Controlling False Alarm/Discovery Rates in Online Internet Traffic Flow Classification

Daniel Nechay, Yvan Pointurier and Mark Coates

McGill University
Department of Electrical and Computer Engineering
Montreal, Quebec, Canada

April 22, 2009

Outline

Introduction

What is Internet traffic classification?

- Associate a user-defined class to a traffic flow
- Class can be broad (P2P) or application specific (BitTorrent, Kazaa, etc.)

Why do we need Internet traffic classification?

There are a variety of applications where Internet traffic classification is needed:

- To help provide QoS guarantees or enforce Service Level Agreements (SLA)
- Prioritize or limit/block traffic
- Network provisioning
- Network security

Current Traffic Classification Methods

Port-Based

- Simplest method
- Not reliable

Deep-Packet Inspection

- Examine the payload of the packets to look for application-specific signatures
- Privacy and legal concerns

Shallow-Packet Inspection

- Derives statistics from the packet headers and uses this information to classify the flow
- Non-invasive and still works on encrypted packets

Our Contribution

Contributions

1. Provide a performance guarantee on the false alarm or false discovery rates
2. Novel methodology: converted binary classifier into a multi-class classifier
3. Online classification

Problem Formulation

Definitions

- X - the d-dimensional random variable corresponding to the flow features
- Each flow is associated an output Y
- $Z = Y \in \{1 \ldots, c + 1\}$ the class of the flow

Problem Statement 1

Goal of Neyman-Pearson classification

To minimize the overall misclassification rate while adhering to certain false alarm rate (FAR) constraints

False Alarm Rate for class i

Expected fraction of the flows that do not belong to traffic class i that are incorrectly classified as belonging to i.

Problem Statement 2

Goal of Learning to Satisfy (LSAT) framework

To provide false discovery rates (FDR) guarantees while minimizing the overall misclassification rate

False Discovery Rate for class i

Expected fraction of incorrectly classified flows among all traffic flows classified as class i.

Background

Support Vector Machines (SVM)

SVMs consist of two steps:

1. Transform the input features x_i via a mapping $\Phi : \mathbb{R}^d \rightarrow \mathcal{H}$ where $\mathcal{H}$ is a high-dimensional Hilbert space
2. Construct a hyperplane (the decision boundary) in $\mathcal{H}$ according to the max-margin principle

Cost-Sensitive Classification

- Regular SVM treats all misclassifications equally
- Cost-Sensitive classification (our case 2ν-SVM) treats the misclassification of each class differently
- Have two parameters ν_- & ν_+ to control the misclassification for the different classes

What is LSAT?

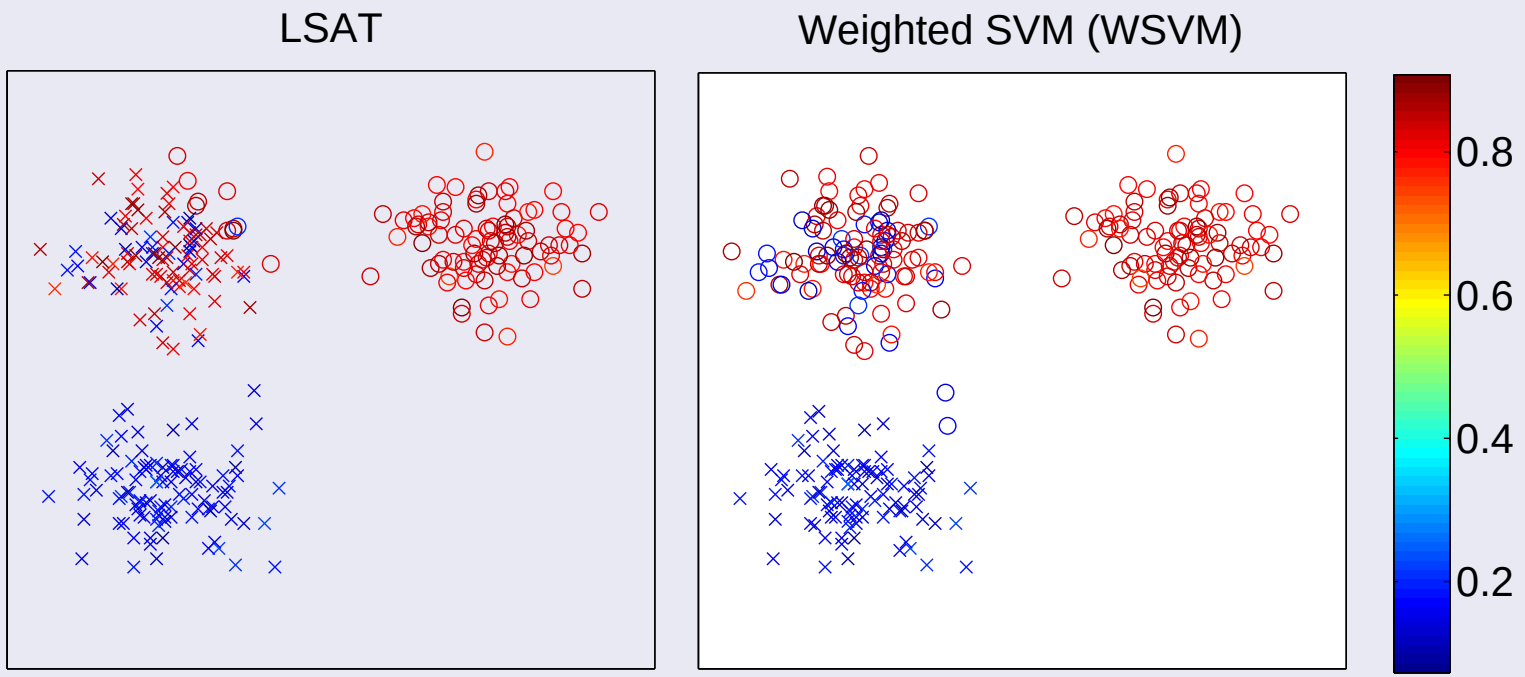
Goal

The goal is to learn a set in the input (feature) space that simultaneously satisfies multiple output constraints. The LSAT framework is distinguished by:

1. multiple performance criteria must be satisfied
2. output behaviour is assessed only on the solution set.

LSAT example

Comparison of LSAT to WSVM



Reference

F. Thouin, M. J. Coates, B. Erikkson, R. Nowak, and C. Scott, Learning to Satisfy, in Proc. Int. Conf. Acoustics, Speech, and Signal Proc. (ICASSP), Las Vegas, NV, USA, Apr. 2008.

Traffic Classification

How to classify c classes?

- Use a chain of c binary classifiers
- Each binary classifier responsible for a particular class
- Ordering is important
- Classified as unknown if there are no mappings to a class

How to determine the best classifier?

- Find the best parameters ν_+, ν_- and σ for the 2ν-SVM
- Introduce cost functions to rank the classifiers

Cost Functions

Traffic classification with FAR constraints

For every classifier, the following risk function is used:

$$R(f) = \sum_{s(i)} \frac{1}{\alpha_{s(i)}} \max(P_F(s(i)) - \alpha_{s(i)}, 0) + P_M(s(i))$$

- s(i): class i
- $\alpha_{s(i)}$: FAR constraint for class i
- $P_F(s(i))$: FAR for class i
- $P_M(s(i))$: Misclassification rate for class i

Traffic classification with FDR constraints

Ensure that it satisfies the constraints set — then choose the classifier that minimizes the misclassification rate

Input Data

Data

- Collected a 24 hour trace using tcpdump in April and split the trace by hour
- Only considered TCP flows for inputs
- tcptrace was able to collect 142 statistics for every flow
- Feature selection reduced the feature space to 5 features
- Classify after the first six packets of a flow
- Bro was used to provide a ground truth

Application Breakdown

Application Breakdown after 6 packets of a flow

Table: Application breakdown for flows $>$ 6 packets

Application	Flows		Size	
	Number	Percentage	GB	Percentage
HTTP	315375	78.3%	4.1	74.6%
HTTPS	20736	5.2%	0.29	5.4%
MSN	3364	0.8%	0.04	0.7%
POP3	1311	0.3%	0.01	0.2%
OTHER	61870	15.4%	1.05	19.1%

Simulation environment

Statistics Used

- total number of bytes sent (C→S)
- number of packets with the FIN field set (C→S)
- the window scaling factor used (C→S)
- total number of bytes truncated in the packet capture (C→S)
- total number of packets truncated in the packet capture (S→C)

FAR-constrained classifier

Classifiers

Three classifiers compared:

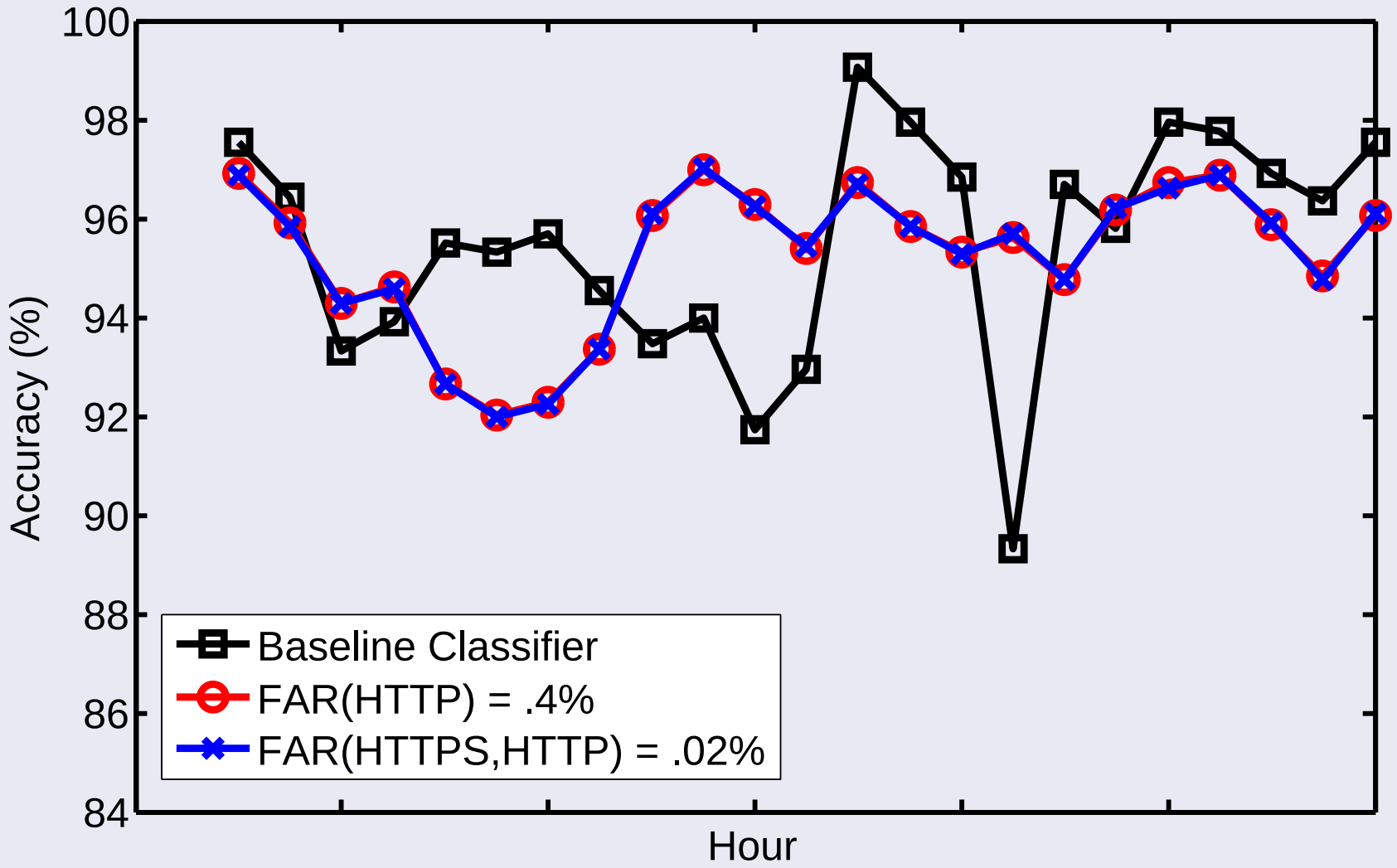
- Baseline Classifier - Multi-class SVM
- FAR-constrained classifier with $\alpha\{HTTP\} = 0.4\%$
- FAR-constrained classifier with $\alpha\{HTTPS, HTTP\} = 0.05\%$

Hour 1 Results

- Trained on 1000 randomly chosen points in hour 1 & validated on the rest of the hour
- Baseline classifier has $\alpha\{HTTP\} = 3.7\%$ and $\alpha\{HTTPS, HTTP\} = 0.07\%$
- Classwise FAR-constrained classifier has $\alpha\{HTTP\} = 0.3\%$ while the pairwise FAR-constrained classifier has $\alpha\{HTTPS, HTTP\} = 0.02\%$

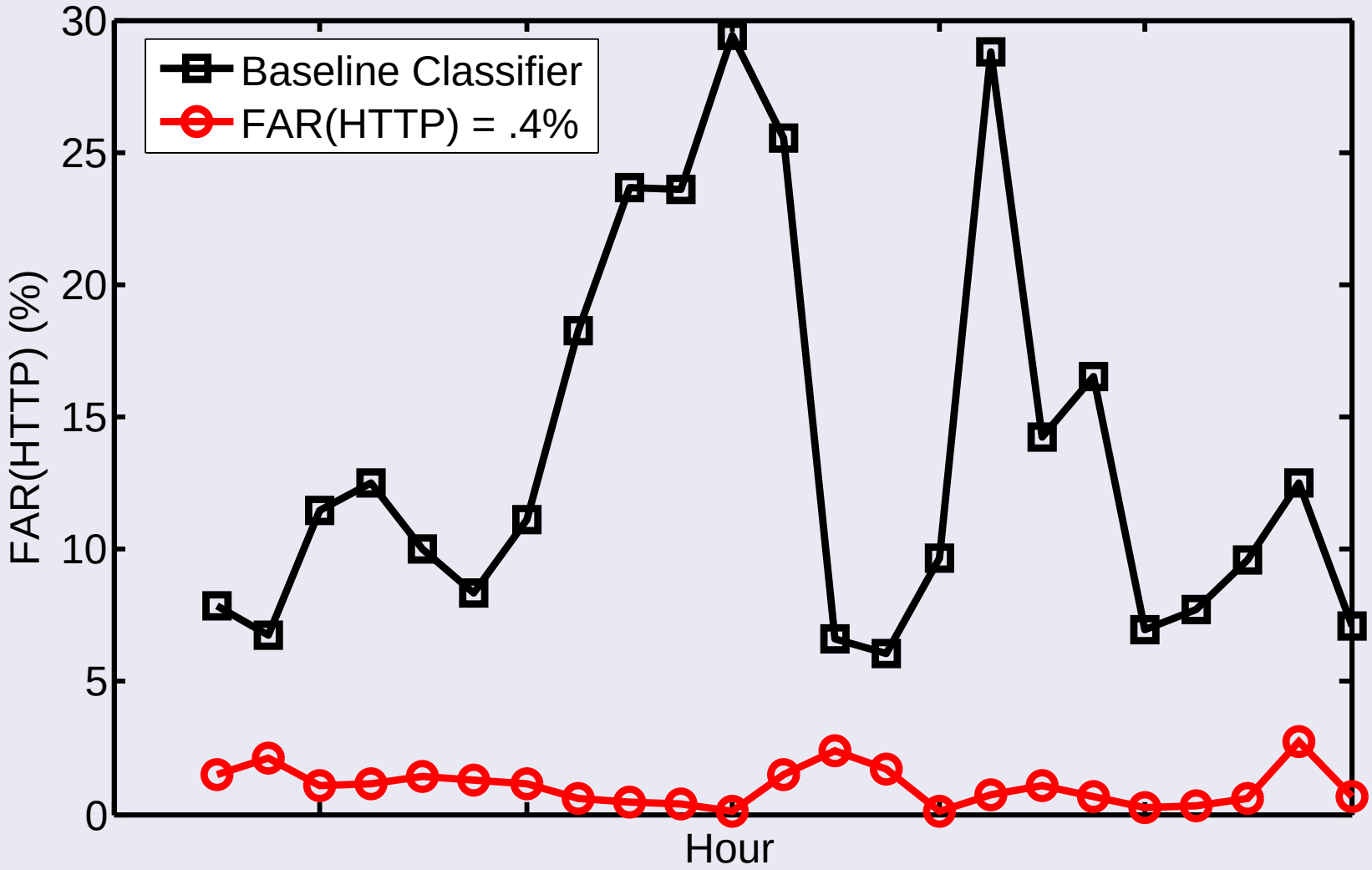
FAR-constrained classifier

Overall Accuracy for Hours 2 - 24



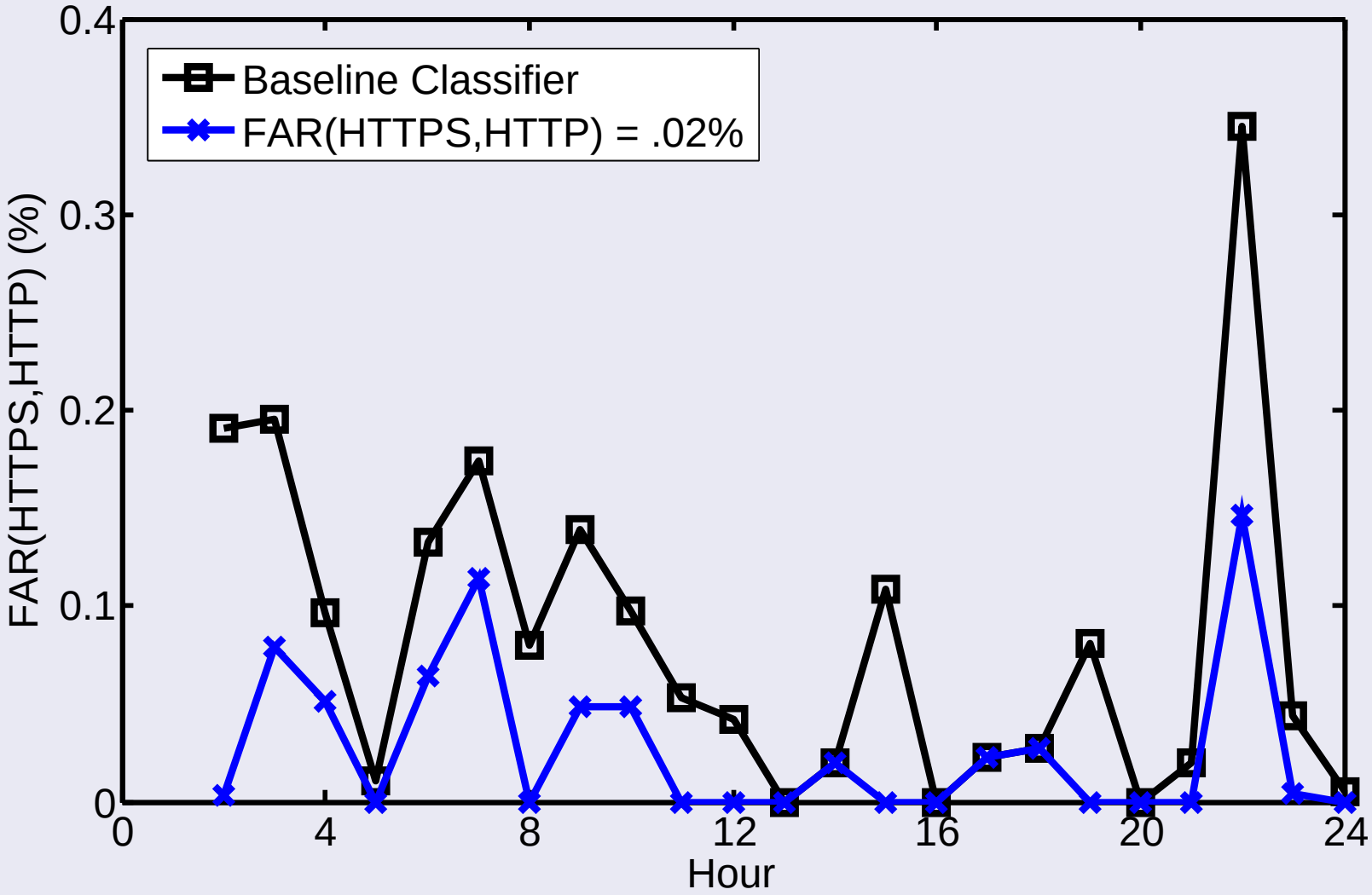
FAR-constrained classifier

FAR(HTTP) for Hours 2 - 24



FAR-constrained classifier

FAR(HTTPS,HTTP) for Hours 2 - 24



FDR-constrained classifier

Classifiers

Three classifiers compared:

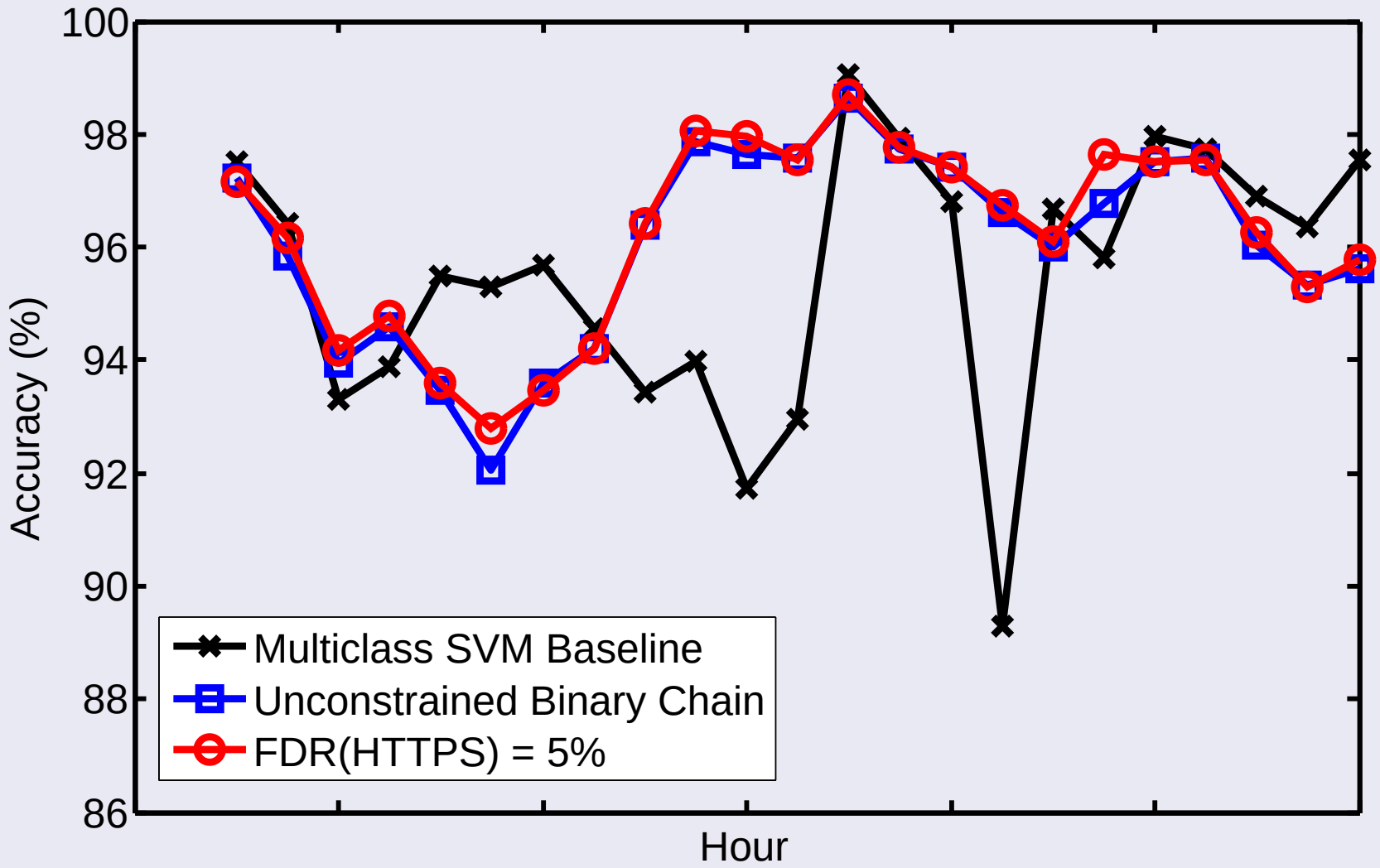
- Baseline Classifier - Multiclass SVM
- Unconstrained binary-chained classifier
- FDR-constrained classifier with $\beta\{HTTPS\} = 5\%$

Hour 1 Results

- Trained on 1000 randomly chosen points in hour 1
- Unconstrained binary-chained classifier has $\beta\{HTTPS\} = 7.0\%$ while the FDR-constrained classifier has $\beta\{HTTPS\} = 4.2\%$

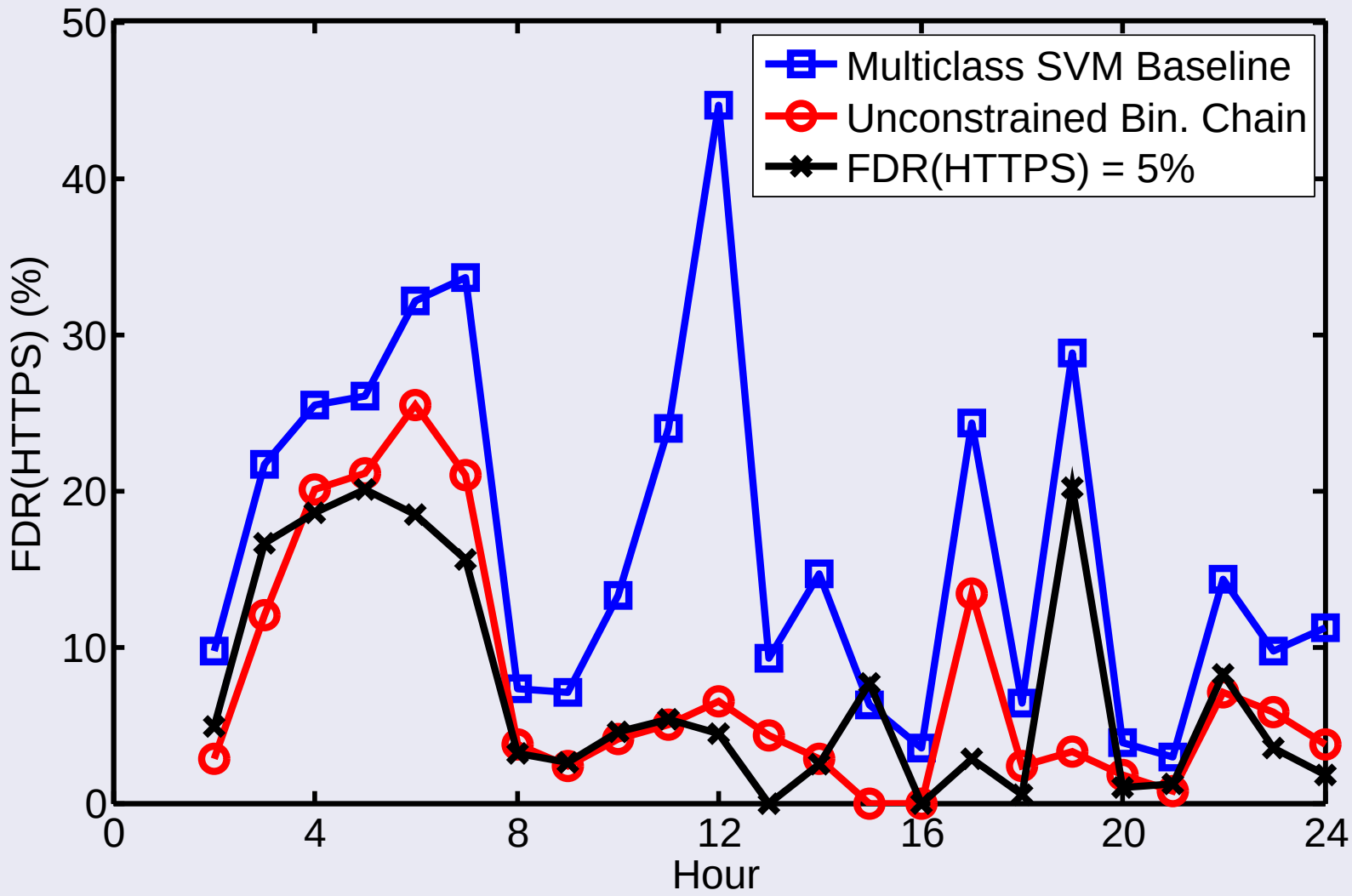
FDR-constrained classifier

Overall Accuracy for Hours 2 - 24



FDR-constrained classifier

FDR(HTTPS) for Hours 2 - 24



Conclusion

Summary

- Two novel algorithms for Internet traffic classification proposed
- Able to provide performance guarantees
- Validated our approach with data provided by an ISP

On-going Research

- Experiment on a more diverse data set
- Creating a hybrid classifier