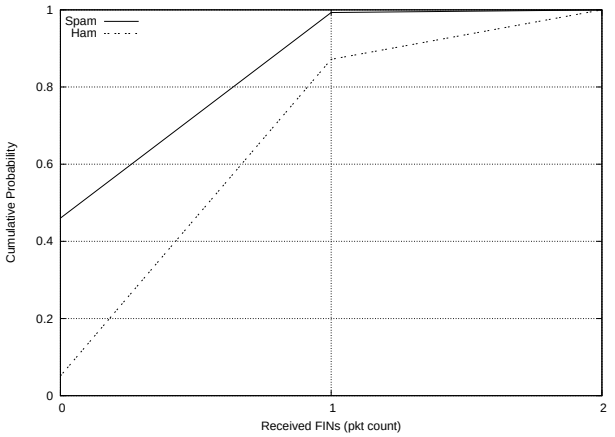
(b) RTT Conditional Probability



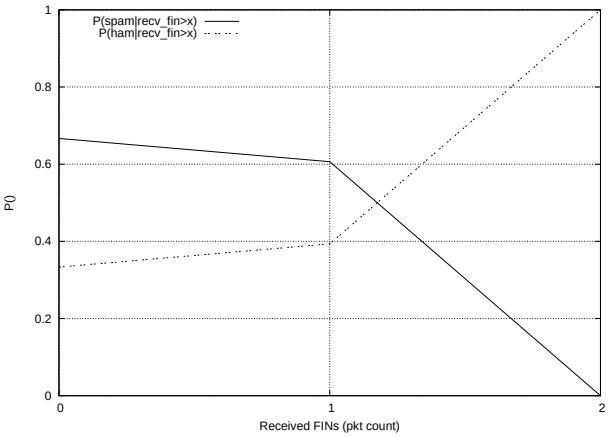
(c) Maximum Idle Time Cumulative Probability Distribution



(d) Maximum Idle Time Conditional Probability



(e) Received FIN Count Cumulative Probability Distribution



(f) Received FIN Count Conditional Probability

Figure 3: Comparing spam and ham probability distributions for RTT, idle time and received FIN count (left column). The resulting conditional probability distributions (right column) serve as a discriminator.

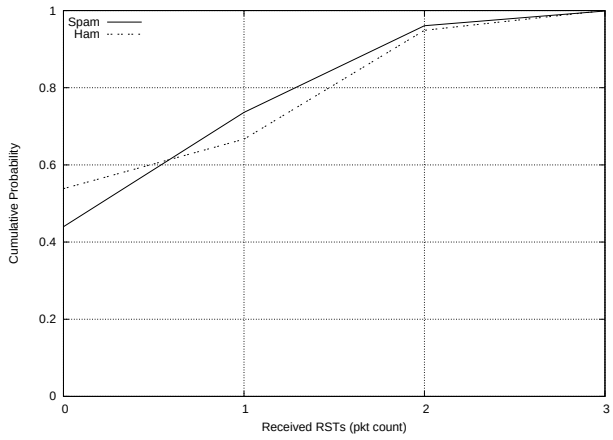


Figure 4: Non-features: distribution of received TCP RST count is similar for spam and ham. Surprisingly, this feature provides little discrimination.

2.4 Non-features

A strength of a statistical approach is in systematically identifying not only good features, but also poor features. Several flow properties we initially expected to be a strong indication of spam provide little differentiation. For example, one might expect ill-behaved flows to tear down the TCP connection using TCP resets (RSTs) rather than a graceful shutdown with FIN packets. However, as Figure 4 demonstrates, the distribution of received RSTs is very similar between spam and ham. Surprisingly, only 53% of ham flows contain no reset packets while 28% contain two RSTs.

Manual investigation of the data reveals that many MTAs, including Postfix and those of popular web mail services such as Google and Yahoo, send RST packets after sending the SMTP quit command. Detailed traces of this abortive close phenomenon are provided in (Beverly & Sollins, 2008).

In all, the preceding analysis provides evidence that spam and ham flows are sufficiently different to reliably distinguish between them. The important point of note is that we examine neither the content nor origins of incoming emails. Instead our determination of an email's legitimacy is based entirely upon the incoming flow's *transport characteristics.*

3 Results

Given our data set and problem formulation as described in §2, we turn to exploiting the differences in transport characteristics. In this Section we build and train a supervised classifier and study its performance.

3.1 Building a Classifier

In this study, we use only the unique ham mails so that our learning algorithm does not hone in on domain specific effects. For instance, if a majority of email arrives from Yahoo and Google MTAs, the primary features may reflect specific properties of flows from these servers. While nothing precludes learning on the basis of multiple mail flows from a single domain, we seek to understand the generality of SMTP flow characteristics. Our results will likely improve given additional training data from the same domain and MTAs.

As a result, our data set contains many more spam messages than legitimate messages. To prevent a large discrepancy in the complexion of training samples, we limit our data set to include only five times as many spam messages as valid messages. In each experiment, we select a random set of spam messages that is no more than five times larger than our ham corpus. Thus, the experiments include 39 valid emails and 195 randomly selected spam emails (234 total labeled messages and corresponding SMTP packets).

In each experiment, we take n data point pairs $(\mathbf{x}_i, y_i)$ from the feature extraction of §2. The n data points are then randomly separated into a training and test set. We horizontally concatenate the $\mathbf{y}$ labels and n x d feature matrix $\mathbf{X}$ to form $\mathbf{D} = [\mathbf{y}^T : \mathbf{X}]$. To ensure generality, we randomly permute rows of $\mathbf{D}$ for each experiment and run each experiment ten times. For a permuted $\mathbf{D}$, the training data consists of the first i rows of $\mathbf{D}$ while the test set is formed from the remaining $n - i$. In this way the training and test samples are different between experiments.

We use Support Vector Machines (SVMs) for classification (Vapnik, 1995) as maximum margin kernel methods with regularization perform well in practice on many tasks. However, we note that the general insight behind SpamFlow is independent of the exact learning algorithm.

3.2 Performance

Figure 5 shows the classification performance, measured in terms of accuracy, precision and recall as a function of the training size. We achieve approximately 90% accuracy using 60 training emails and more than 80% accuracy with only 20. This accuracy is relatively insensitive to the size of the data set, for instance if we include only twice as many spam as valid messages. However, the standard deviation is tighter as the number of training emails increases.

Note that accuracy may be misleadingly high as the true composition of our test set includes fives times

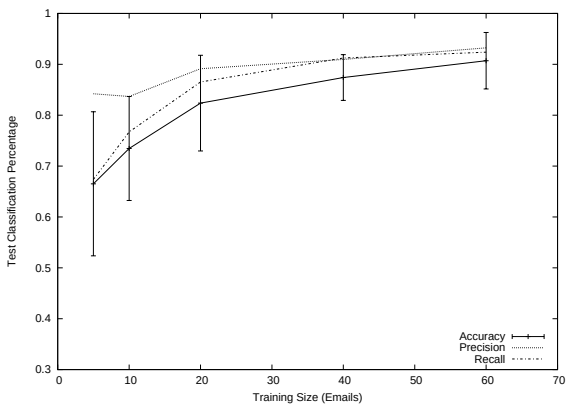


Figure 5: Classification accuracy, precision and recall vs. training size

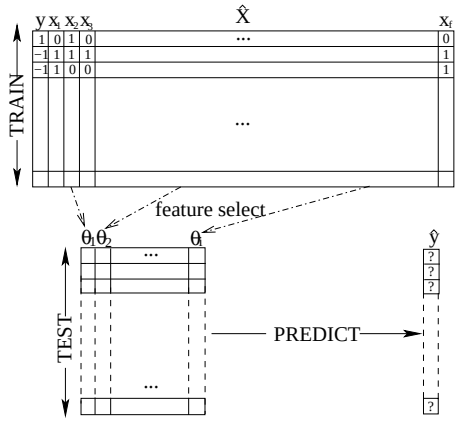


Figure 6: Forward fitting finds a set of features $|\boldsymbol{\theta}| < d$ that provide the least training error. These features are then used in test prediction.

as many spam messages as ham. A naïve classifier need only guess "spam" to achieve high accuracy. Thus, we also include recall, or the true positive rate and precision measures. Recall is the ratio of true positives to the number of actual positives, $recall = TP/(TP + FN)$ and is therefore a proxy for the number of false negatives. Precision is most important in this application where the majority of messages are spam. Precision is the ratio of true positives to all predicted positives, $precision = TP/(TP + FP)$, providing a metric of false positives. We see that at 40 training mails, the precision is more than 90%, corresponding to an average of two false positives per iteration.

Our results are from a fourth degree polynomial kernel without any SVM tuning or care in the input feature space. The current false positive rate is higher than is ideal for our application. With further effort, we can likely achieve higher performance. However, we envision SpamFlow as an additional metric in an overall decision tree in just the same way modern filters use multiple tests to form an overall spam decision.

3.3 Feature Selection

In order to optimize its performance to different users and network environments, SpamFlow determines which features provide the most discrimination power. To find these, we turn to feature selection methods (Yang & Pedersen, 1997). Greedy forward fitting (FF) requires computing a combinatorial number of possible feature combinations. However forward fitting effectively eliminates features that themselves are closely dependent. Often two features individually provide significant power, but the second feature provides little additional classification power. For example, the RTT and maximum idle time may be highly correlated. Forward fitting will continually seek the next best performing feature without this potential dependence.

Forward fitting feature selection simply finds, in succession, the next single feature j that minimizes an error function $V(\cdot)$. Therefore, training error decreases monotonically with the number of features. Figure 6 provides the basic intuition behind feature selection.

Feature selection proceeds in rounds. In round i let $\mathbf{S}^j$ be a d x i binary selection matrix with $s_{j,i} = 1$ and $s_{k\neq j,i} = 0$. The $i-1$ columns of $\mathbf{S}^j$ are set in previous rounds. Recall that the data $\mathbf{X}$ is an n x d matrix containing n emails with d features each. Let:

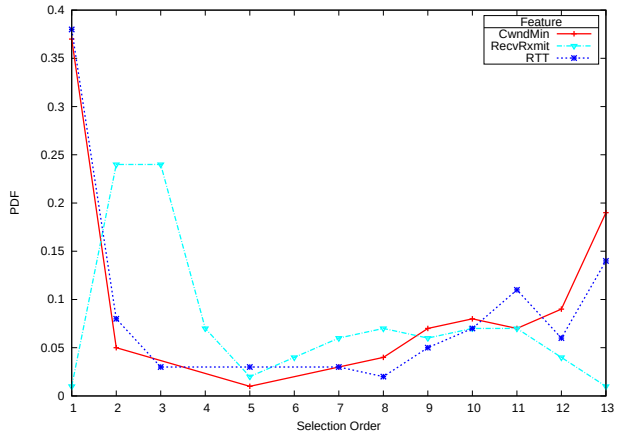
$$\mathbf{Z}^j = \mathbf{X}\mathbf{S}^j \tag{2}$$

Thus, $\mathbf{S}^j$ selects feature j in round i. Let $D = \{1 \ldots d\}$ indicate the set of all possible features. We denote θ^i as the set of best features in round i. Then, for a prediction function $f(\cdot)$ and error function $V(\cdot)$, find:

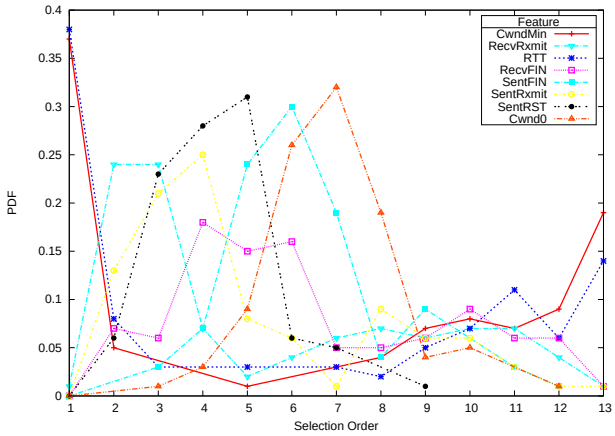
$$\underset{j \in F - \theta^{i-1}}{\operatorname{argmax}} \; V(f(\mathbf{Z}^j), \mathbf{Y}) \tag{3}$$

The feature that best minimizes the error in round i is j, so we update the set of best features: $\theta^i = \theta^{i-1} + j$. Training error is typically an effective proxy for test error. We use SVM training accuracy as the error function although forward fitting can be used with any model and error function.

Figure 7 shows the cumulative probability distributions of the selection order for each feature. We split the results into two plots only to improve readability. Figure 7(a) illustrates that both RTT and CwndMin are the most likely features to be selected first, each with approximately 40% probability. Maxidle has



(a) Primary features: minimum congestion window and initial RTT are frequently the best feature; received retransmits are a strong second feature.



(b) Secondary features: have probability centered in the middle of the selection order indicating flow properties that distinguish ham and spam well.

Figure 7: Feature selection order probability distributions demonstrate the relative discriminatory strength of different flow properties.

around a 10% chance of being the first selected feature and the other features comprise the remaining 10%. In other words, if the learner were given the choice of one and only one feature with which to classify, the learner would choose RTT or CwndMin. RecvRxmit and SentRxmit are typically not the first or second feature, but frequently serve as the third and fourth best features.

Figure 7(b) gives the secondary features, those that are more likely to be chosen fifth or later in the order. These features include the RecvFIN, SentFIN, Cwnd0 and JitterVar.

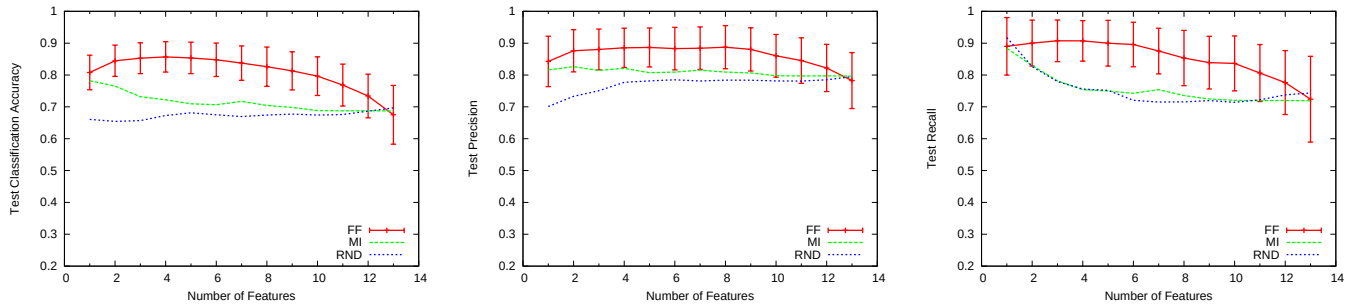
To leverage the results of feature selection, we measure the prediction dependence on the number of best features. Figure 8 gives the results of performing forward fitting, mutual information and random features in each round to select a given number of best features. We include random features to provide a useful baseline. As expected, the random features perform the worst, yet still yield 60-70% accuracy. Forward fitting achieves much higher accuracy, precision and recall, but suffers from over-fitting as the number of features is increased beyond five.

4 Related Work

Current best practices for defending against spam are multi-pronged with four main techniques: content filters, collaborative filtering, reputation systems and authentication schemes. The most successful attempts thus far to combat spam have relied on fundamental weaknesses in spam messages or their senders. We review these systems as well as previous network and traffic characterization studies.

Content Filtering: A wealth of content analysis systems are used to great effect today in filtering spam. Learning methods have been effectively applied to building classifiers that determine discriminatory word features (Sahami et al., 1998). Such content analyzers exploit the fact that a spam message contains statistically different words from a user's normal mail. Even innocuous looking commercial spam, intended to subvert content filters, typically includes a link to an advertised service – thereby providing a basis for differentiation. A popular open source solution is SpamAssassin (Mason, 2002), although there are many competing commercial alternatives. Our system, SpamFlow, does not perform any content analysis on the messages themselves. By providing an alternative classification mechanism, SpamFlow helps address blocking innocuous junk mail, for instance that used to "detrain" Bayesian filters (Vascellaro, 2006).

Collaborative Filtering: Spam is typically sent to many users thereby providing a signature. By aggregating the collective spam of a distributed set of users, collaborative filtering (Prakash, 2007; SpamCop, 2007) aims to prevent previously marked spam messages from being accepted. For example, popular web mail clients can easily provide collaborative filtering as their servers are under common administrative control and can leverage spam marked by one user to block spam to other users. Unfortunately, not all mail installations are large enough to take advan-



(a) Prediction Accuracy (b) Prediction Precision (c) Prediction Recall

Figure 8: Prediction performance relative to the number of features for Forward Fitting (FF), Mutual Information (MI) and Random (RND) feature selection with an SVM model. The points represent average performance across nodes in our data set, while the error bars show the standard deviation.

tage of collaborative filtering and are unwilling to rely on vulnerable centralized repositories. Further, spammers can trivially make each spam unique in an effort to avoid collaborative techniques.

Reputation Systems: Reputation systems attempt to aggregate historical knowledge over specific MTA IP addresses or mail domains. For instance, a large number of messages to unknown recipients might be recognized as a dictionary attack. MTAs that continually send undeliverable mail are given a low reputation. Often, spam honeypots are used in conjunction with reputation systems to gather additional data on spam origination. MTAs which have previously sent spam are likely to continue sending spam. Real-time databases (Spamhaus, 2007; SORBS, 2007; Secure Computing, 2007) of these offending MTAs and IP addresses provide blacklists which MTAs can query before accepting mail. However, Ramachandran's analysis of DNS blacklists (Ramachandran et al., 2007) shows that as much as 35% of spam is sent from IP addresses not listed in common blacklists. Their work brings to light an important point about the dynamism of IP addresses in relation to spam. Not only are the IP addresses of botnets changing as hosts acquire new addresses, spammers are rapidly changing addresses in order to evade blacklist reputation schemes. SpamFlow, however, has no dependence on IP addresses making it particularly attractive in defending against botnet spam.

Authentication Schemes: Authentication schemes attempt to verify the sender or sender's domain to prevent spoofing-based attacks. Sender Policy Framework (Wong & Schlitt, 2006) limits IP addresses to sourcing mail only for authorized domains. Domain keys (Allman et al., 2007) uses public keys to associate each email with an entity.

Characterization Studies: Casado et al. perform passive flow analysis on approximately 25,000 spam messages to determine bottleneck bandwidths (Casado et al., 2005). Their study finds significant modes at modem, Ethernet and OC-12 speeds, suggesting that spammers employ both farms of low-speed as well as high speed servers. In contrast, we perform a detailed passive flow analysis in order to find relevant features for forming classification decisions.

Brodsky's trinity system identifies botnets by counting email volumes, thereby identifying spam without content analysis. Similarly, the spamHINTS project (Clayton, 2006) leverages the sending patterns of spammers to identify the sources of spam. In addition to analyzing server logs, spamHINTS proposes to examine sampled flow data from a network exchange point to obtain a large cross section of email traffic patterns and volumes. For instance, hosts that source email continually or have particular patterns can be identified through a set of heuristics. In contrast, our work analyzes the individual packets of SMTP transactions to obtain much more detailed flow information, e.g. congestion windows and round trip times. Further, SpamFlow relies on machine learning techniques rather than heuristics to build a classification system.

Our work is in a similar spirit to (Ramachandran & Feamster, 2006) which attempts to characterize the network properties of spammers, for instance the IP blocks to which they belong. Instead, by taking a step down the protocol stack and examining the transport level properties of spam, we hope to take advantage of previously unexploited information.

5 Conclusions and Future Work

Our results are promising, demonstrating that even rough metrics of a flow's character can aid in differentiating incoming emails. By providing a method that does not rely on either content or reputation analysis, SpamFlow is a potentially useful tool in mitigating spam. Whereas reputation systems are vulnerable to IP address dynamics, SpamFlow has no reliance on addresses. While content analysis is easy to game, SpamFlow attempts to exploit the fundamental character of spam traffic. We plan to gather a significantly larger data set that includes more valid messages and additional features.

Can spammers adapt and avoid a transport-based classification scheme? By utilizing one of the fundamental weaknesses of spammers, their need to send large volumes of spam on bandwidth constrained links, we believe SpamFlow is difficult for spammers to evade. A spammer might send spam at a lower rate or upgrade their infrastructure in order to remove any congestion effects from appearing in their flows. However, either strategy is likely to impose monetary and time costs on the spammer.

The initial RTT is the strongest indication of spam for our data set. A spammer might attempt to artificially lower the inferred RTT by optimistically acknowledging packets that have not yet been received. However, an adversary cannot reliably know the remote host's initial sequence number for the TCP connection and therefore cannot easily fake the initial RTT. Such attempts to hack TCP to disguise the effects we observe are likely to expose other features, for instance retransmits and duplicate packets.

While RTT is the strongest discriminator on our data, other mail users may have different email interactions with geographically dispersed MTAs. Further, the observed spam RTT may vary for MTAs in countries other than ours. However, such differences demonstrate the strength of a statistical approach. Just as content based filtering is personalized for individual users, the particular features for transport based filtering can be tailored to the end recipients.

Because SpamFlow performs neither content nor reputation analysis, its functionality could be pushed deeper into the core of the network without compromising privacy concerns. SpamFlow is unique in this regard. In addition, with a wider cross-sectional view the performance of SpamFlow would likely improve.

Utilizing available flow information may aid not only in preventing spam, but also other types of attacks that originate from botnets and compromised machines. For instance, denial of service attacks similarly rely on sending large quantities of data over constrained links. We wish to gather data to better understand the broader applicability of our approach.

Our hope is that this work serves as a step forward in providing a means to combat spam and impose a greater cost on parties sourcing spam.

References

Afergan, M., & Beverly, R. (2005). The state of the email address. *ACM SIGCOMM Computer Communications Review, Measuring the Internet's Vital Statistics*, *35*, 29–36.

Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., & Thomas, M. (2007). DomainKeys Identified Mail (DKIM) Signatures. RFC 4871 (Proposed Standard).

Beverly, R., & Sollins, K. (2008). *Exploiting transport-level characteristics of spam* (Technical Report MIT-CSAIL-TR-2008-008). MIT.

Casado, M., Garfinkel, T., Cui, W., Paxson, V., & Savage, S. (2005). Opportunistic measurement: Extracting insight from spurious traffic. *Proceedings of the ACM HotNets Workshop*.

Clayton, R. (2006). Using early results from the spamHINTS project to estimate an ISP abuse team's task. *Third Conference on Email and Anti-Spam*.

Cooke, E., Jahanian, F., & McPherson, D. (2005). The zombie roundup: Understanding, detecting, and disrupting botnets. *Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet (SRUTI) Workshop*.

IronPort (2006). Spammers continue innovation: Ironport study shows image-based spam, hit & run, and increased volumes latest threat to your inbox. `http://www.ironport.com/company/ironport_pr_2006-06-28.html`.

Klensin, J. (2001). Simple Mail Transfer Protocol. RFC 2821 (Proposed Standard).

Mason, J. (2002). Filtering spam with spamassassin. *Proceedings of SAGE-IE*.

Messaging Anti-Abuse Working Group (2007). Email metrics report. `http://www.maawg.org/about/EMR`.

Postel, J. (1981). Transmission Control Protocol. RFC 793 (Standard). Updated by RFC 3168.

Prakash, V. V. (2007). Vipul's razor. `http://razor.sourceforge.net/`.

Ramachandran, A., & Feamster, N. (2006). Understanding the network-level behavior of spammers. *Proceedings of ACM SIGCOMM*.

Ramachandran, A., Feamster, N., & Vempala, S. (2007). Filtering spam with behavioral blacklisting. *Proceedings of ACM Conference on Computer and Communications Security*.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A bayesian approach to filtering junk e-mail. *AAAI Workshop on Learning for Text Categorization*.

Secure Computing (2007). Ironmail. `http://www.securecomputing.com`.

SORBS (2007). Spam and open-relay blocking system (SORBS). `http://www.sorbs.net`.

SpamCop (2007). Spamcop. `http://www.spamcop.net`.

Spamhaus (2007). `http://www.spamhaus.org/sbl/`.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer Verlag.

Vascellaro, J. (2006). Empty spam feasts on in-boxes. `http://online.wsj.com/article_email/SB115448102123224125-lMyQjAxMDE2NTA0MjQwODIxWj.html`.

Wong, M., & Schlitt, W. (2006). Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. RFC 4408 (Experimental).

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *Proceedings of ICML-97, 14th International Conference on Machine Learning* (pp. 412–420).