Behavior-based Tracking

Tracking Users on the Internet with Behavioral Patterns: Evaluation of its Practical Feasibility

Christian Banse
Fraunhofer AISEC

Dominik Herrmann, Hannes Federrath
University of Hamburg, Germany

Agenda

Motivation & Scenario

Tracking Technique

Evaluation

Countermeasures

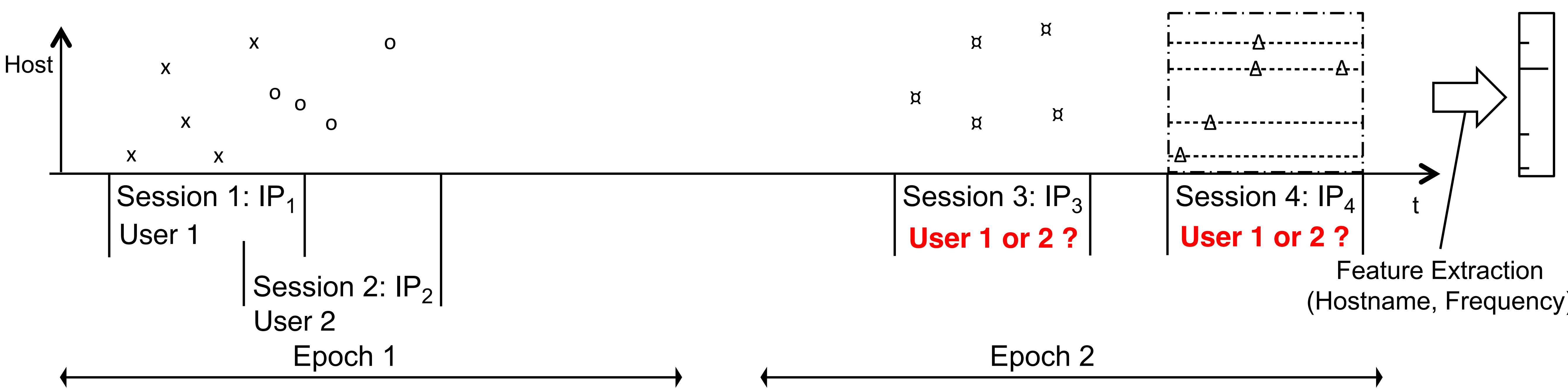
Motivation

- **Explicit tracking** with cookies or other unique IDs is common practice today on the Internet
- We study **behavior-based tracking**
  - works without cookies
  - exploits characteristic patterns within users' activities (in this paper: hostnames contained in DNS queries)

- **Objective:** passive linkage of consecutive sessions
  - without the user's cooperation
  - tracking cannot be detected

Our Scenario and Conceivable Attackers

- Users are represented by dynamic IP addresses that change after fixed amount of time (**epochs** of 24 hours)
- **Observer,** who can record interactions of its users with destination hosts, e.g., a third-party DNS resolver or a web proxy server; *also:* ad networks



Agenda

Motivation & Scenario

Tracking Technique

Evaluation

Countermeasures

Behavior-based Tracking can be Modeled as a Classification Problem

class	=	user (pseudonym)
instance	=	session observed in one epoch
attribute	=	accessed hostname
attribute value	=	number of queries to hostname

Example instance for user *u* in epoch *e* :

www.google.de	45
www.facebook.com	12
www.cnn.com	2
…	

Instance vector:
(..., 45, ..., 12, ..., 2, ...)

We use the Multinomial Naïve-Bayes (MNB) Classifier for Session Linkage

- Popular classifier used for text mining (e.g. for spam detection).
  We apply it to observed hostnames.
- Application of MNB motivated by power-law distribution of access frequencies (very similar to human language)
- **Classification Rationale of MNB:** the more often frequently accessed hosts seen during training of some class *c* do appear in a given test instance, the more likely does this test instance belong to *c*
- For the paper we ported the MNB implementation of *Weka* to Apache Hadoop **(MapReduce)** and also built a fully automated evaluation suite.

Applying Best Practice Transformations

- Access frequencies are scaled down by a sub-linear transformation to minimize bias by large values
- All vectors are normalized to uniform Euclidean length

TFN

- Weight of common/popular hostnames, which are accessed by many users, is reduced

IDF

- Characteristic patterns of adjacent queries are extracted

N-GRAMS

Agenda

Motivation & Scenario

Tracking Technique

Evaluation

Countermeasures

We Study the Feasibility of Behavior-based Tracking for the Case of a Malicious DNS Resolver

- Log of DNS queries of users of a German university (mostly students)
- Each user is assigned a unique, static IP address (allows for validation)

- Privacy concerns were addressed
  - Users' source address was replaced with a a pseudonym using a salted hash function (salt was not disclosed to us)
  - Access to log file is limited to authors of this paper

- 4153 users in total, 2123 active users per day on average

Evaluation is Carried out in Two Phases

We simulate sessions with daily changing, dynamic IP addresses.

1. **Cross Validation (CV)**
   to assess suitability of classifier
   - 3000 randomly chosen users
   - 20 randomly selected sessions per user
   - 10-fold CV: 18 training sessions, 2 test sessions

RESULTS
next slide

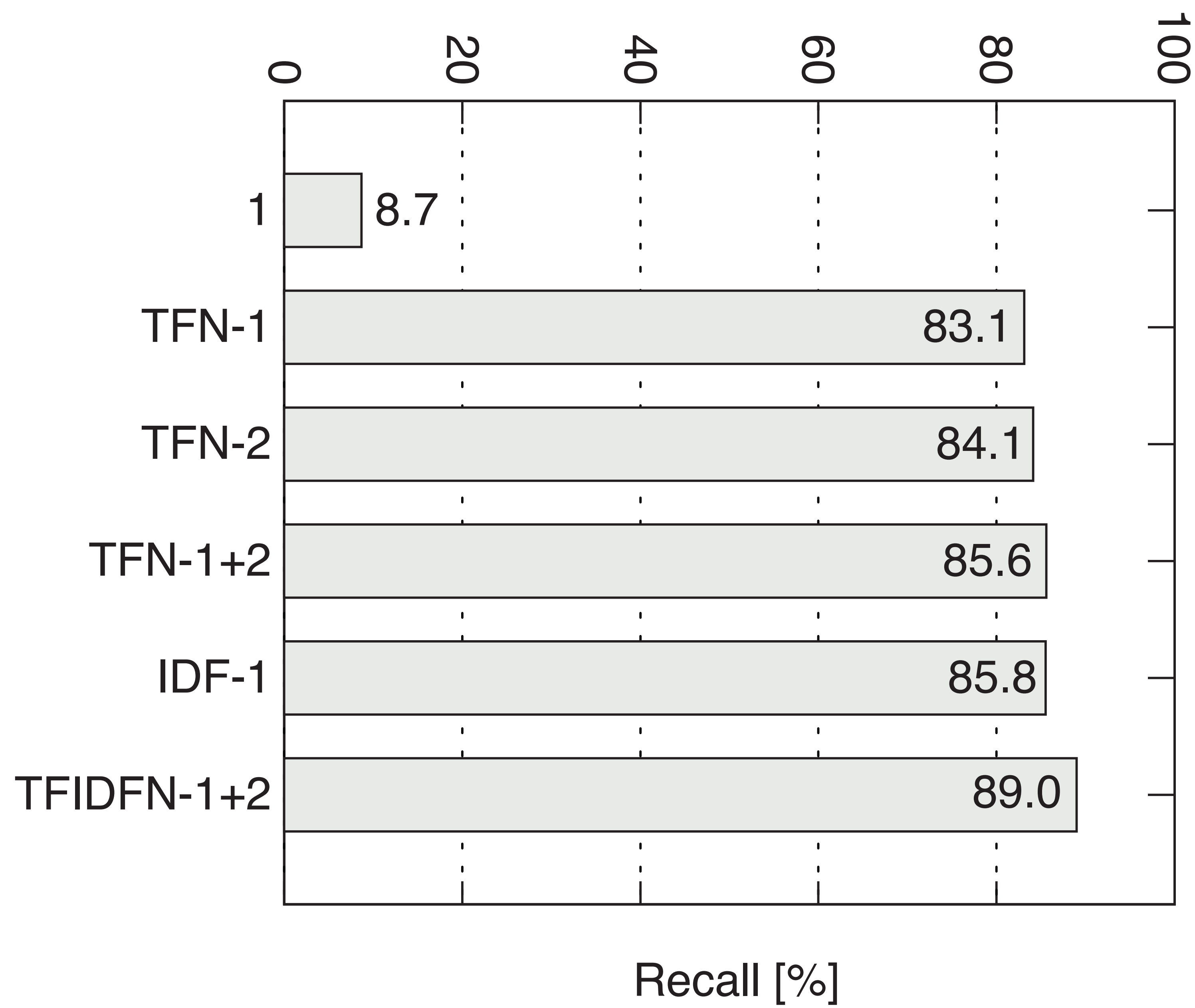
2. **Real-World Evaluation**
   using actual day-to-day traffic from log files

Phase 1: Cross Validation Results Demonstrate that Transformations are Effective

Recall: avg. proportion of correctly classified test instances

TFN + IDF + 1+2-grams achieves the best result

No further improvements by addition of higher-order n-grams

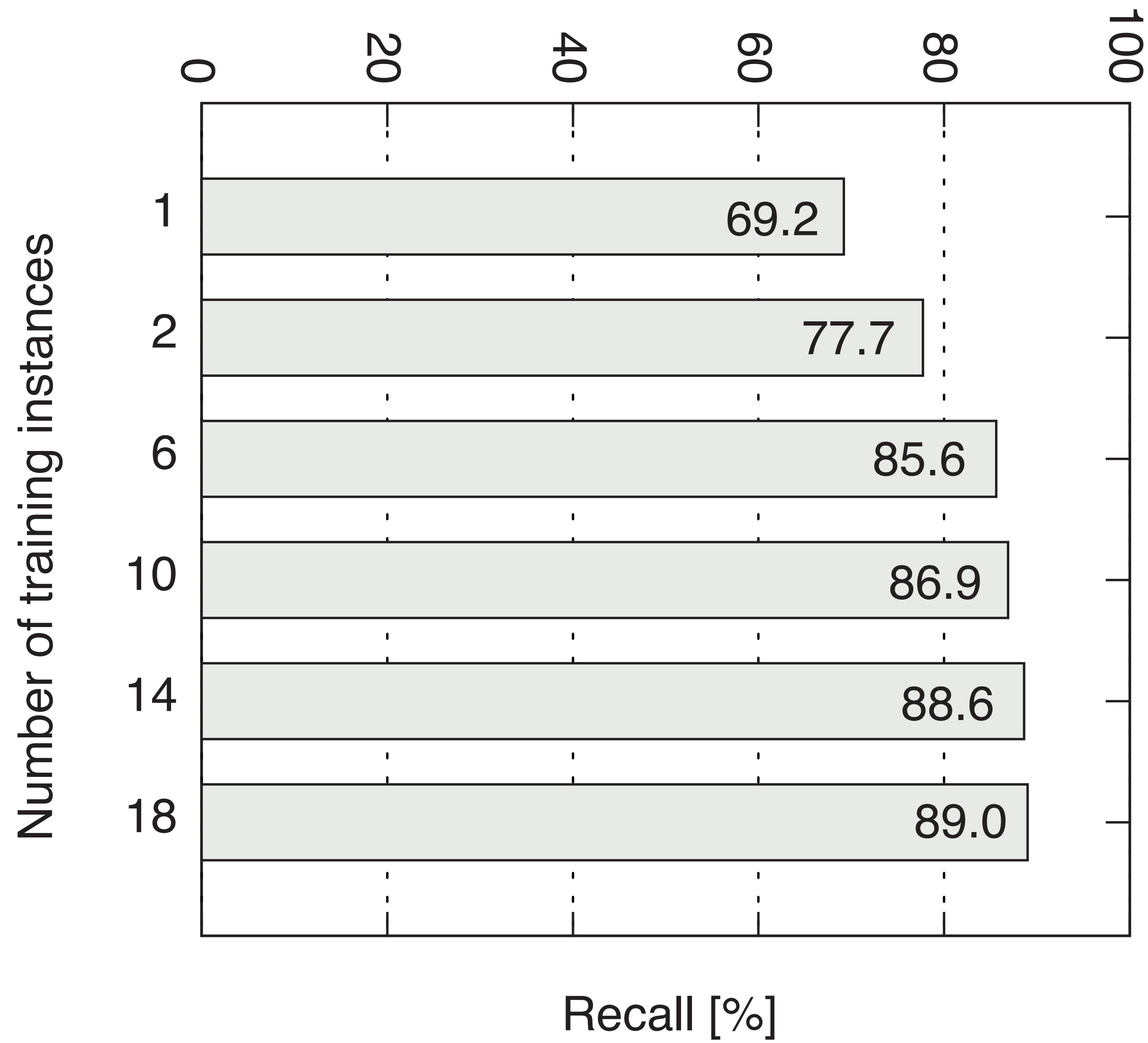


Is Session Linkage Still Possible with only 1 Training Instance?

Assumed observer does not have access to 18, but only 1 instance per user for training!

Repetition of the previous experiment **with less training instances** causes recall to drop (as expected)

Result for 1 instance is quite promising: avg. recall is 69.2 %



Phase 2: Evaluation of all MNB Configurations in Real-World Setting

We simulate a service provider who tries to track all users from day to day

PROCEDURE

- Split log file into 24 hour epochs starting at midnight
- iterate over each epoch *e*
  - for every active user in *e* set up a class *c* and train the MNB classifier with the corresponding instance **(training instances)**
  - predict most probable class for all **test instances** present in $e + 1$ using model built from training instances in *e* (i.e., link the sessions)
  - compare classifier's prediction with ground truth from DNS log file
- Report *avg. accuracy* for all users on all days

Our Accuracy Metric is an Indication of the Proportion of "Correct Links"

Correct
(= "accuracy")

- If u is active on both days and the classifier assigned only his instance to the class of u
- If u is *inactive* on $e + 1$ and the classifier assigned no instance to the class of u

Type 1 Error
(non-detectable)

- If exactly one instance is assigned to the class of u that is from a different user $v \neq u$

Type 2 Error
(detectable)

- If instances from multiple users (maybe including u) are assigned to the class of u

Our Accuracy Metric is an Indication of the Proportion of "Correct Links"

TFN + IDF + 1+2-grams:

Correct (= "**accuracy**")	***76.6%***	‣ If *u* is active on both days and the classifier assigned only his instance to the class of *u* ‣ If *u* is *inactive* on $e + 1$ and the classifier assigned no instance to the class of *u*
Type 1 Error (non-detectable)	***9.8%***	‣ If exactly one instance is assigned to the class of *u* that is from a different user $v \neq u$
Type 2 Error (detectable)	***13.6%***	‣ If instances from multiple users (maybe including *u*) are assigned to the class of *u* ***("ambiguous results")***

Analysis of Results Reveals User Fluctuation to be Responsible for Most of the Type 2 Errors

- In contrast to the cross validation setting a real-world observer is faced with **user fluctuation**. The classifier will encounter
  - **training instances** for which no test instance exists in the consecutive epoch because the user was inactive (students that leave the city on weekends)

  as well as

  - **test instances** for which no class has been trained on the previous day (students that return Sunday evening)

- But: the default implementation of the MNB classifier will assign every instance it encounters to the most likely class, no matter what!

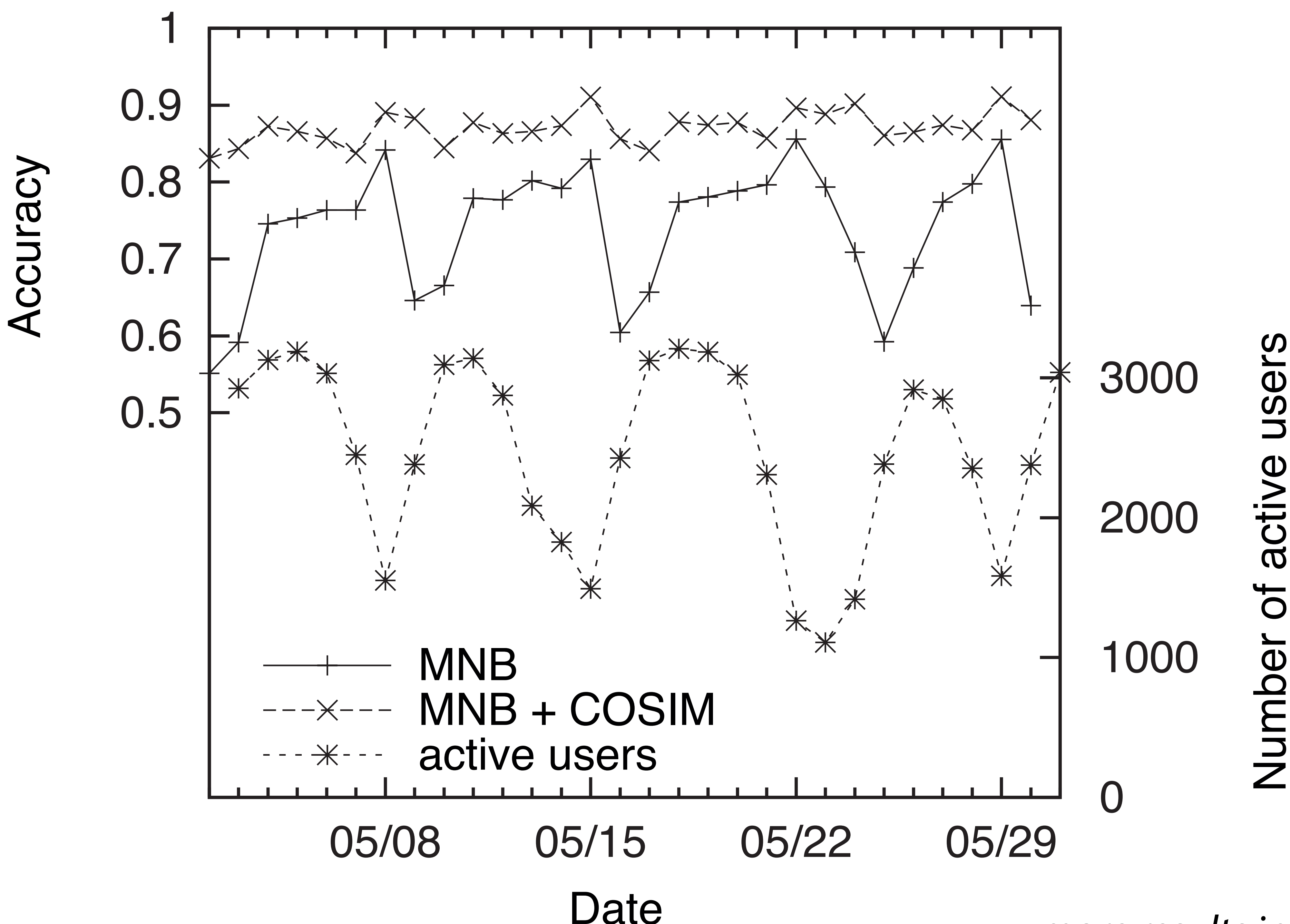
Resolving Ambiguous Predictions with the Cosine Similarity Decision Criterion

- Observer cannot know whether or not the instance of the correct user is part of an *ambiguous result*, but he can make an educated guess.

PROCEDURE

- determine **cosine similarity** between the training instance of c and all the test instances from the ambiguous result in $e + 1$
- select the test instance that is **most similar to the training instance**
- drop all the remaining instances that have been assigned to the class

Resolving Ambiguous Results is Effective: Average Accuracy Increases from 76.6% to 88.2%



more results in the paper

Agenda

- Motivation & Scenario
- Tracking Technique
- Evaluation
- **Countermeasures**

Three Countermeasures Considered Briefly

- **Caching system** to hide patterns caused by repeated requests
  - consider the extreme case: only 1 request per day can be observed
  - only limited effectiveness: accuracy drops from 88.2% to 80.5%
- **Range Queries**
  - issue multiple dummy queries to hide the actual query
  - In case of 5 random dummies per actual query, which are selected from a set of 5000 random hostnames, accuracy drops to 10%
- **“Very” dynamic IP addresses**
  - accuracy drops to 60% for sessions of 3 hours (50% for 1 hour)
  - IPv6 may offer opportunities for implementing better protection

Behavior-Based Tracking

- Using a DNS query log we studied whether **linking consecutive sessions based on behavioral patterns** is feasible in practice
- Our MapReduce implementation of **Multinomial Naïve Bayes** classifier correctly links majority of sessions for a group of up to 3000 concurrent users
- In real-world setting user fluctuation causes **ambiguous results** that can be resolved using **cosine similarity criterion**
- **Changing IP addresses multiple times per day** offers only limited protection against behavior-based tracking

Christian Banse
Fraunhofer AISEC

Dominik Herrmann, Hannes Federrath
University of Hamburg, Germany

Fraunhofer AISEC

SVS

http://tinyurl.com/bbtracking

BACKUP

Feasibility of Behavior-based Tracking cannot be Deduced from Prior Studies

FEASIBILITY DEPENDENCIES

- more difficult for more users
- less difficult if more training data is available

RELATED STUDIES

- Yang (2010): session linkability accuracy of 87% (100 training sessions, 100 concurrent users), but only 62% with 1 training session
- Kumpost et al. (2009): false positive rate of 68% using destination IPs from monthly aggregated NetFlow logs

OUR PREVIOUS WORK

- Herrmann et al. (2010): accuracy of 73 % with 1 training session using HTTP traffic of 28 users