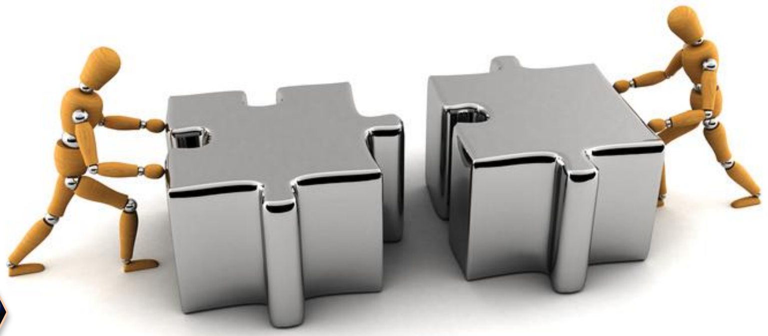
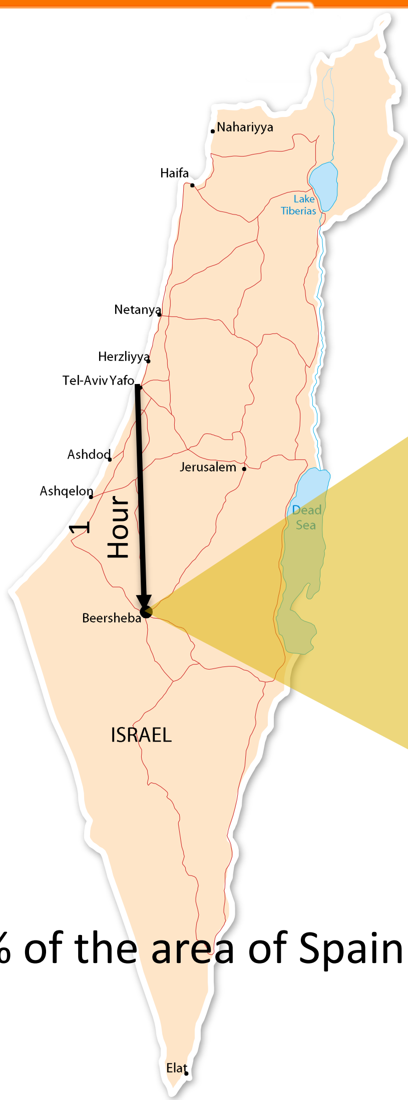
When Cyber Security Meets Machine Learning

Lior Rokach

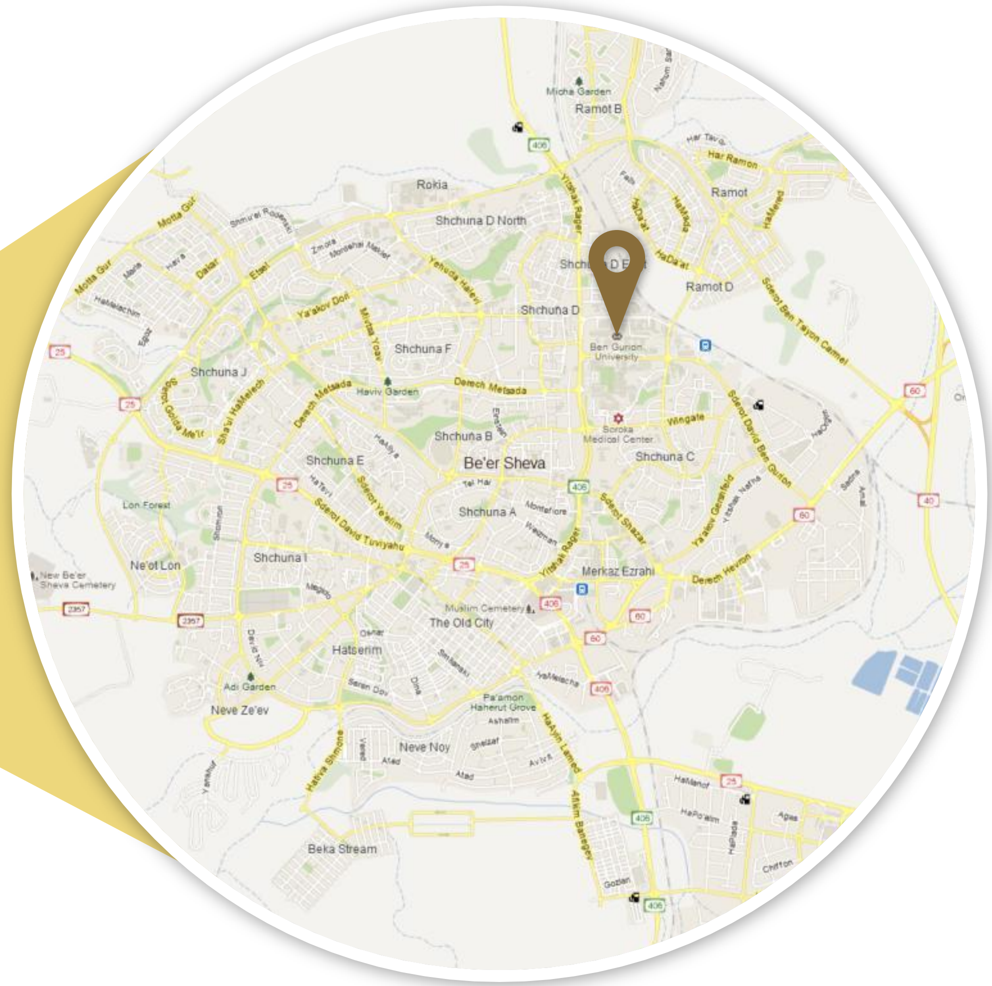


Ben-Gurion University of the Negev

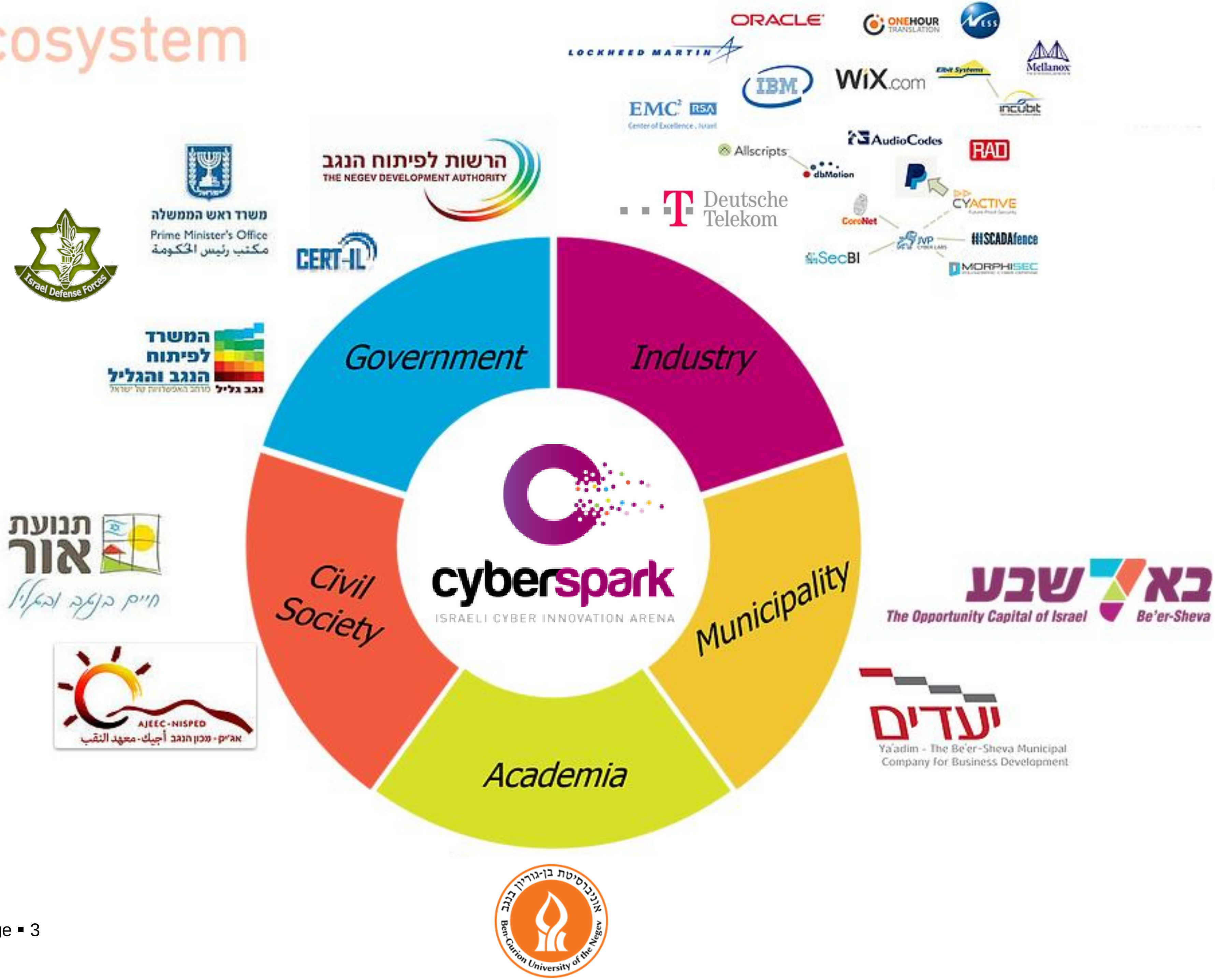
Beer Sheva



4% of the area of Spain



Ecosystem


Government
Industry
Municipality
Academia
Civil Society
cyberspark
ISRAELI CYBER INNOVATION ARENA
ORACLE
ONEHOUR TRANSLATION
LOCKHEED MARTIN
IBM
WiX.com
Elbit Systems
Mellanox
EMC² RSA
Center of Excellence, Israel
incubit
AudioCodes
RAD
Allscripts
dbMotion
Deutsche Telekom
CYACTIVE
CoroNet
JVP
SCADAfence
SecBI
MORPHISEC
הרשות לפיתוח הנגב
THE NEGEV DEVELOPMENT AUTHORITY
משרד ראש הממשלה
Prime Minister's Office
مكتب رئيس الحكومة
CERT-IL
Israel Defense Forces
המשרד לפיתוח הנגב והגליל
נגב גליל
תנועת אור
AJEEC-NISPED
באר שבע
The Opportunity Capital of Israel
Be'er-Sheva
יעדים
Ya'adim - The Be'er-Sheva Municipal Company for Business Development
אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

Beer-Sheva ATP Inauguration

Inauguration Ceremony

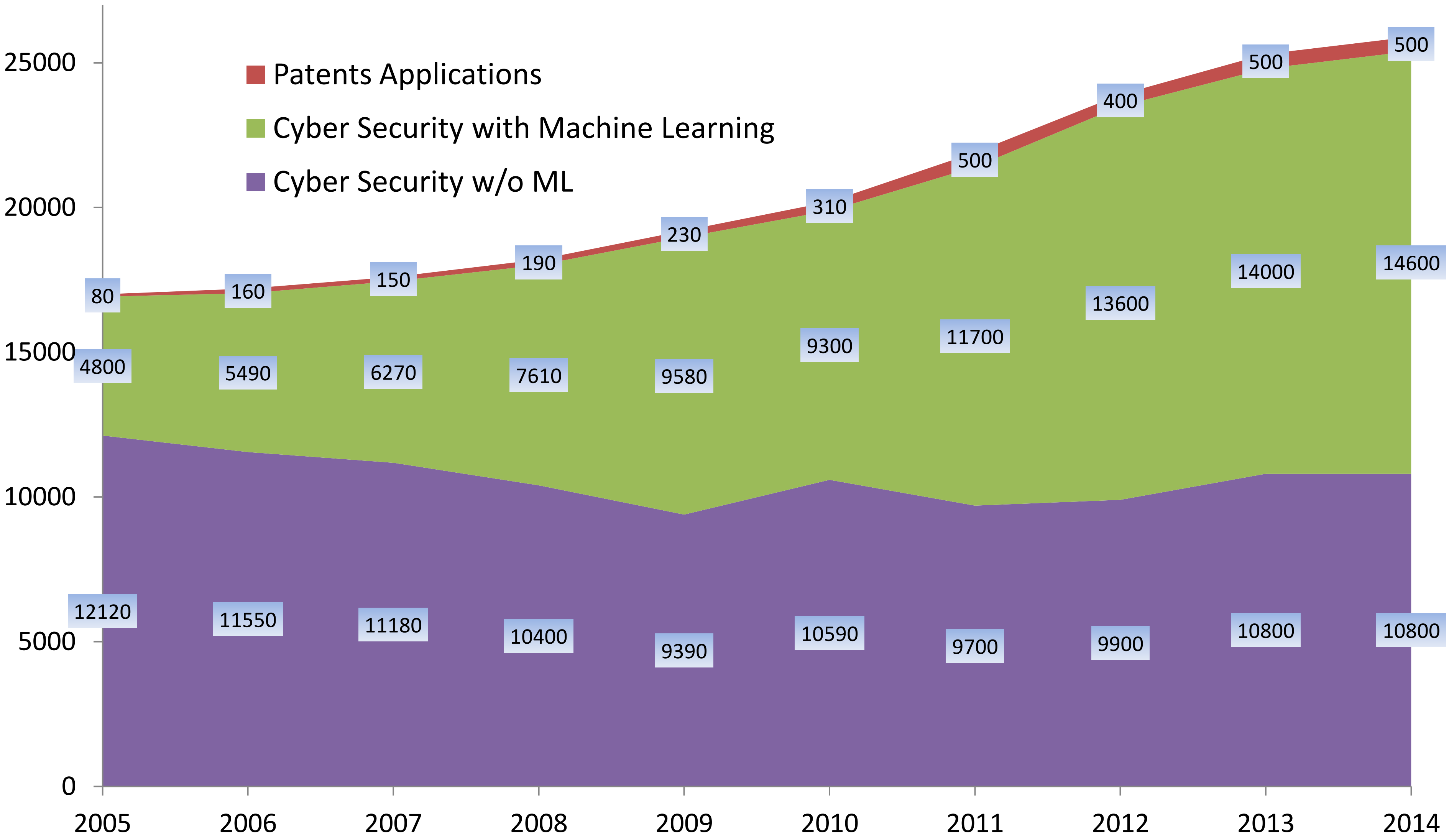
At the inauguration ceremony Prime Minister, Benjamin Netanyahu, declared:

"We are launching the economic anchor that will turn Beer-Sheva into a national and international center for cyber security. We are changing the future of Israel and we are doing it in Beer-Sheva."

All Within a Walking Distance

Cyber Security Machine Learning

Machine Learning in Cyber Security



Successful ML applications in Cyber Security

- Spam Mitigation
- Malware detection
- Mitigating the Denial of Service Attacks
- Reputation in Cyber Space
- User Identification
- Detecting Identity Theft
- Information Leakage Detection and Prevention
- Social Network Security
- Detecting Advanced Persisted Threats
- Detecting Hidden Channels

The concept of learning in a ML system

- Learning = Improving with experience at some task
  - – Improve over task *T*,
  - – With respect to performance measure, *P*
  - – Based on experience, *E*.

Phishing Attack with Social Engineering

Savyon Dafni <savyonda@bgu.ac.il>
to Andreev

May 9

English > Hebrew   Translate message   Turn off for: English

Dear User,

This message is to inform you that your access to the BGU Moodle will soon expire. You will have to login to your account to continue to have access to this service.
You need to reactivate it just by logging in through the following URL. A successful login will activate your account and you will be redirected to your BGU Moodle page.

http://moodle.bgu.ac.raae.cf/login22targetURLNe2T3d0jdVUniti22nde3dHSP2VyO2mp23bdscnt21YUHXLb226N23bdaL226wFmp232hs2alizeIBba22f22floyola23fvidtt46Rstmp23ip23bx2226amClTTp3dtrue/

If you are not able to login, please contact Savyon Dafni at savyonda@bgu.ac.il for immediate assistance.

Sincerely,

Savyon Dafni
Computing & Information Systems
Ben-Gurion University of the Negev
08-6461953
savyonda@bgu.ac.il

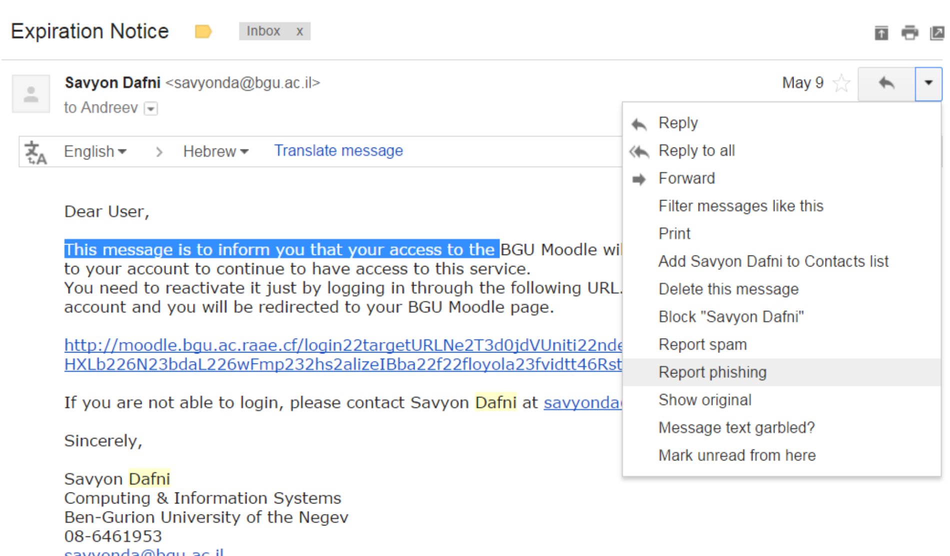
Learning to Filter Spam or Phishing Emails

T: Identify Spam/Phishing Emails

P:

- % of spam/phishing emails that were filtered
- % of ham/ (non-spam) emails that were incorrectly filtered-out

E: a database of emails that were labelled by users



Training

Testing

From Emails to Feature Vectors

- Textual-Based Content Features:
  - Email is tokenized
  - Each token is a feature

- Meta-Features:
  - Number of recipients
  - Size of message
  - Has attachment
  - IP

Textual-Based Content Features Data Set

Vocabulary

Target Attribute

Earn	Lottery	. . .	Free	Email Type
0	1		0	Ham
1	0		1	Ham
0	0		0	Spam
1	1		1	Spam
0	0		0	Ham
0	1		1	Ham
1	0		0	Spam

Instances

Binary/TF

Meta-Features Data Set

Input Attributes

Target Attribute

Number of new Recipients	Email Length (K)	Country (IP)	IP Provider Rank	Email Type
0	2	Germany	Gold	Ham
1	4	Germany	Silver	Ham
5	2	Nigeria	Bronze	Spam
2	4	Russia	Bronze	Spam
3	4	Germany	Bronze	Ham
0	1	USA	Silver	Ham
4	2	USA	Silver	Spam

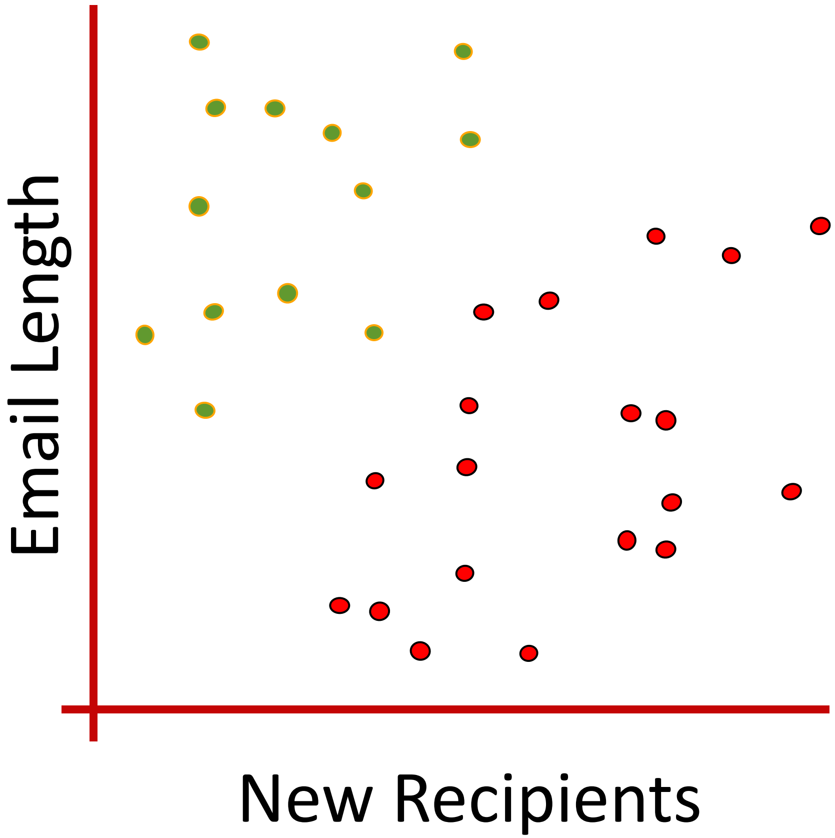
Instances

Numeric

Nominal

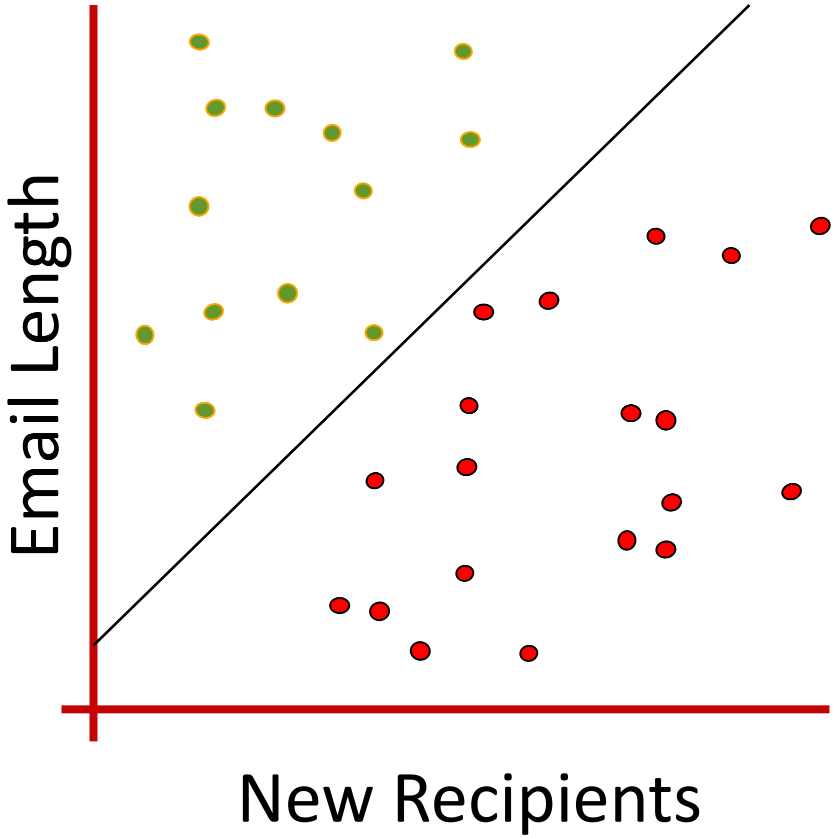
Ordinal

Linear Classifiers



How would you classify this data?

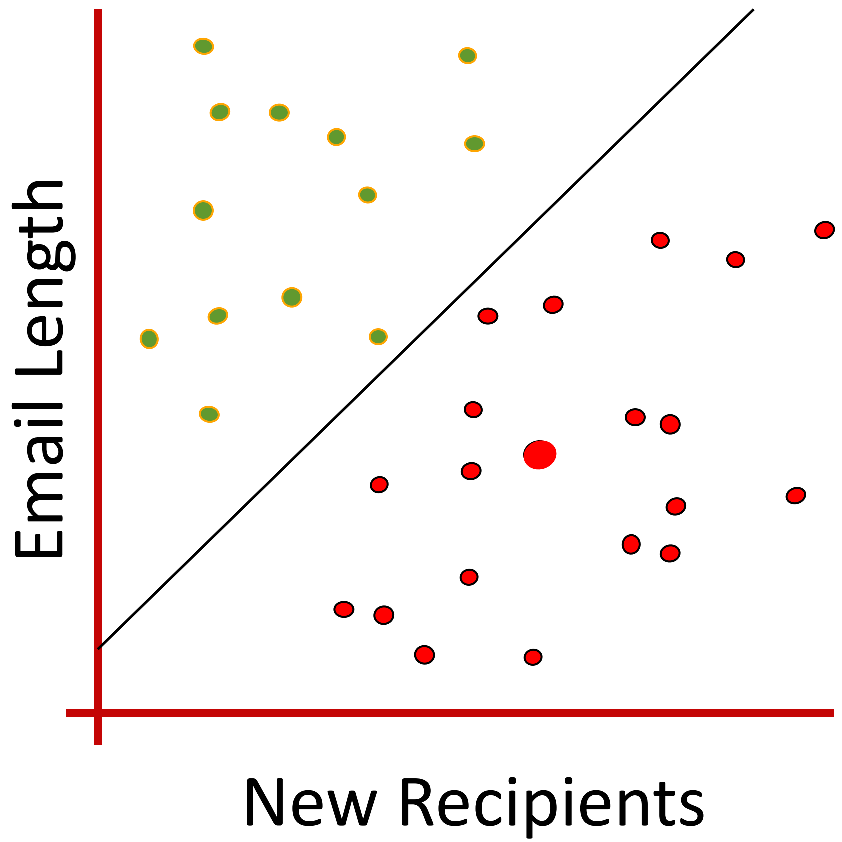
Linear Classifiers



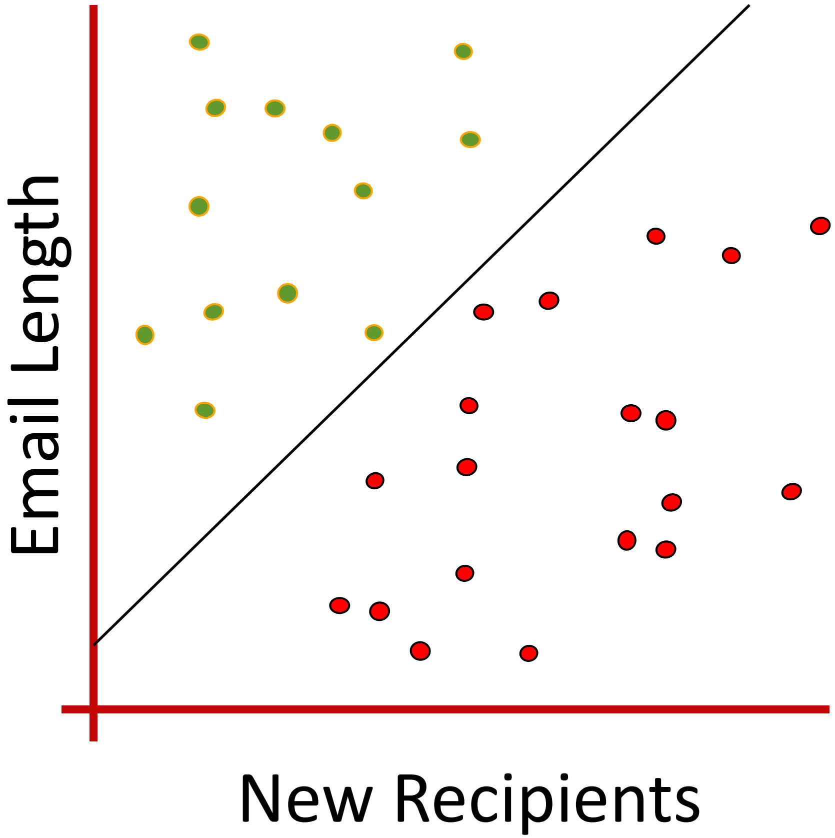
How would you classify this data?

When a new email is sent

1. We first place the new email in the space
2. Classify it according to the subspace in which it resides

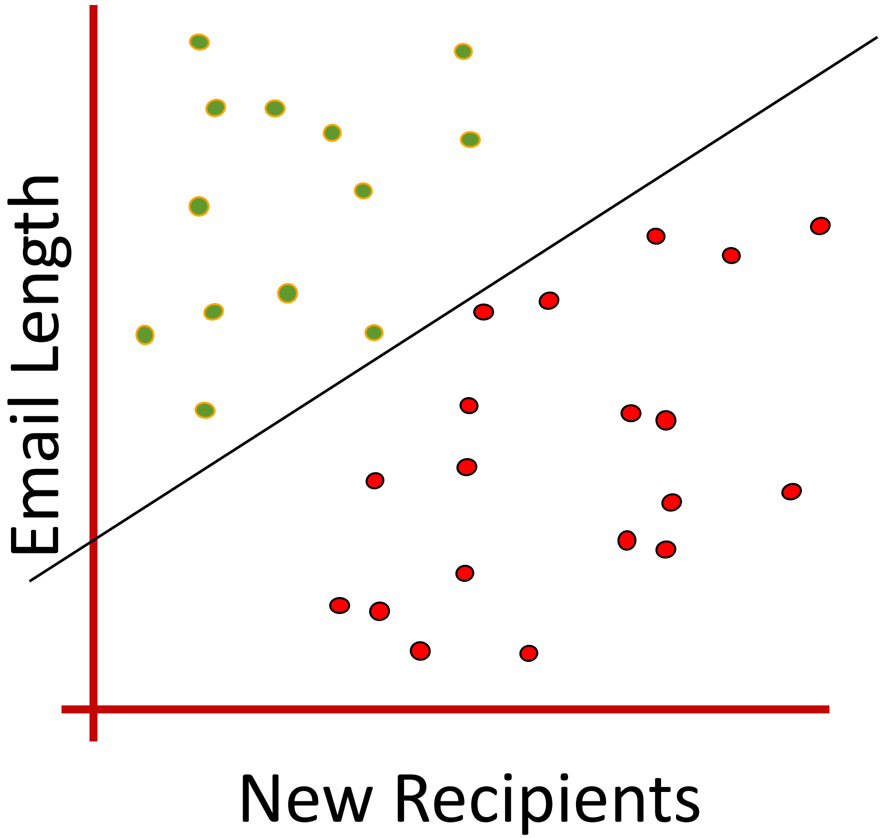


Linear Classifiers



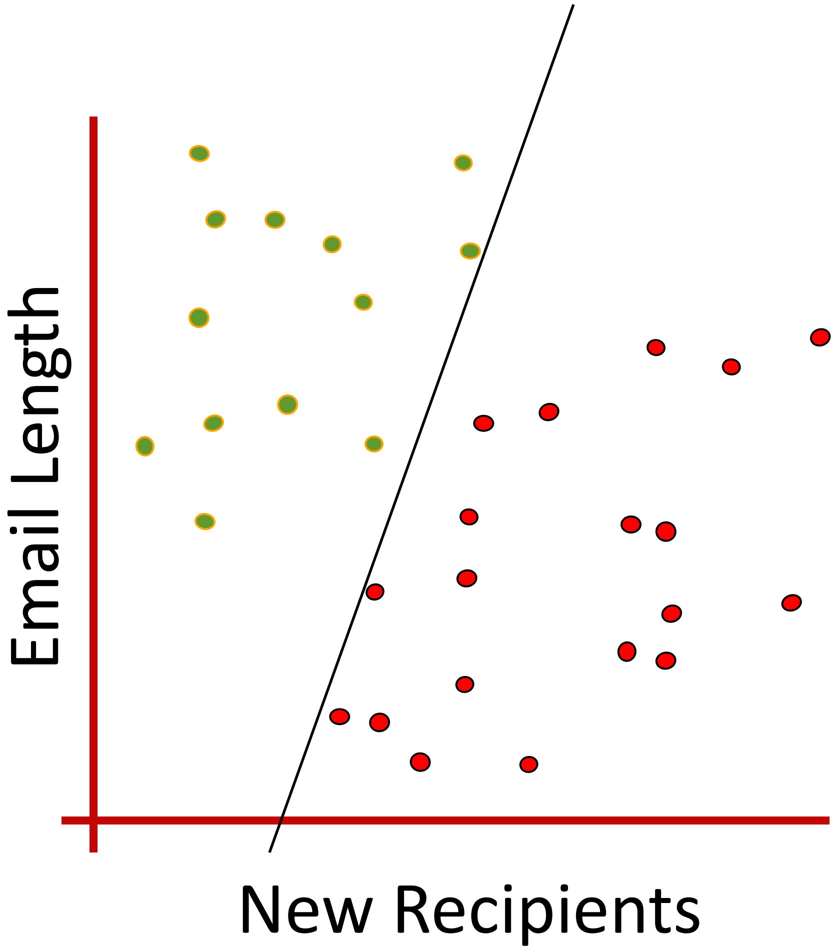
How would you classify this data?

Linear Classifiers



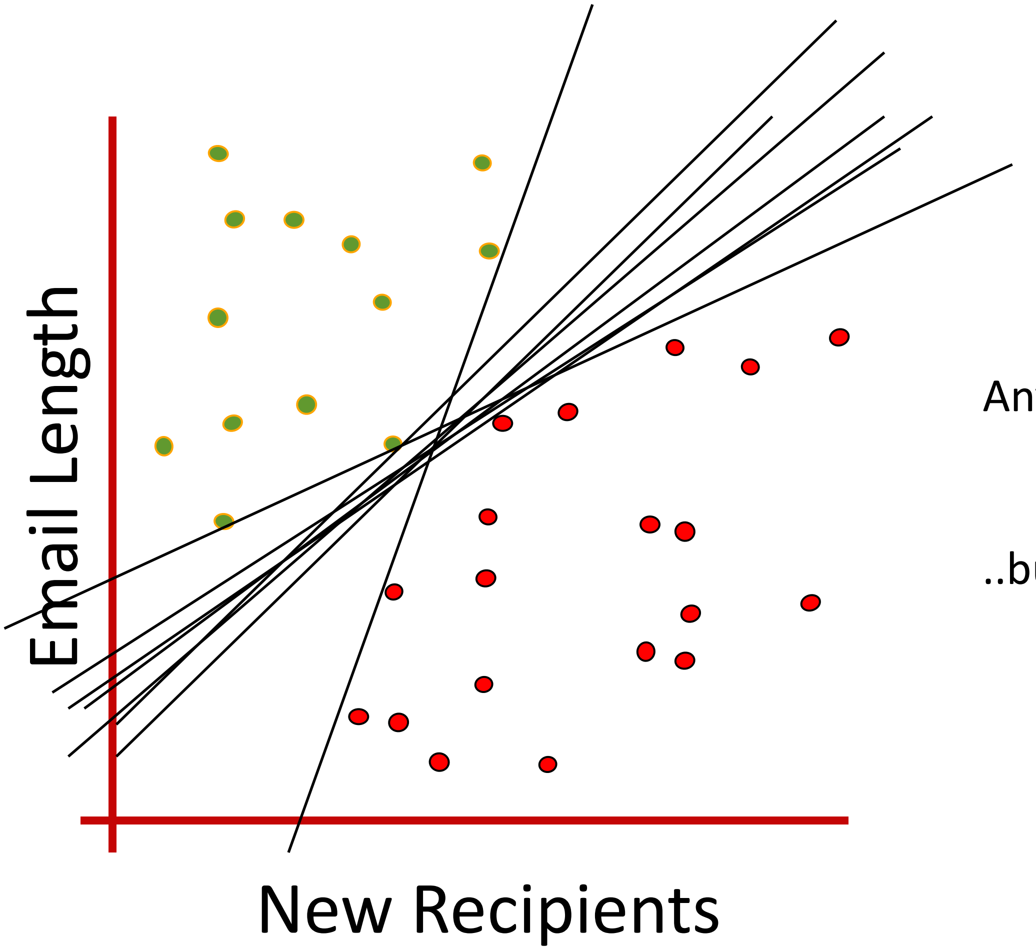
How would you classify this data?

Linear Classifiers



How would you classify this data?

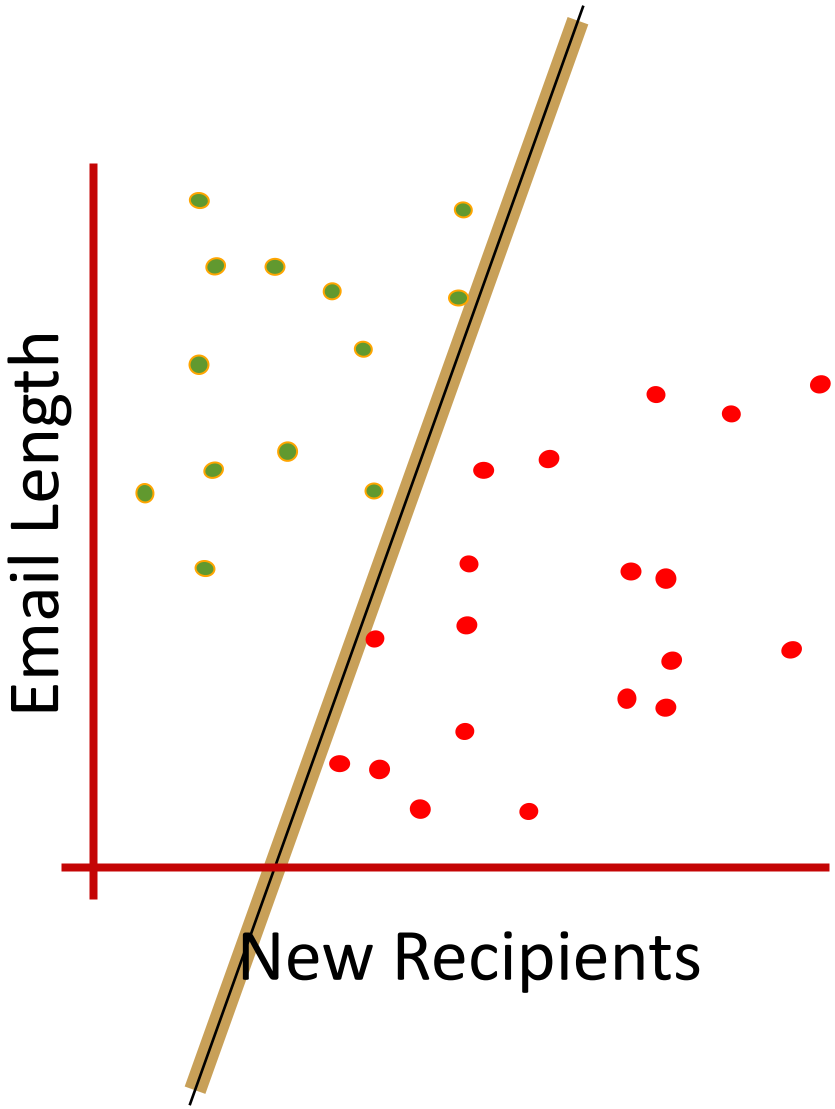
Linear Classifiers



Any of these would be fine..

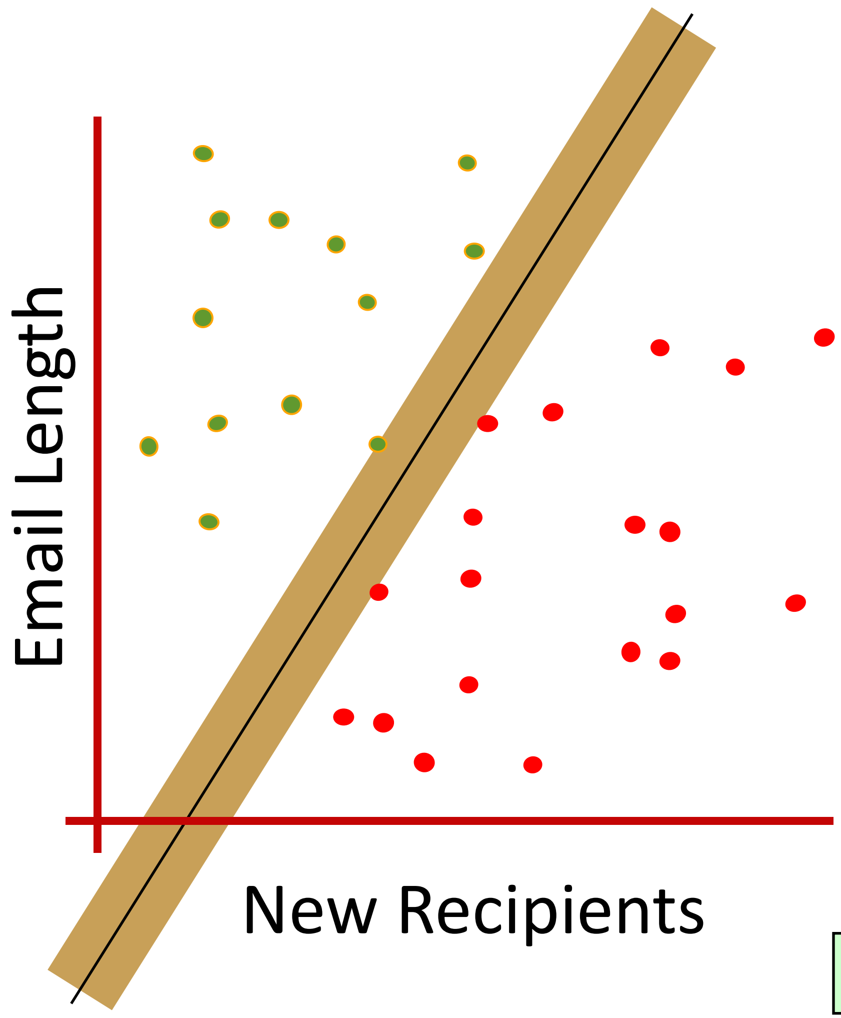
..but which is best?

Classifier Margin



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

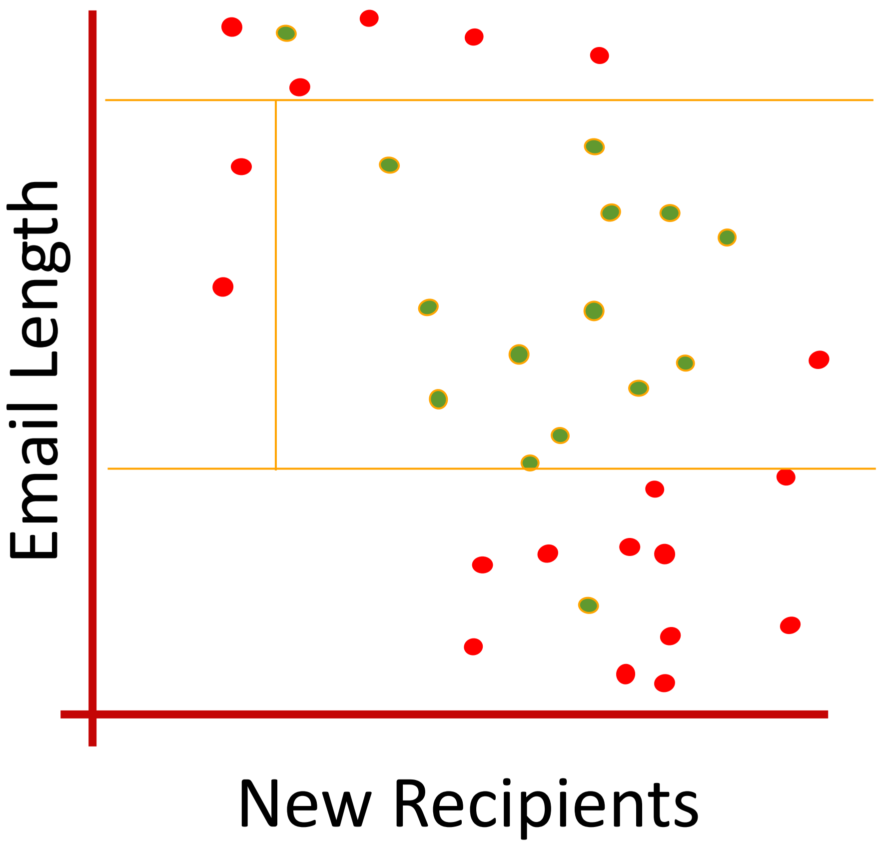
Maximum Margin

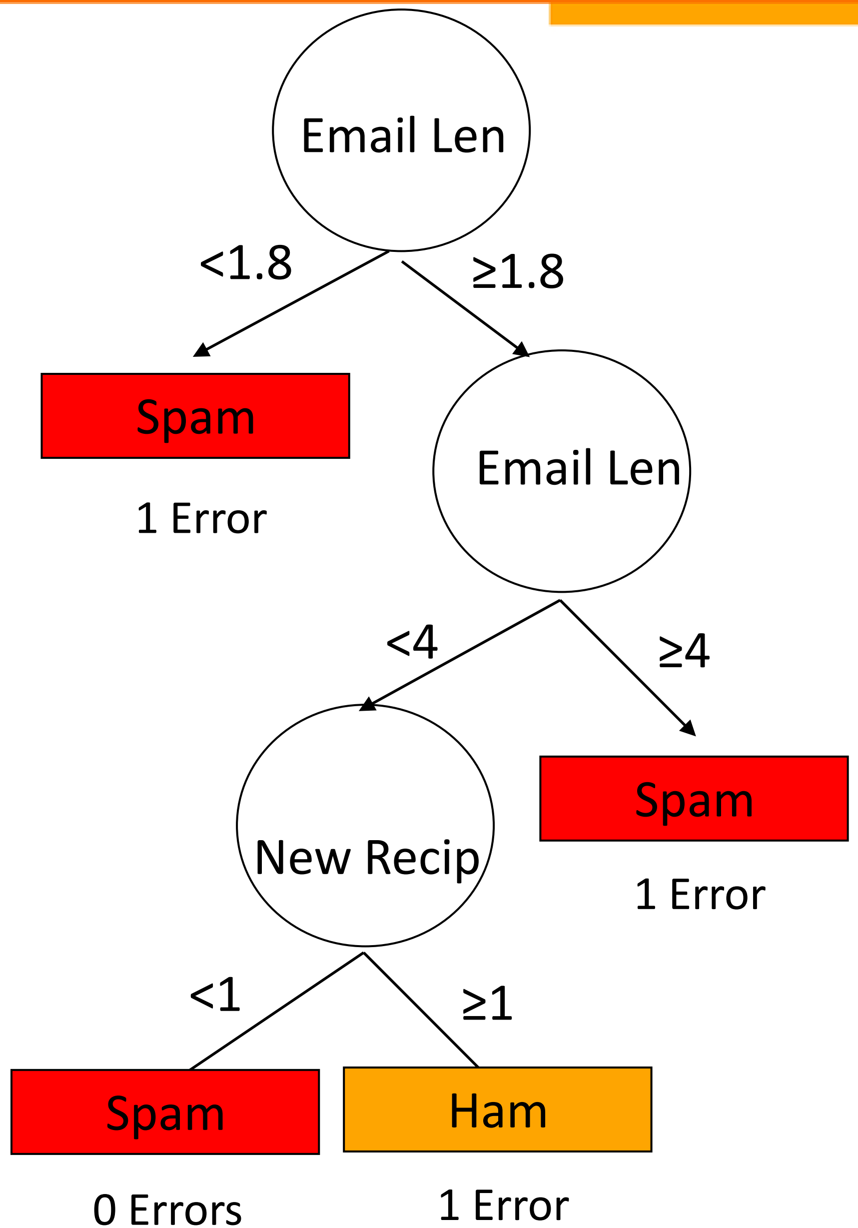


The maximum margin linear classifier is the linear classifier with the, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

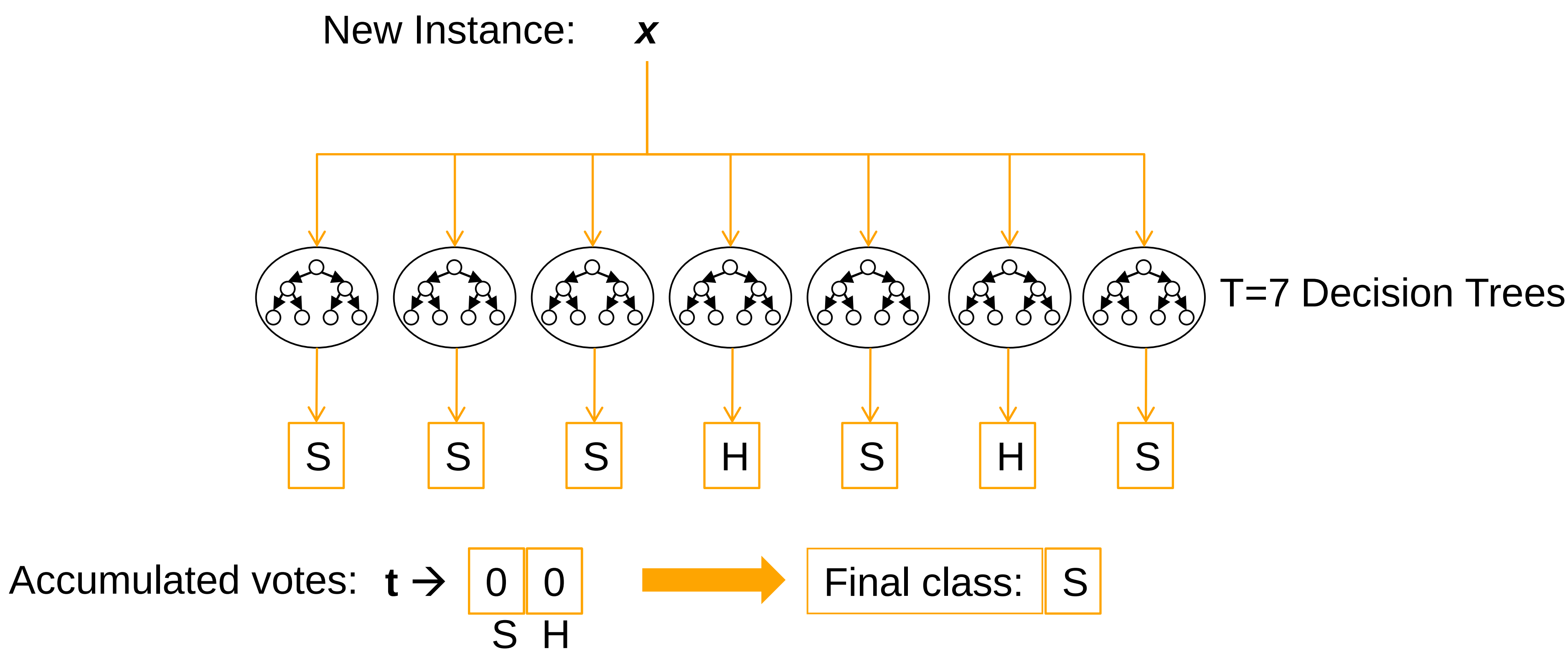
Linear SVM

Top Down Induction of Decision Trees



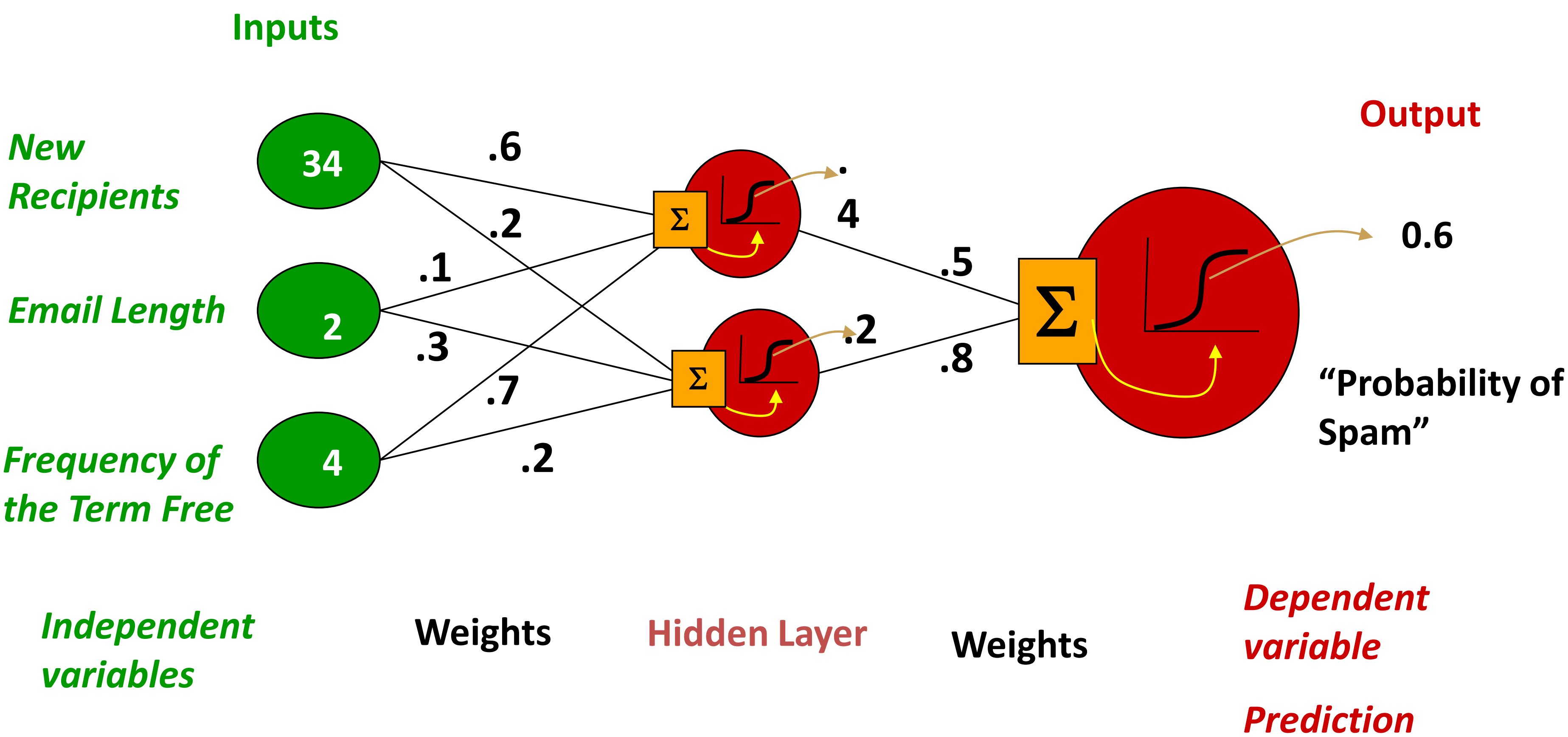


Decision Forest by majority voting



Obtained from Alberto Suárez, 2012

Neural Network Model



Malware Detection

Malware Detection

- Static – Analyze the program (code) –
  - leverage structural information (e.g. sequence of bytes)
  - attempts to detect malware before the program under inspection executes
- Dynamic – Analyze the running process –
  - leverage runtime information (e.g. network usage)
  - attempts to detect malicious behavior during program execution or after program execution.

Features Extraction

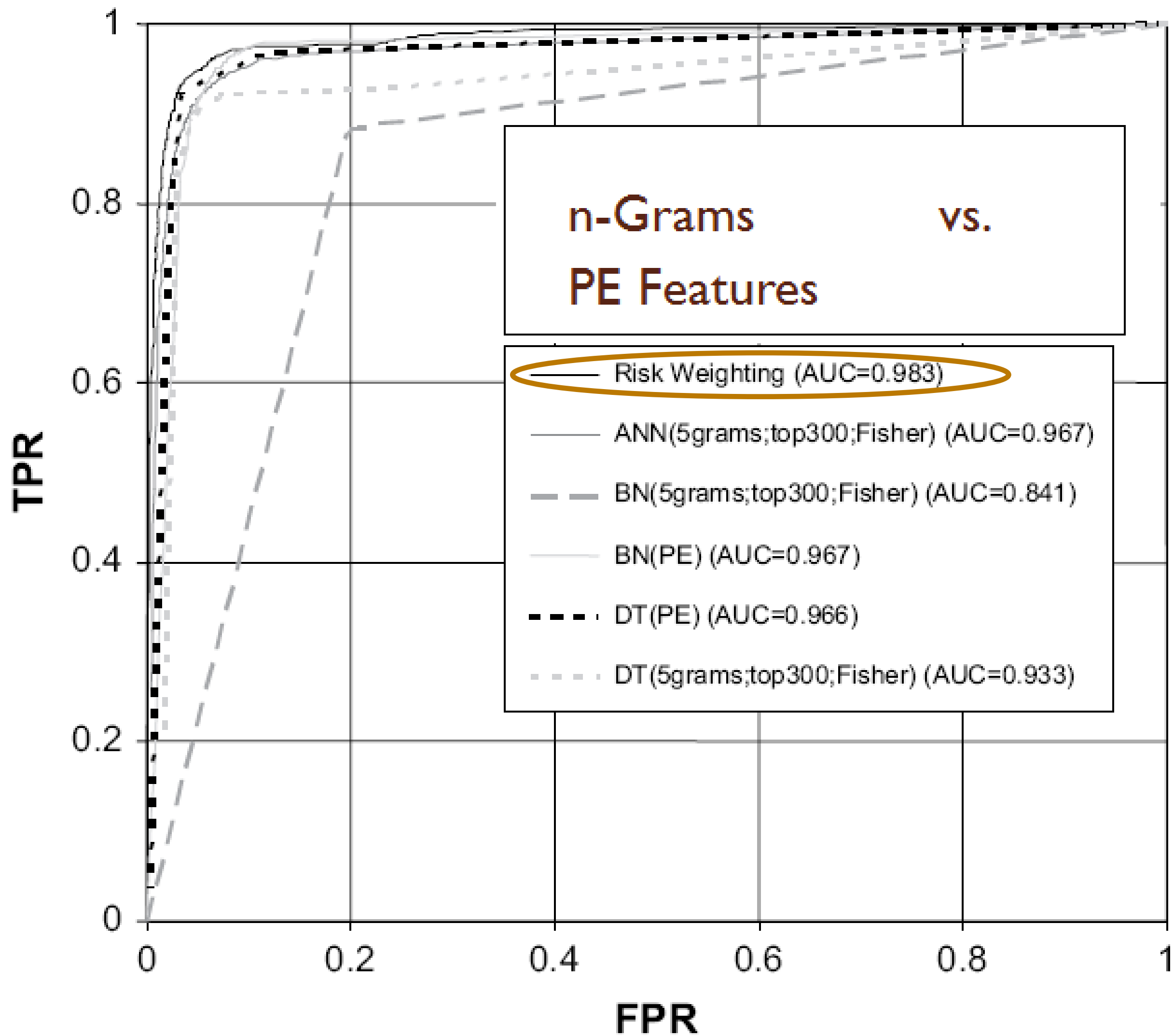
- Creating Vocabularies (TF Vector)

N-Grams	Vocabulary Size
3-gram	16,777,216
4-gram	1,084,793,035
5-gram	1,575,804,954
6-gram	1,936,342,220

Portable Executable (PE)

- Extracted from certain parts of EXE files stored in binaries (EXE or DLL).
- PE Header that describes physical structure of a PE binary (e.g., creation/modification time, machine type, file size)
- Import Section: which DLLs were imported and which functions from which imported DLLs were used
- Exports Section: which functions were exported (if the file being examined is a DLL)
- Resource Directory: resources used by a given file (e.g., dialogs, cursors)
- Version Information (e.g., internal and external name of a file, version number)

n-Grams vs. PE Features



Expert Based Features

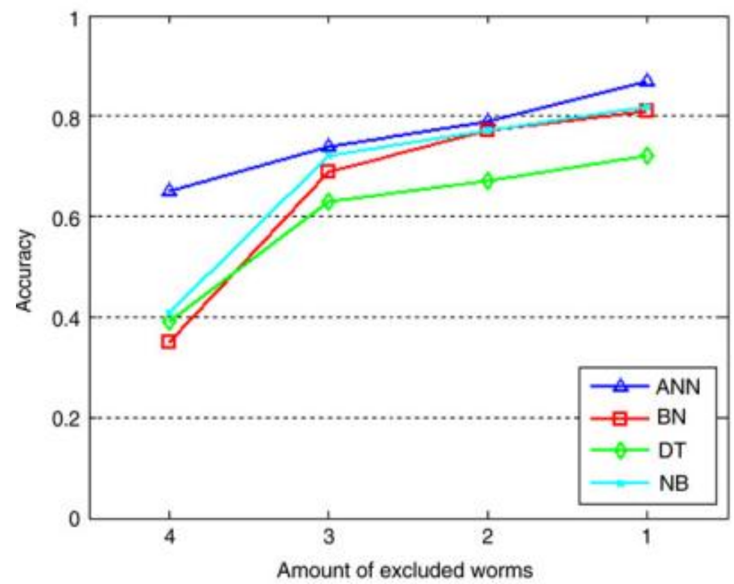
- Look for Common Libraries
- Identify anti-forensic means to avoid their detection
- Aggregate-features – address the “curse of dimensionality” by aggregating the features into a small set of meaningful meta features
- Chronological evolution of malware – Most viruses are variants of previous malwares.

Method	Feature selection	FPR	TPR	Acc	AUC
GR500BDT (un-patched + RF)	Gain Ratio	0.094	0.959	0.948	0.929
Mal-IDP+GR500BDT (patched + RF)	Gain Ratio	0.093	**0.977**	0.963	0.946
Mal-ID basic	Mal-ID	**0.006**	0.909	**0.986**	0.951
Mal-IDF+RF (Mal-ID features + RF)	None	**0.006**	0.916	0.985	**0.995**

G Tahan, L Rokach, Y Shahar, Mal-ID: Automatic Malware Detection Using Common Segment Analysis and Meta-Features, Journal of Machine Learning Research 1 (2012) 1-48

Dynamic Analysis for Unseen Malware

All (V_All)	
ChiSqr	ReliefF
_A_1MemoryCache_Bytes_Peak_	_A_1ICMPReceived_Dest__Unreachable_
_A_1Process__Total_Virtual_Bytes_Peak_	_A_1ICMPSent_Destination_Unreachable_
_A_1MemoryFree_System_Page_Table_Entries_	_A_1SystemFile_Control_Bytes_sec_
_A_1Process__Total_Virtual_Bytes_	_A_1Process__Total_IO_Other_Bytes_sec_
_A_1Process__Total_Pool_Nonpaged_Bytes_	_A_1ICMPMessages_Outbound_Errors_
_A_1MemoryPool_Nonpaged_Bytes_	_A_1MemorySystem_Code_Total_Bytes_
_A_1Process__Total_Thread_Count_	Netobj_disconnect
_A_1SystemThreads_	_A_1ICMPSent_Echo_sec_
_A_1Process__Total_Pool_Paged_Bytes_	_A_1ICMPMessages_Sent_sec_
_A_1TCPConnections_Active_	_A_1Process__Total_Handle_Count_
_A_1Network_Interfac___Packet_Scheduler_Miniport _Bytes_Sent_sec_	_A_1ICMPMessages_sec_
_A_1TCPConnection_Failures_	_A_1Processor__Total___Processor_Time_
_A_1MemoryPool_Nonpaged_Allocs_	_A_1SystemException_Dispatches_sec_
_A_1Process__Total_Handle_Count_	_A_1TCPConnections_Reset_
_A_1Network_InterfacTX____Packet_Scheduler _Miniport_Packets_sec_	_A_1Processor__Total___Idle_Time_
_A_1Network_Interfac__Packet_Scheduler_Miniport _Bytes_Total_sec_	_A_1Processor__Total___User_Time_
_A_1Process__Total_Page_File_Bytes_Peak_	_A_1Process__Total___User_Time_
_A_1IPDatagrams_sec_	_A_1Thread__Total__Total___User_Time_
_A_1SystemFile_Control_Bytes_sec_	_A_1Processor__Total_Interrupts_sec_
_A_1Process__Total_IO_Other_Bytes_sec_	_A_1Memory__Committed_Bytes_In_Use_



Computer	Background application	User activity
Old	No	No
Old	No	Yes
Old	Yes	No
Old	Yes	Yes
New	No	No
New	No	Yes
New	Yes	No
New	Yes	Yes

R Moskovitch, Y Elovici, L Rokach, Detection of unknown computer worms based on behavioral classification of the host, Computational Statistics & Data Analysis 52 (9), 4544-4566

Active Learning Framework for Detecting Malicious PDF Files

- PDF files may contain malicious functionality:
  - JavaScript code.
  - Embedded files. (Executables, PDF, MS-office, Flash)
  - Form submissions and URI attacks.

- Scanning **20M** of scholarly papers with VirusTotal reveal 0.5% are infected with a malware.

- Known malicious PDF files are detected by AV using signatures.

- Unknown malicious PDF files evade AV.

- AV must be frequently updated with new malicious PDF files.

Attacking Open-Web Academic Libraries (Google, CiteseerX, etc.)

- Grant access to an university web-page (e.g. individual home page)
- Find a well-cited paper (not even your paper)
- Put its PDF in the web-site
- Wait for Google Scholar to index the paper
- Add malicious code to your PDF
- Wait for users to be infected by the file

The Challenge

- Both AV must be frequently updated.
- Many new PDF files to inspect (mass daily creation).
- Security experts are a limited resource for inspection.
- Therefore - only part of the new files can be inspected.

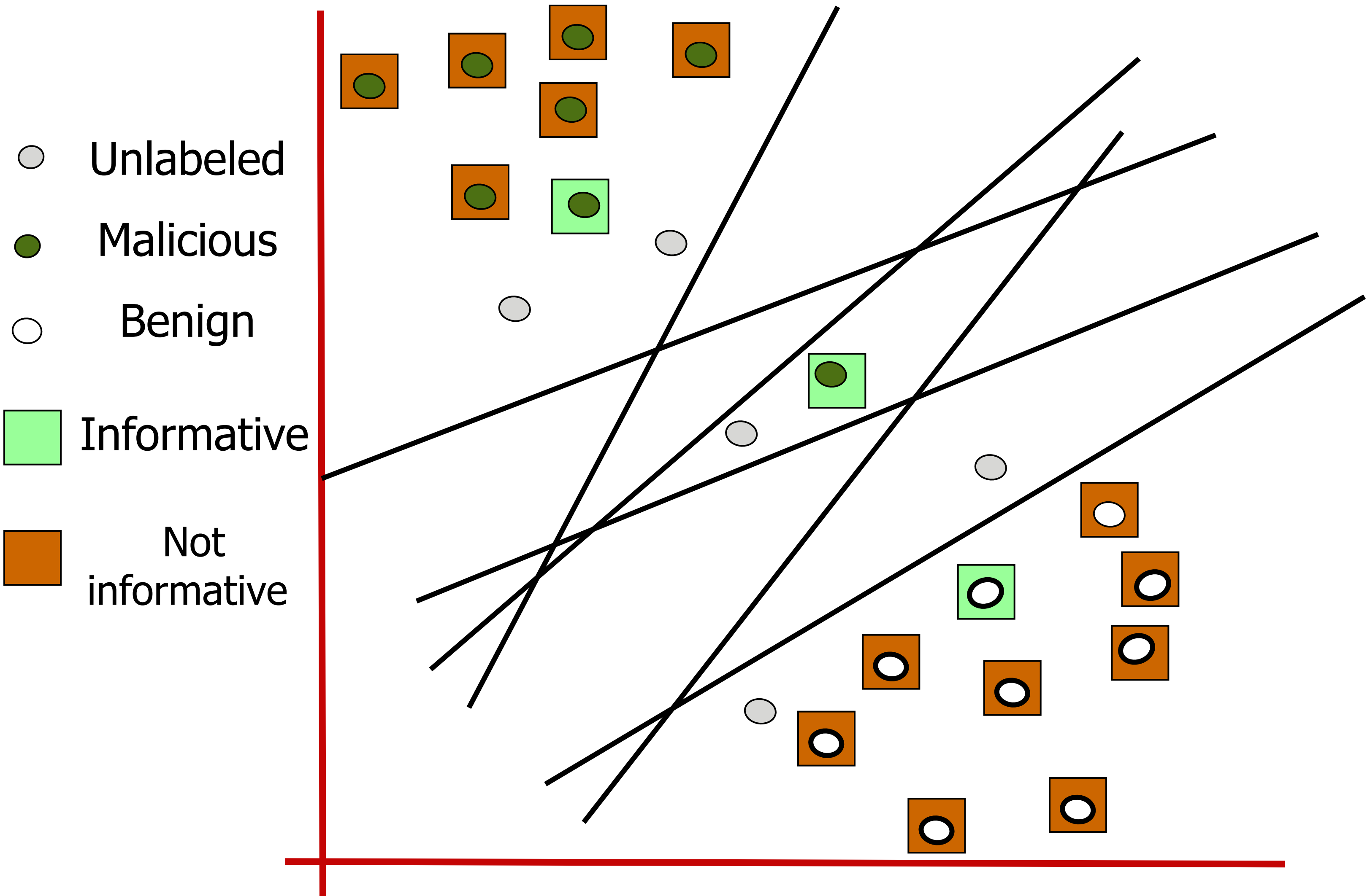
  - <u>Which of the new PDF files need to be inspected?</u>

Possible Approach

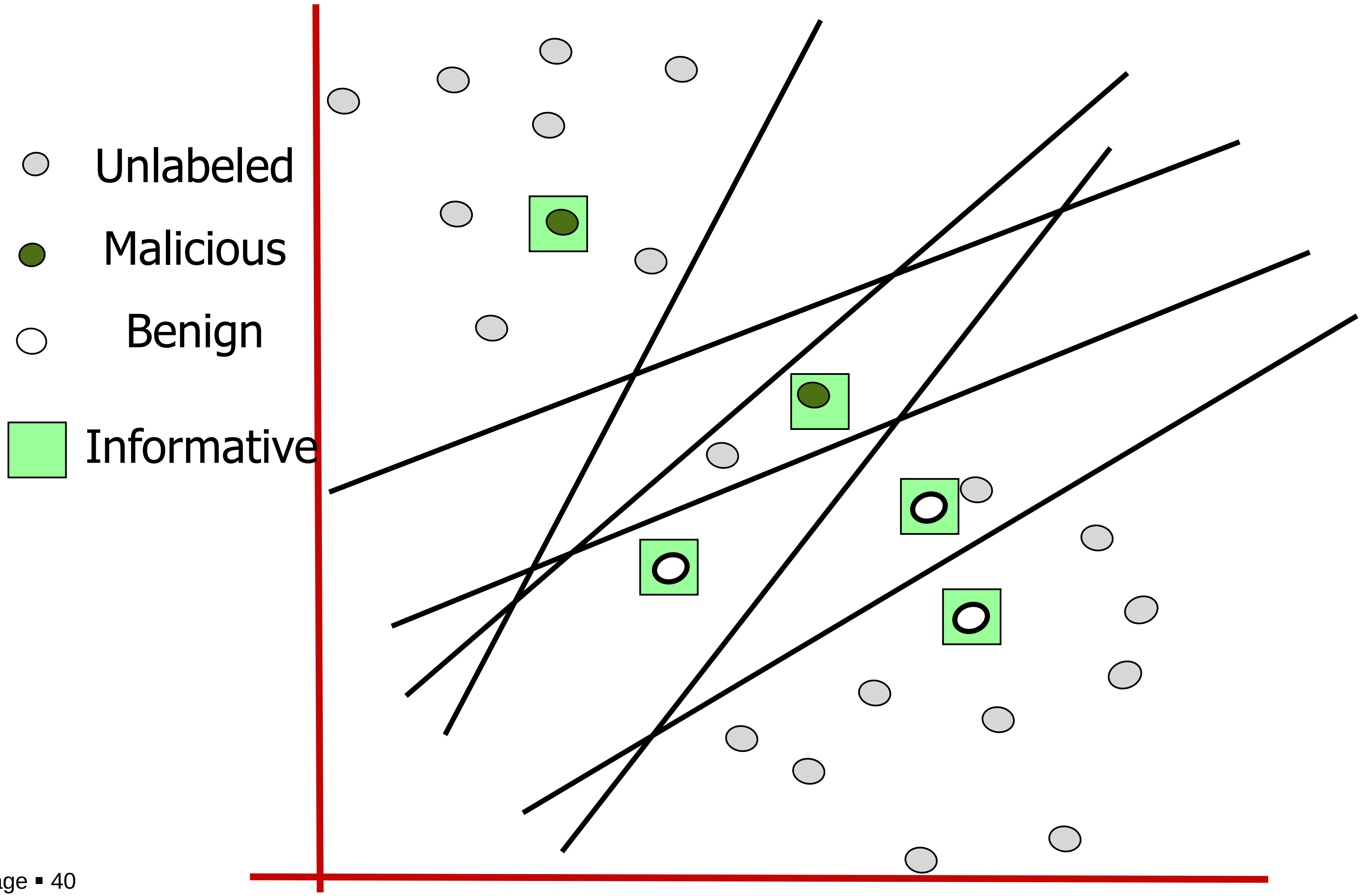
- Random Selection = Passive learning
  - New PDF files are randomly selected.
  - Files Might not be informative.
  - Won’t contribute the detection model’s capabilities and knowledge.
  - Waste of experts inspection efforts.

- Active Learning:
  - Efficient and intelligent selection of small yet informative set of new PDF files
  - Files that bear most of the new information and new attacks.
  - Improves the detection model’s accuracy and keeps it frequently updated
  - Reduction of experts inspection efforts.

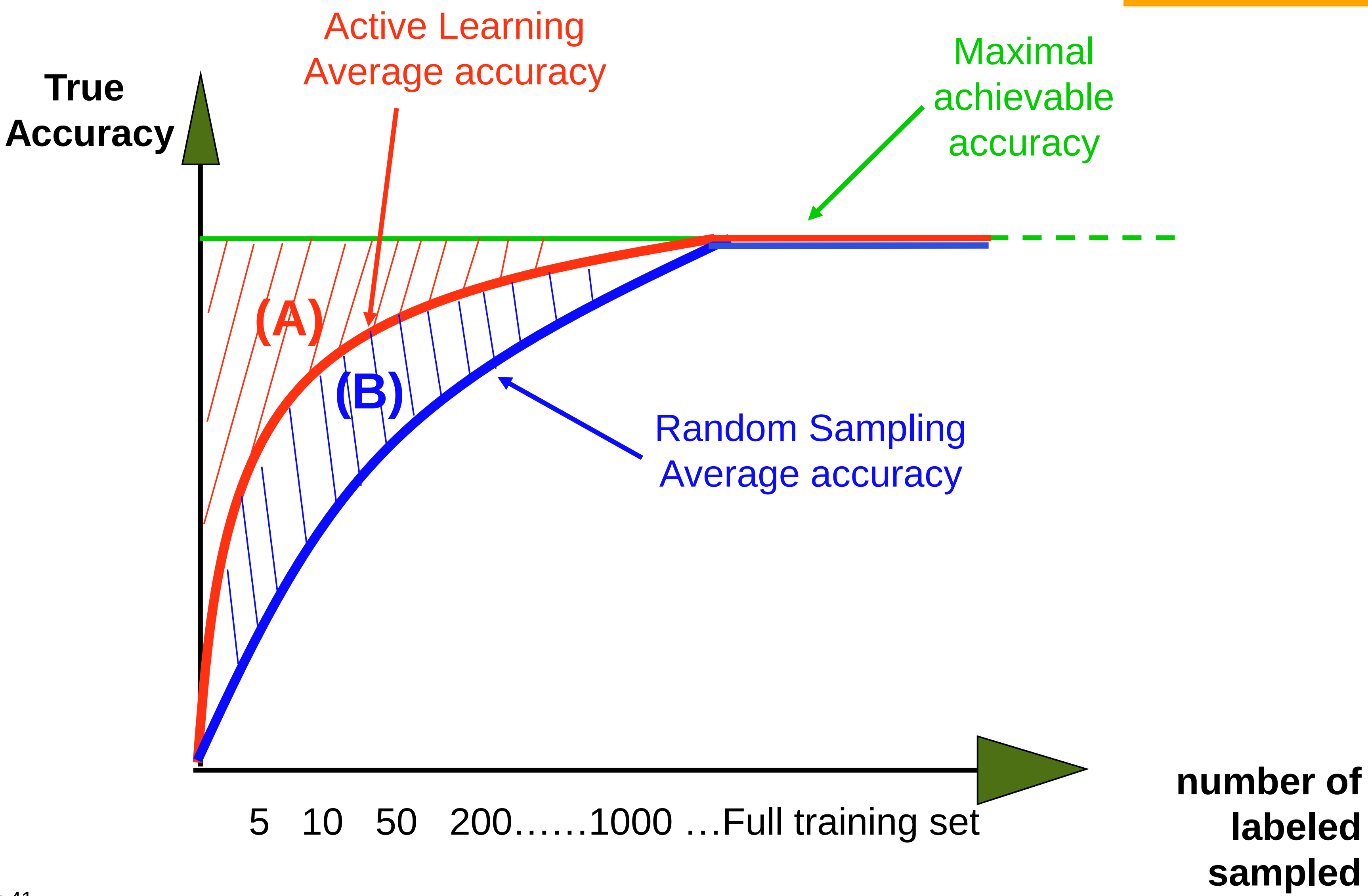
Random selection



Active Learning – Selective Sampling



Active learning – the advantage

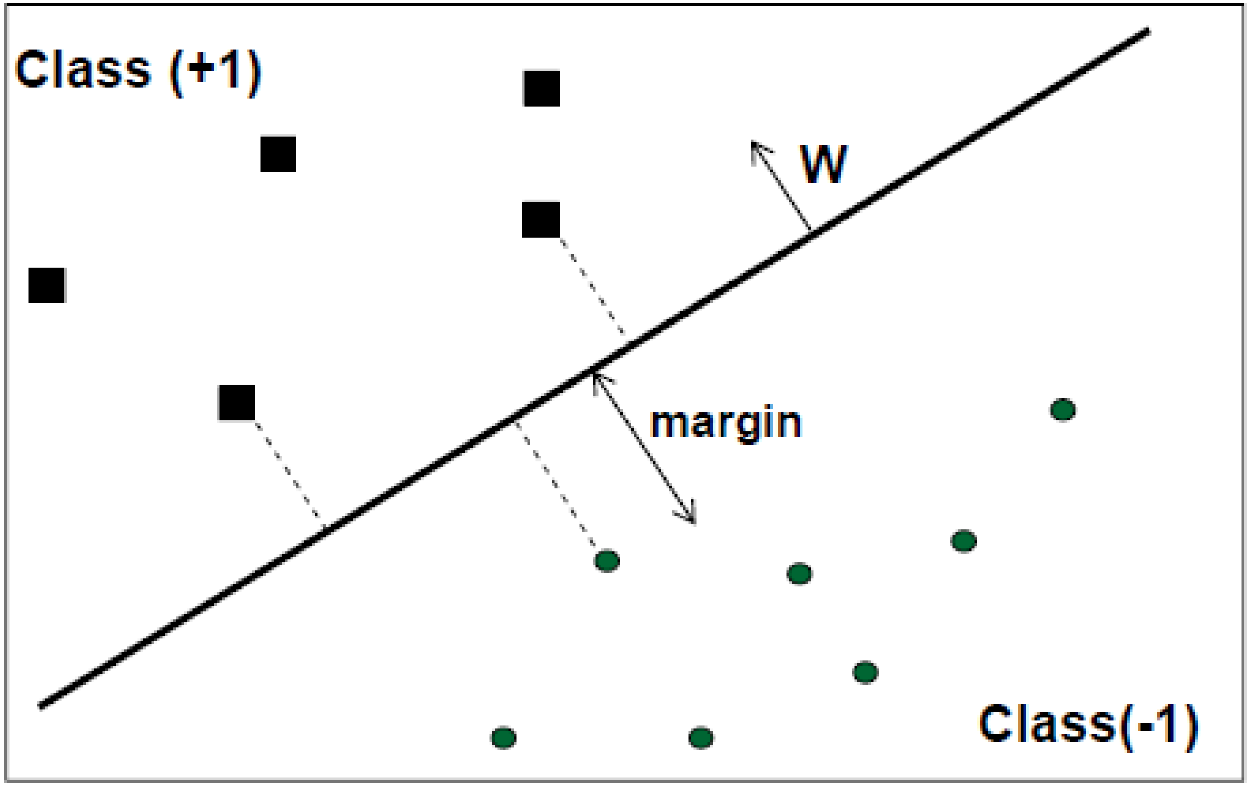


Active Learning Methods

Selective Sampling:

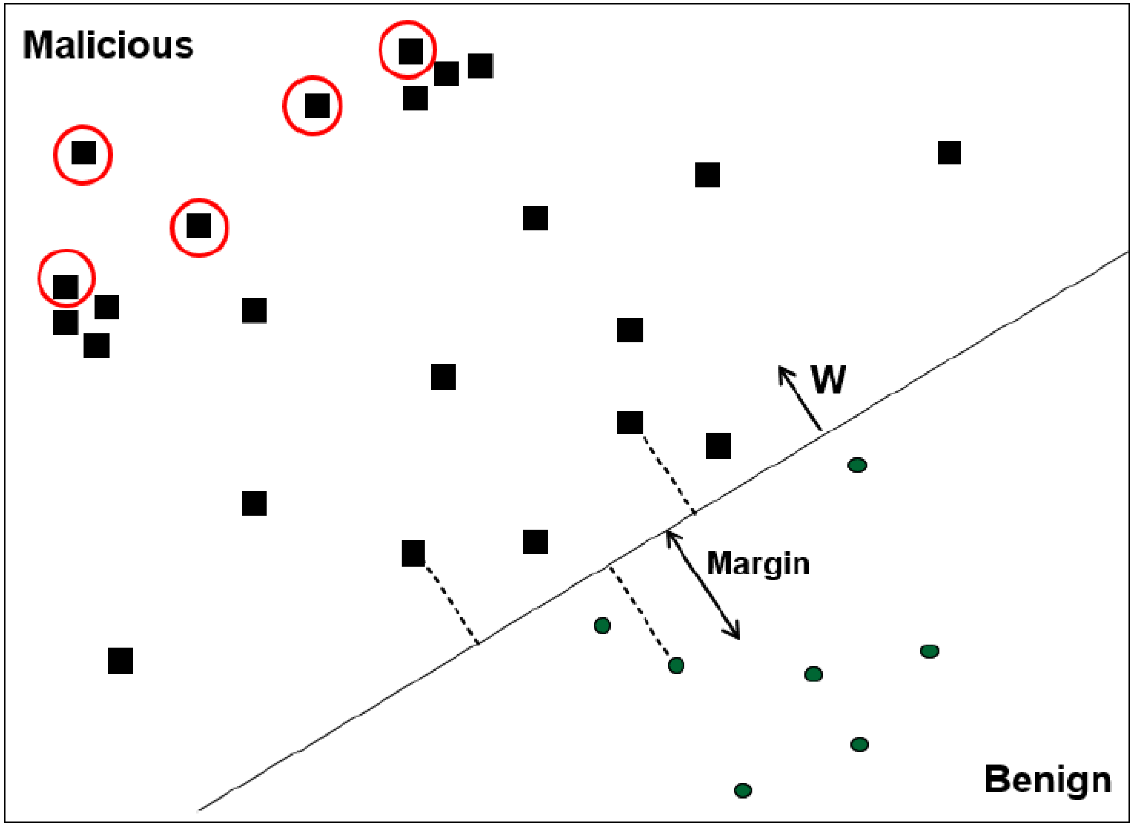
- SVM-Margin - Exploration
- Exploitation
- Combination

SVM-Margin - Exploration



- Select samples lies inside the SVM-Margin.
- Rough approximation for the minimizing the Version Space(VS).

Exploitation



- Select <u>representative</u> + <u>most probable malicious PDF files</u>.
- Selects also <u>confusing benign PDF files.</u>

TPR levels

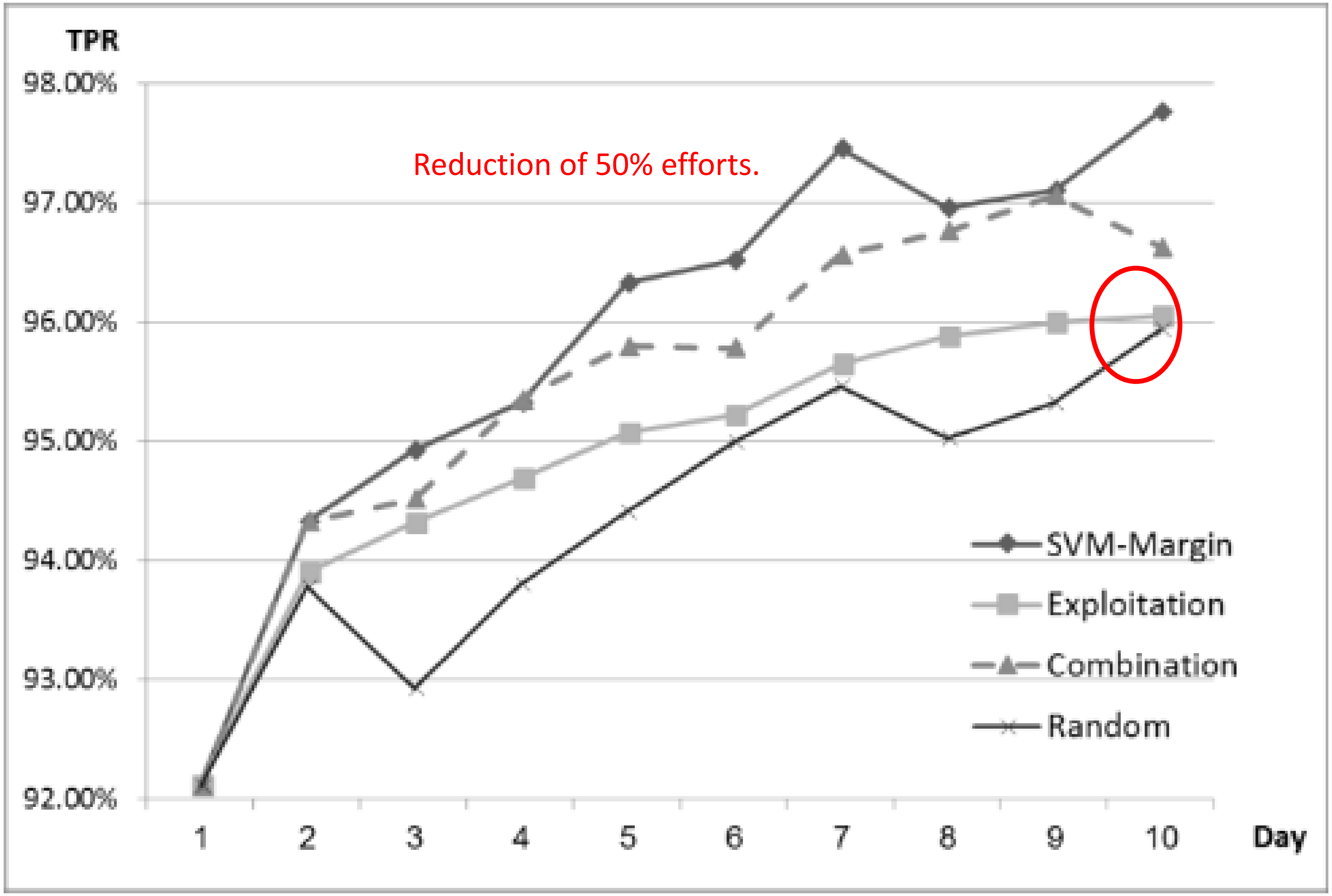


Figure 4: The TPR of the framework over the 10 days for different methods through the acquisition of 160 PDF files daily.

FPR levels

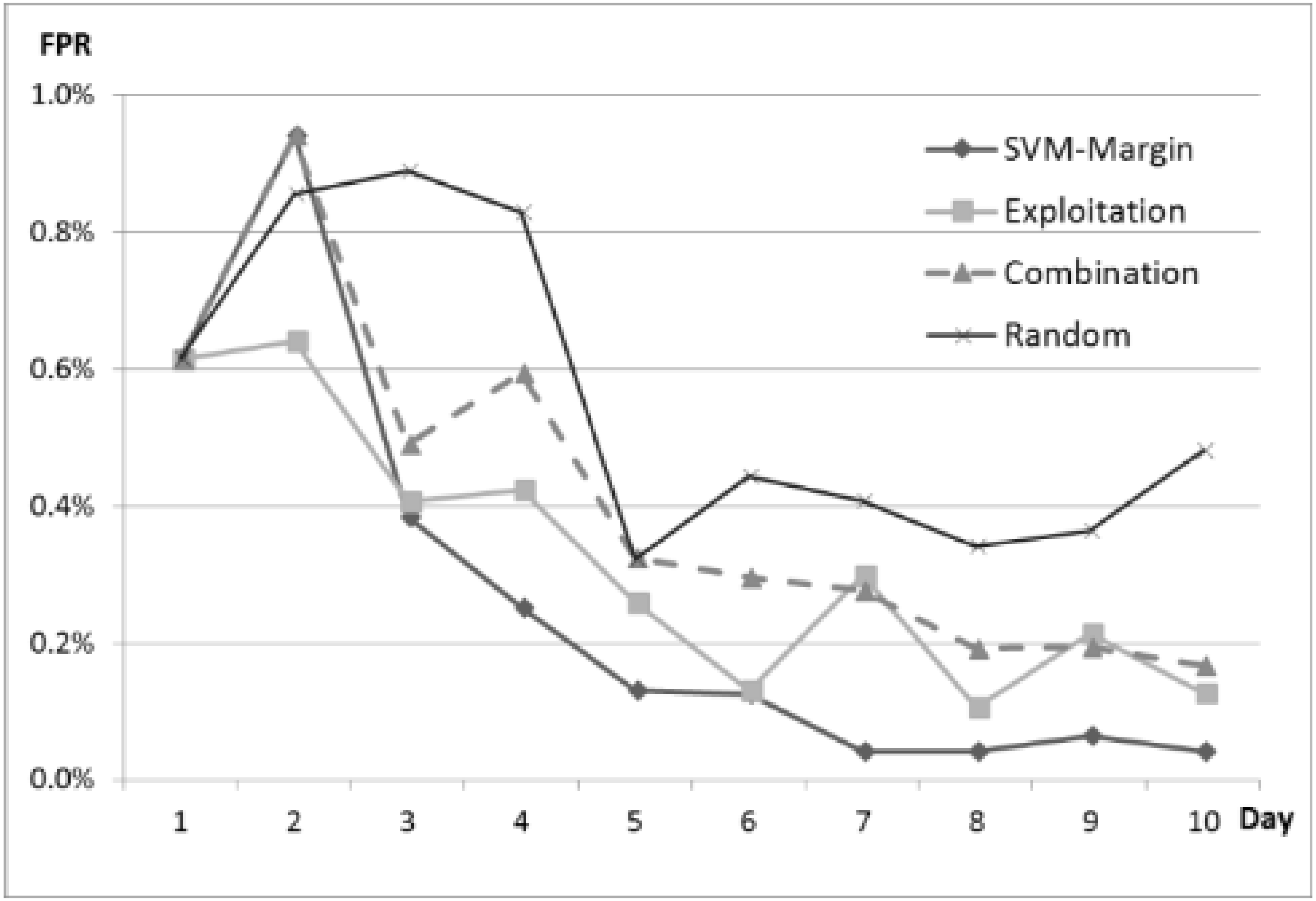
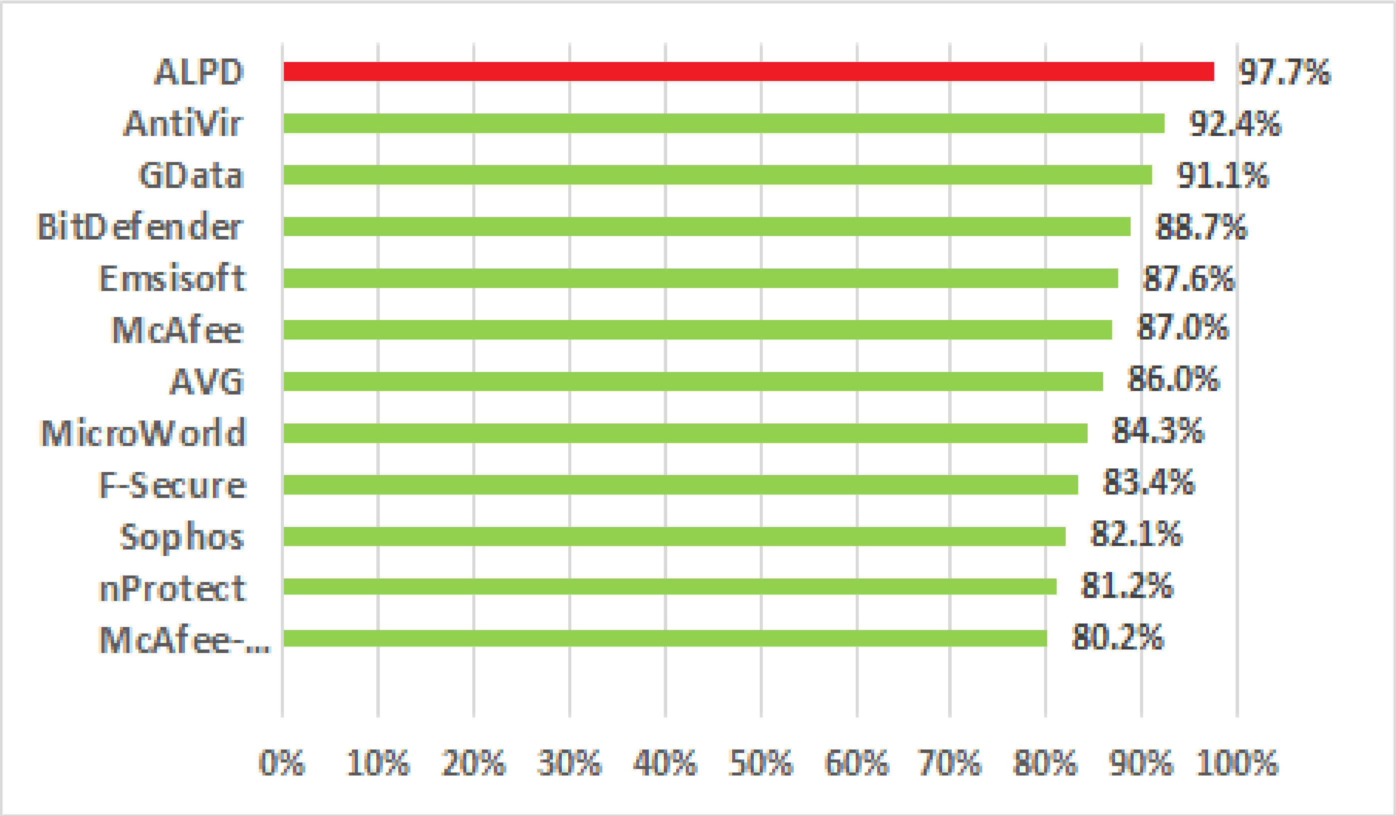


Figure 5: The FPR of the trends of the framework for different methods based on acquiring 160 PDF files daily.

Comparing to Anti-Virus Software - TPR



Smartphone Security

Risk to the user:

- **Privacy breach.**
- **Confidential information theft.**
- **Financial loss.**

Risks to the cellular infrastructure:

- **Coordinated DDoS attacks can shutdown the network using a relatively small set of malware instances.**
- **The malware can be dormant waiting for coordinated commands from the DDoS master.**

Smartphones' popularity and the number of available mobile applications has significantly grown. The number of mobile malware applications has increased correspondingly.

Dynamic Analysis for Malware Detection in Mobile Phones

- **<u>Android.Dropdialer Malware</u>**
- A *<u>self-updating</u>* capabilities.
  - Applications hosted on the Google Play Store were absolutely benign and did not contain any malware.
  - The malicious payload was downloaded from the Internet after the market application was installed on the device.
- The downloaded malicious package sent SMS messages to premium-rate numbers.
- Prompts to *<u>uninstall itself</u>* after sending out the premium SMS messages.

Asaf Shabtai, Lena Tenenboim-Chekina, Dudu Mimran, Lior Rokach, Bracha Shapira, Yuval Elovici, Mobile Malware Detection through Analysis of Deviations in Application Network Behavior, Computers & Security, Volume 43, June 2014, Pages 1–18

Our Approach – in brief

- Malware activities regularly affect the application's network behavior.
- **Can we detect the malware by solely monitor its network footprint?**
- Thus, we focus on monitoring applications network behavior and aim to detect unexplained changes any time they occur.

Monitoring Network Behavior → Learning Normal Patterns → Anomaly Detection

Utilized Features

	Feature	Brief Description
1	avg_sent_bytes	Represent the average amount of data sent or received by an application at the observed time interval (of 1 min.)
2	avg_rcvd_bytes	
3	avg_sent_pct	Represent the average portion of sent and received amount of data at the observed time interval (of 1 min.)
4	avg_rcvd_pct	
5	pct_avg_rcvd_bytes	Represents the portion of average received amount of data at the observed time interval (of 1 min.)
6	inner_ sent	Average time intervals between send\receive events occurring within the time interval of less than 30 seconds.
7	inner_ rcvd	
8	outer_ sent	Average time intervals between send\receive events occurring within the time interval above or equal to 30 seconds.
9	outer_ rcvd	

Feature Chains (FC)

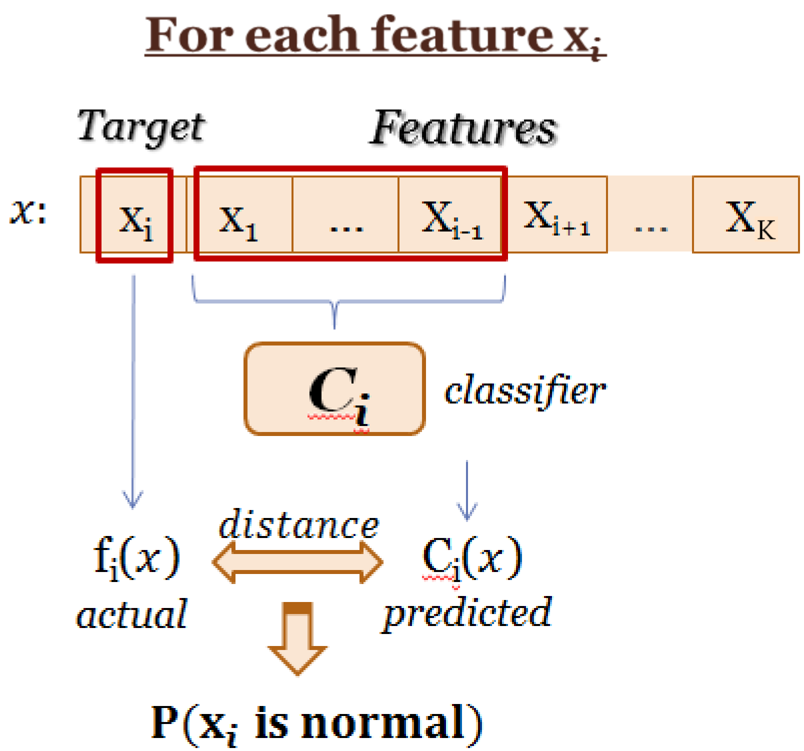
- *The idea*:
- A chain of models is trained on the feature space.
  1. Randomly sort the features in a chain.
  2. Learn a classifier for each one of the features using all previous features in the chain:
     $$C_i: \{f_1, \dots, f_{i-1}\} \rightarrow \{f_i\},$$
  3. Combine predictions:
     $$P(f_1, f_2, \dots, f_K) = \prod_{i=1}^{K} P(f_i | f_1, f_2, \dots, f_{i-1})$$
- Theoretically correct (applying Bayes rule):

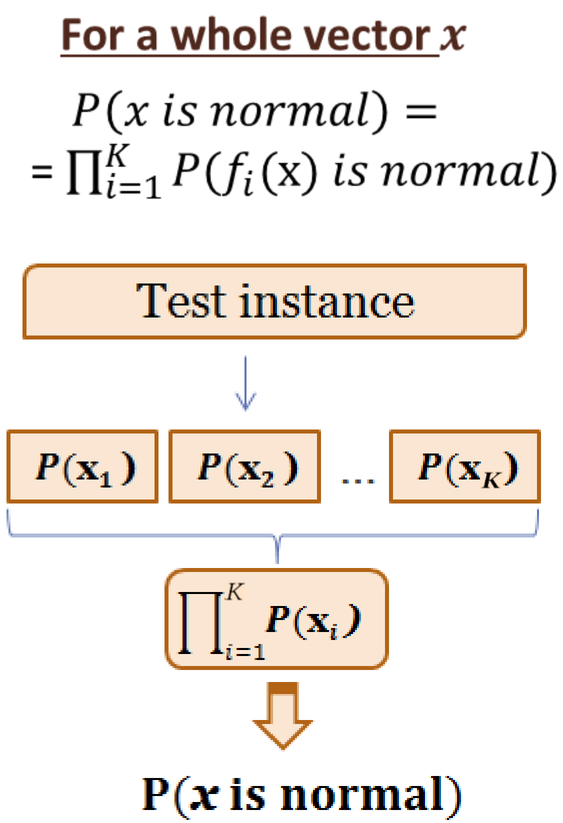
$$P(f_1) * P(f_2|f_1) * P(f_3|f_1, f_2) * \cdots * P(f_L|f_1, f_2, \dots, f_{K-1}) =$$

$$= \cancel{P(f_1)} * \frac{\cancel{P(f_2, f_1)}}{\cancel{P(f_1)}} * \frac{\cancel{P(f_3, f_2, f_1)}}{\cancel{P(f_2, f_1)}} * \cdots * \frac{P(f_K, f_{K-1}, \dots, f_1)}{\cancel{P(f_{K-1}, \dots, f_1)}}$$

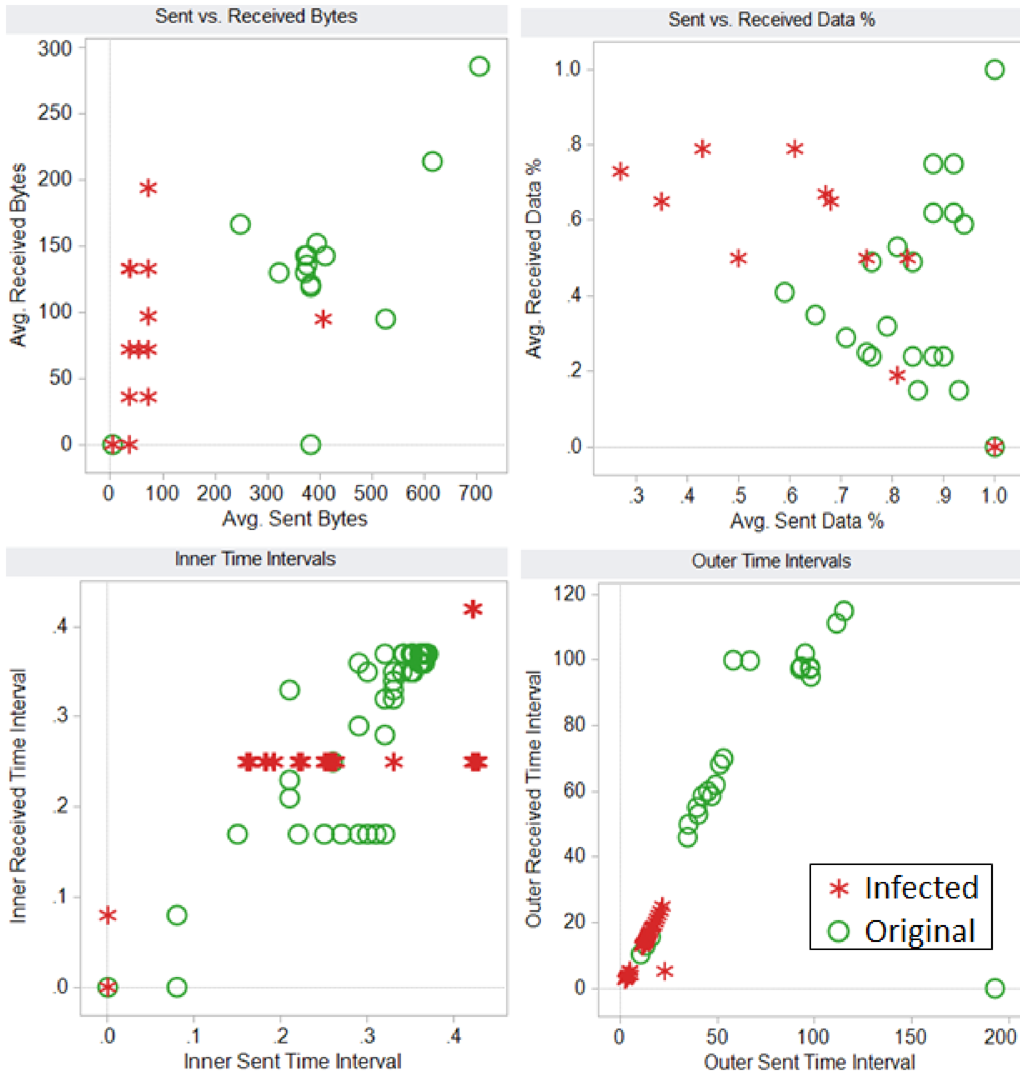
Feature Chains – detection

- For evaluating each new instance x:



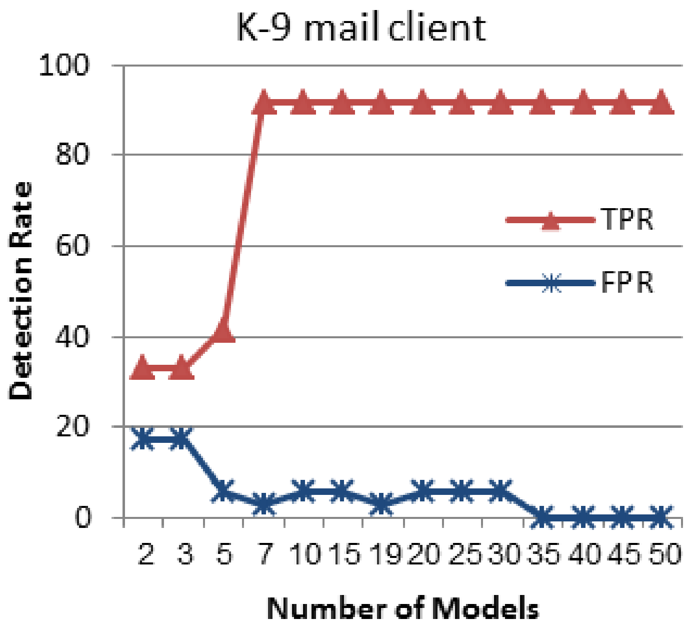


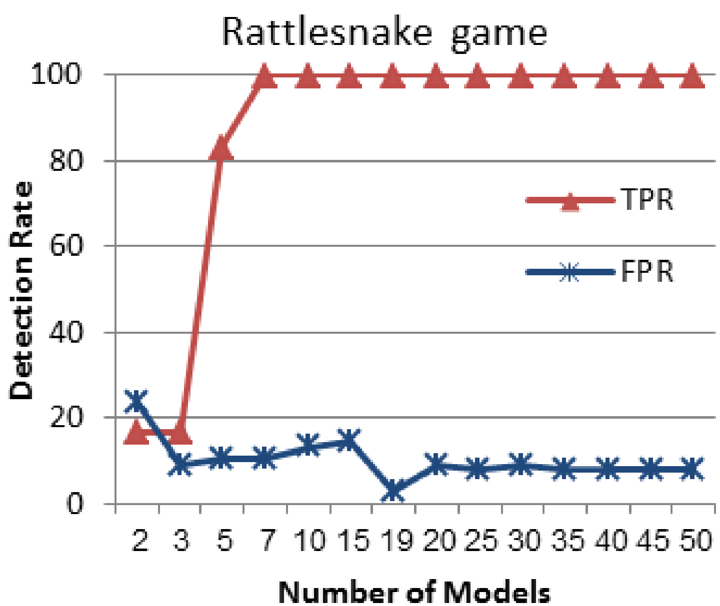
New Malware on Android Market – some results



Results

– Evaluating EFC performance with respect to the number of ensemble models.





- High and stable TPR is achieved at relatively low number of models, $m \geq 7$.
- larger number of models leads to lower FPR
- for achieving a stable low FPR - a larger number of models, regularly $m \geq 30$, is needed.

Malware detection using network traffic analysis

- Employ machine learning techniques to model user normal network access and detect tiny anomalies
- Based on anomalies and known malicious activity patterns detect APTs and C&C servers
- Improve detection algorithm performance for integration in real time network traffic analysis systems. (IDS, IPS and etc.)

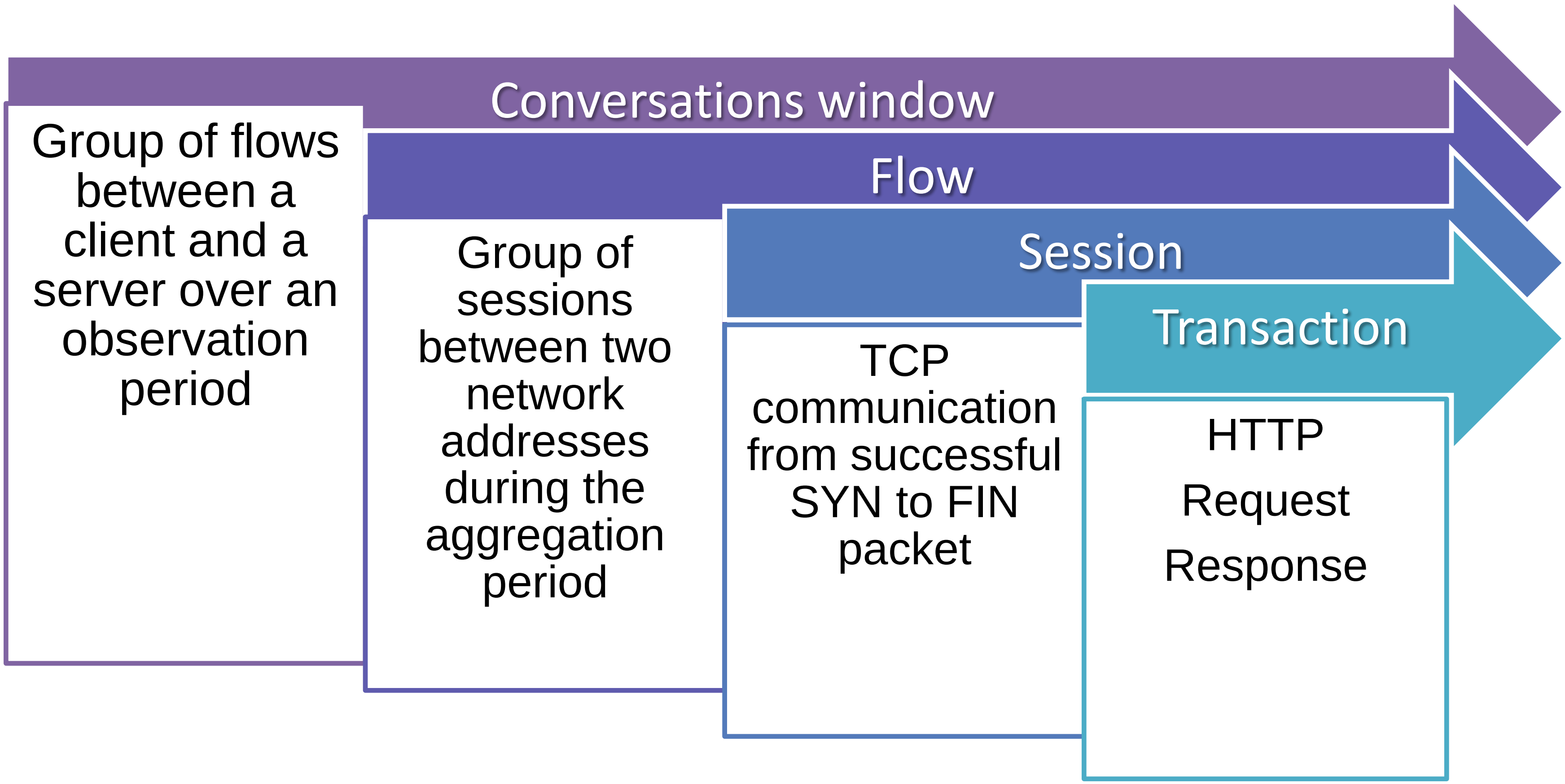
Dmitri Bekerman, Bracha Shapira, Lior Rokach, Ariel Bar, Unknown Malware Detection Using Network Traffic Classification, IEEE CNS (Communications and Network Security), 28-30 September Florence, Italy 2015,

Feature Engineering

Examples

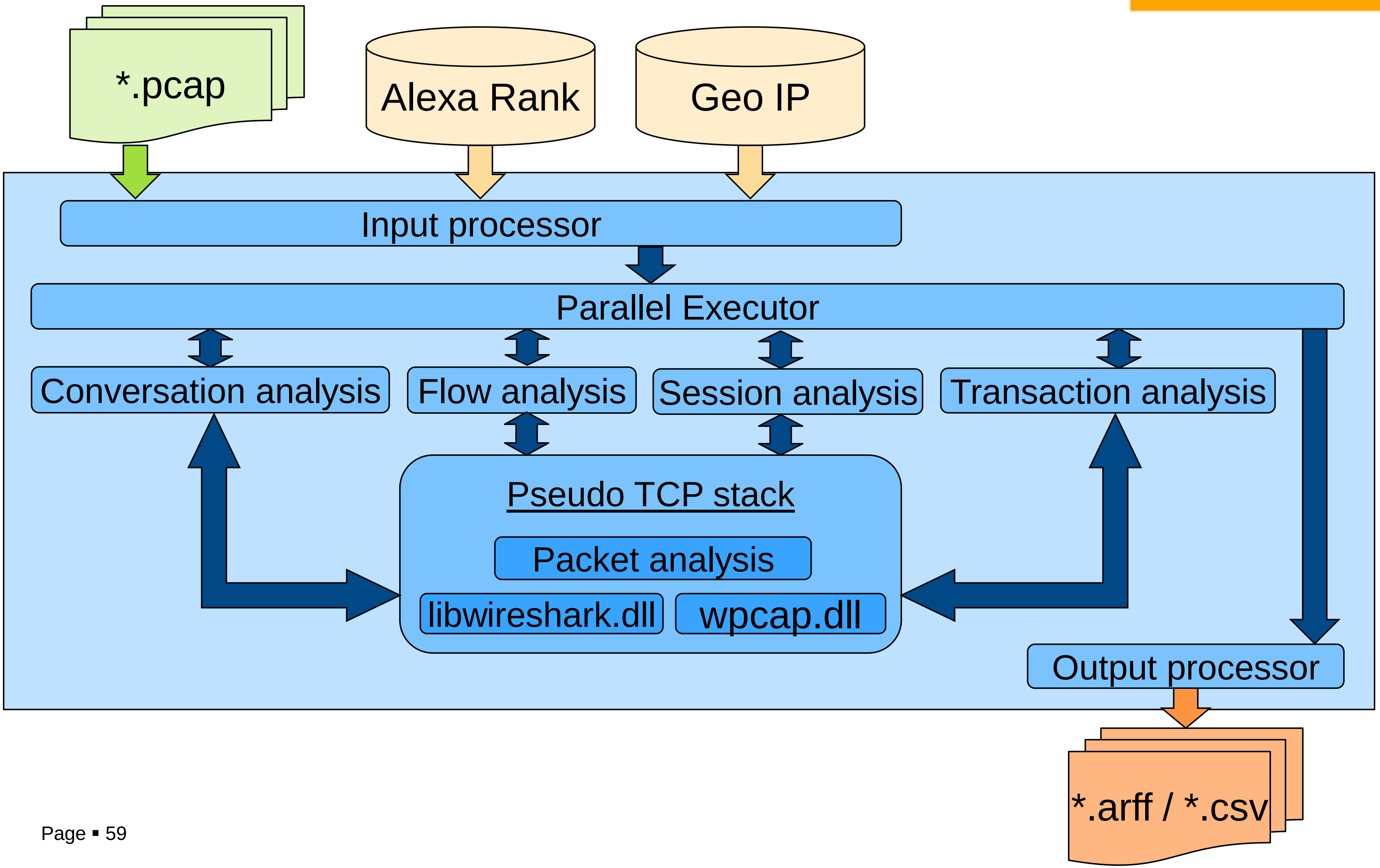
- DNS query address Alexa 1M ranking
- DNS query address exist or not
- HTTP hostname zone
- HTTPS/SSL certificate
- Flow daytime
- Packets inter-arrival time
- Total number of ACKs
- Count of out-of-order packets

Feature Engineering



<u>≈ 920 unique features at different network layers</u>

Feature Extractor



Evaluation Procedure

Data Set

- ≈ 8000 from academic malicious bank sandbox
- ≈ 2500 from Verint© sandbox
- ≈ 4500 from public available sandboxes in web
- Benign and malicious data captured by Verint© from corporate networks

Goal

- Train a model on network traffic from environment A and employ it on network traffic from environment B.

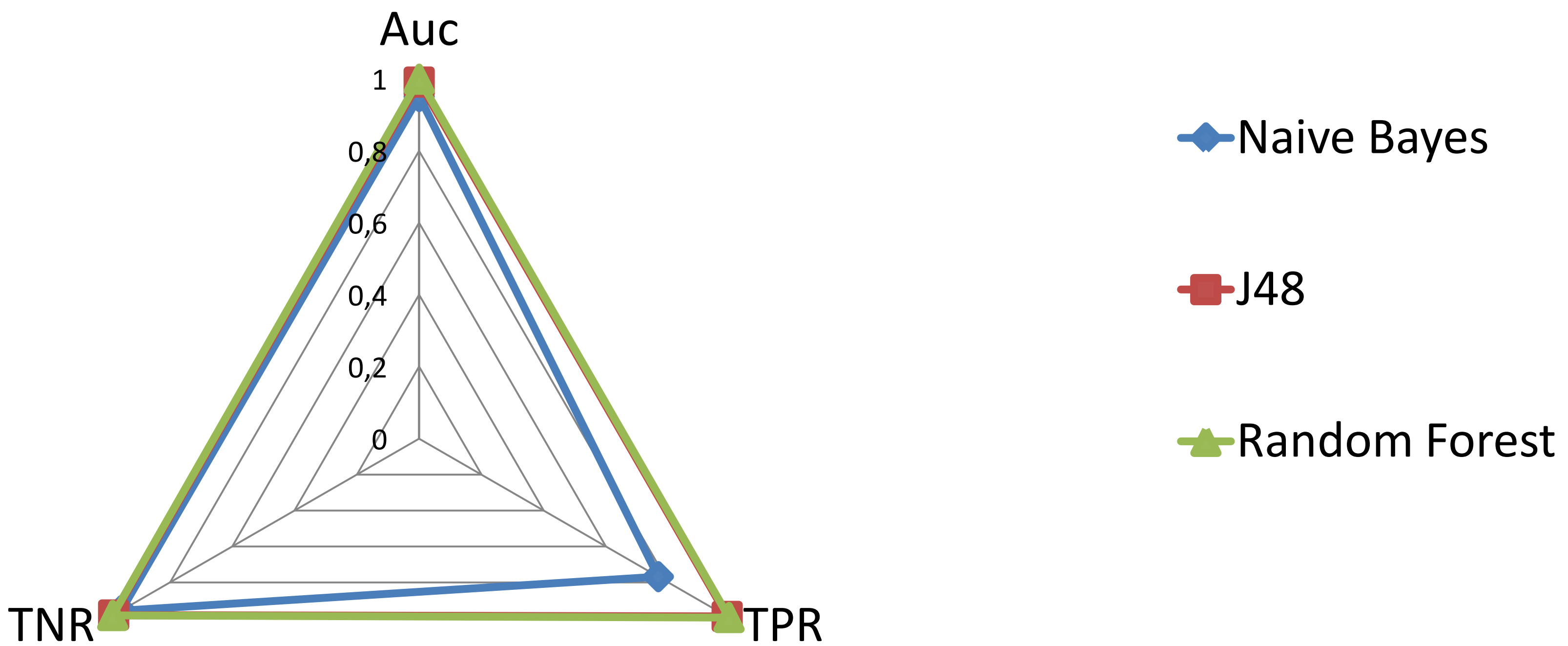
Top 10 Features

- cw_count_flows numeric
- cw_dns_good_tcp_sess_ratio numeric
- cw_tcp_analysis_duplicate_ack numeric
- cw_tcp_analysis_keep_alive numeric
- flow_ack_A numeric
- flow_dns_alexaRank numeric
- flow_dns_count_addresses numeric
- flow_dns_count_answer_records numeric
- flow_http_inter_arrivel_median numeric
- session_reset numeric

10 cross validation on real network

Based on 35 features selected by CFS algorithm

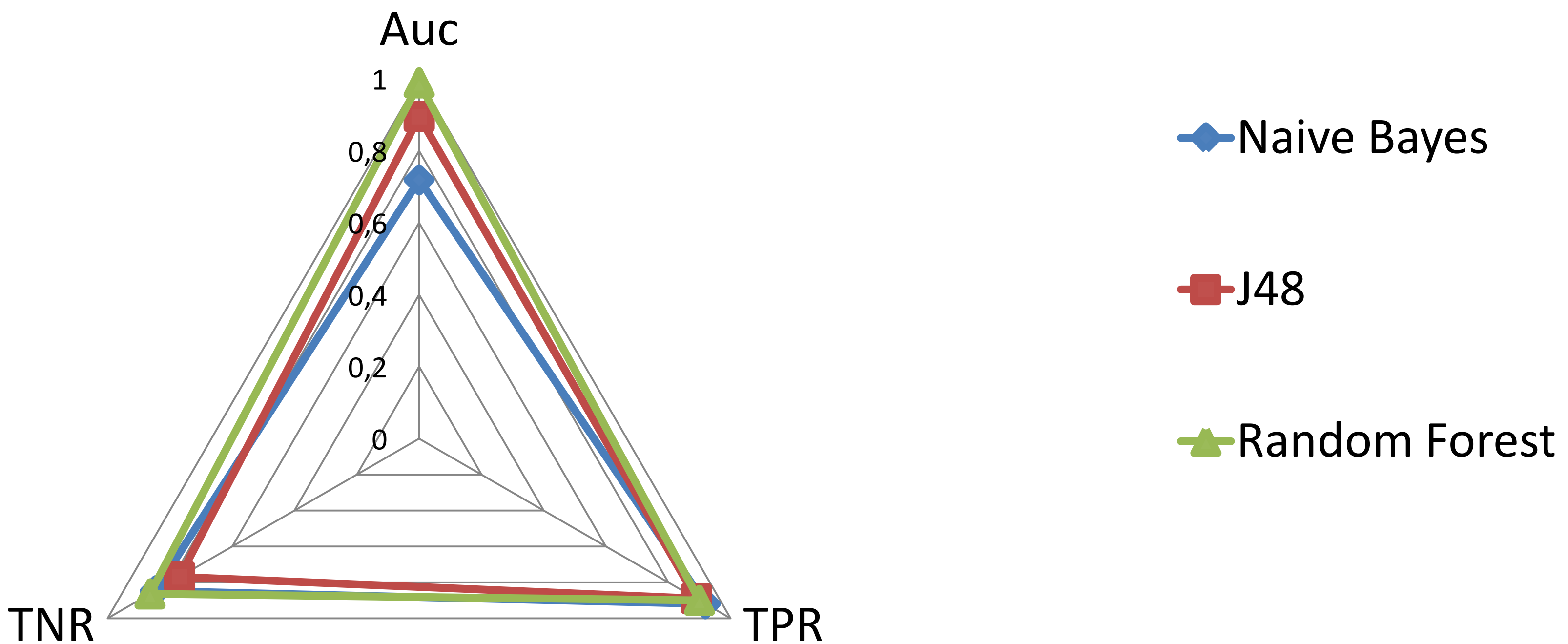
	TPR	FPR	AUC
Naïve Bayes	0.768	0.043	0.951
J48	0.989	0.019	0.991
Random Forest	**0.995**	**0.016**	**0.999**



Leave one malware family out (Unseen Family)

Based on 58 features selected by CFS algorithm

	TPR	FPR	AUC
Naïve Bayes	0.919	0.153	0.719
J48	0.89	0.231	0.895
Random Forest	**0.9**	**0.136**	**0.989**



Insider Threat

What is Insider Threat

“Malicious insider threat to an organization is a **current or former employee**, **contractor**, or other **business partner** who has or had authorized access to an organization’s network, system, or data and intentionally exceeded or misused that access in a manner that negatively affected the **confidentiality**, **integrity**, or **availability** of the organization’s information or information systems. In addition, **insider threats can also be unintentional** (non-malicious).”

(From the CERT Division of the Software

Engineering Institute (SEI), CMU.)

23% of the cyber-security events, recorded in a 12-month period, were caused by insiders (2015 Cyber Security Watch Survey)

Examples from the News

- Government:
  - Edward Snowden, NSA contractor, leaked classified info on NSA's PRISM project.
  - NSA failed to detect his activities.
  - Edward Snowden had administrator privileges.
- Industry:
  - "Ofcom data breach highlights insider threat," "UK communications regulator Ofcom has revealed that a former employee offered stolen – commercially sensitive – information to his new employer, highlighting the insider threat."

    ComputerWeekly.com, 11 Mar 2016 13:30.

Ofcom

Independent regulator and competition authority for the UK communications industries.

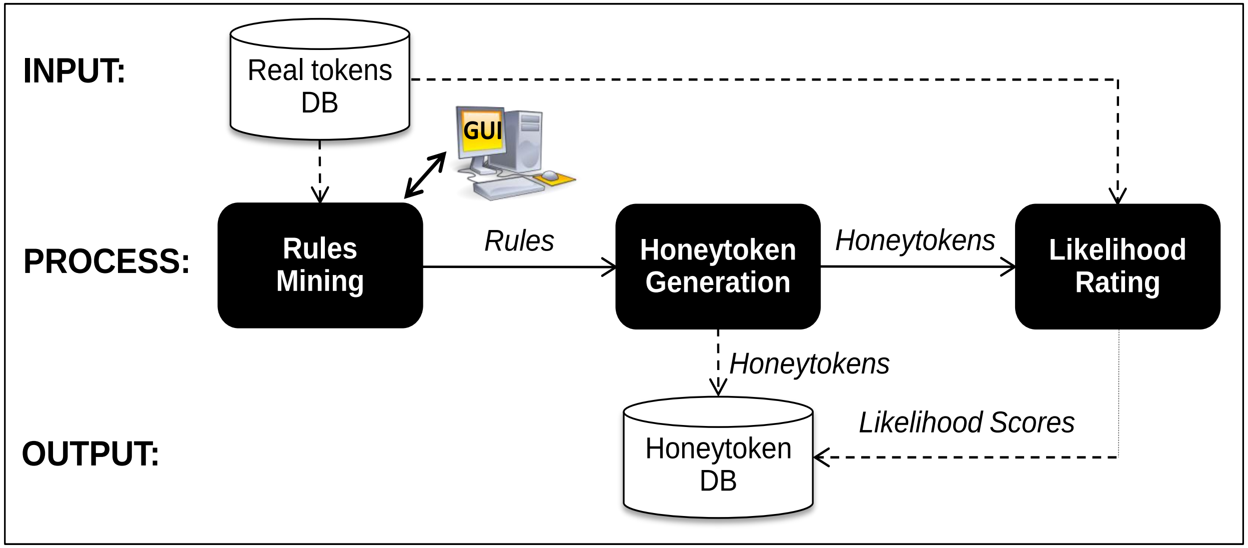
Using Honeytokens for Insider Detection

- A honeytoken is a fabricated data item that may indicate the presence of malicious activity in a computer system.
- Honeytokens can be used to detect insiders, mainly when they are more attractive for misuse than typical data items, for example, a fake dormant account.

Asaf Shabtai, Maya Bercovitch, Lior Rokach, Ya'akov (Kobi) Gal, Yuval Elovici, Erez Shmueli: Behavioral Study of Users When Interacting with Active Honeytokens. ACM Trans. Inf. Syst. Secur. 18(3): 9 (2016)

Using Honeytokens for Insider Detection

- Challenge: A good honeytoken is an artificial data item that is hard to distinguish between real tokens and the honeytoken
- We developed and used **HoneyGen** - a generic framework for automatically creating **high-quality** honeytokens for **any** database.



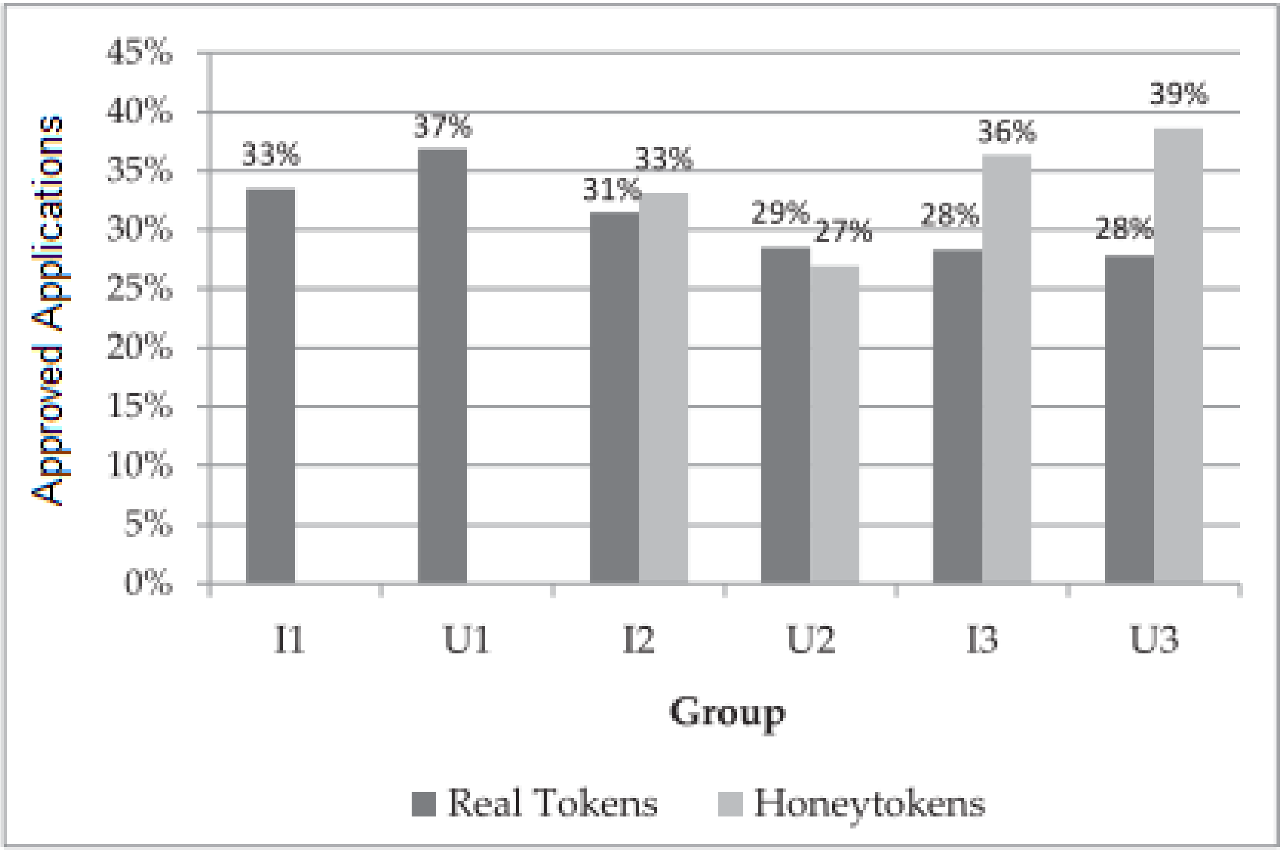
68

Behavioral Study

- 173 participants in a financial case-study
- The participants were divided into six groups, based on two factors:
  - informed/uninformed about the use of honeytokens
  - percentage of honeytokens being used

Participant type (count)	No honeytokens	10% honeytokens	20% honeytokens	Total
Informed about the use of honeytokens	I1 (31)	I2 (29)	I3 (30)	90
Uninformed about the use of honeytokens	U1 (27)	U2 (28)	U3 (28)	83
Total	58	57	58	173

Results



45%
40%
35%
30%
25%
20%
15%
10%
5%
0%
Approved Applications
33%
37%
31%
33%
29%
27%
28%
36%
28%
39%
I1
U1
I2
U2
I3
U3
Group
Real Tokens
Honeytokens

Using Honeytokens for Insider Detection

- The **detection rate** when the list contained **20% honeytokens was 100%** for both *I3* and *U3*.
- The **detection rate** of participants with lists containing **20% honeytokens** was **higher** than that of participants with lists containing **10%** honeytokens.
- We also examined whether the **number of honeytokens** used (10% or 20%) had a **significant** effect on detection and found this effect to be statistically significant (X-square= 9.8927, p= 0.001659).

M-Score: Misuseability Weight

- A new measure to estimate the level of harm that might be caused when the data is leaked or misused.
- M-score is the misuseability weight measure for tabular data
  - Quality of the information - the importance of the information
  - Quantity of the information - the amount of the information
  - The distinguishing factor - the amount of efforts required in order to discover the specific entities that the table refers to

F Name	L Name	City	Account Type
Anton	Richter	Berlin	Gold
Otto	Hecht	Bonn	Gold
Hedy	Gruber	Berlin	Bronze
Mirjam	Fried	Berlin	White

Amir Harel, Asaf Shabtai, Lior Rokach, Yuval Elovici: M-Score: A Misuseability Weight Measure. IEEE Trans. Dependable Sec. Comput. 9(3): 414-428 (2012)

Misuse detection in databases

- The “quality” function

Customer Group –

Business = 0.8	Private = 0

Average Monthly Bill –

More then 700$ = 1	500$ - 699$ = 0.8	350$ - 499$ = 0.5	Less then 350$ = 0.1

Account Type –

Gold = 1	Silver = 0.7	Bronze = 0.3	White = 0.1

Contract Expiration Date (in days) –

0 or less = 1	1-30 days = 0.8	31-180 days = 0.5
181-365 days = 0.1	More then 365 days = 0	

Main Usage -

Phonecalls = 1	SMS = 0.7	Data = 0.3	Paid services = 0.1

Misuse detection in databases

- Raw Record Score

$$RRS_i = min\left(1, \sum_{S_j \in T} f(c, S_j[x_i])\right)$$

(A) THE SOURCE TABLE

Job	City	Sex	Account Type	Average Monthly Bill
Lawyer	NY	Female	Gold	$350
Gardener	LA	Male	White	$160
Gardener	LA	Female	Silver	$200
Lawyer	NY	Female	Bronze	$600
Teacher	DC	Female	Silver	$300
Gardener	LA	Male	Bronze	$200
Teacher	DC	Female	Gold	$875
Programmer	DC	Male	White	$20
Teacher	DC	Female	White	$160

(B) THE PUBLISHED TABLE

Job	City	Sex	Account Type	Average Monthly Bill
Lawyer	NY	Female	Gold	$350
Lawyer	NY	Female	Bronze	$600
Teacher	DC	Female	Silver	$300
Gardener	LA	Male	Bronze	$200
Programmer	DC	Male	White	$20
Teacher	DC	Female	White	$160

RRS_1 = *min*(1,1+0.5)=1

f(Account Type[Gold])=1 and *f*(Average Monthly Bill[$350])=0.5

Misuse detection in databases

- Distinguishing factor

(A) THE SOURCE TABLE

Job	City	Sex	Account Type	Average Monthly Bill
Lawyer	NY	Female	Gold	\$350
Gardener	LA	Male	White	\$160
Gardener	LA	Female	Silver	\$200
Lawyer	NY	Female	Bronze	\$600
Teacher	DC	Female	Silver	\$300
Gardener	LA	Male	Bronze	\$200
Teacher	DC	Female	Gold	\$875
Programmer	DC	Male	White	\$20
Teacher	DC	Female	White	\$160

(B) THE PUBLISHED TABLE

Job	City	Sex	Account Type	Average Monthly Bill
Lawyer	NY	Female	Gold	\$350
Lawyer	NY	Female	Bronze	\$600
Teacher	DC	Female	Silver	\$300
Gardener	LA	Male	Bronze	\$200
Programmer	DC	Male	White	\$20
Teacher	DC	Female	White	\$160

D_1 = 2 since the tuple {*Lawyer*, *NY*, *Female*} appears twice in Table A

Misuse detection in databases

- Final Record Score

$$RS = \max_{0 \le i \le r}(RS_i) = \max_{0 \le i \le r}\left(\frac{RRS_i}{D_i}\right)$$

(A) THE SOURCE TABLE

Job	City	Sex	Account Type	Average Monthly Bill
Lawyer	NY	Female	Gold	$350
Gardener	LA	Male	White	$160
Gardener	LA	Female	Silver	$200
Lawyer	NY	Female	Bronze	$600
Teacher	DC	Female	Silver	$300
Gardener	LA	Male	Bronze	$200
Teacher	DC	Female	Gold	$875
Programmer	DC	Male	White	$20
Teacher	DC	Female	White	$160

(B) THE PUBLISHED TABLE

Job	City	Sex	Account Type	Average Monthly Bill
Lawyer	NY	Female	Gold	$350
Lawyer	NY	Female	Bronze	$600
Teacher	DC	Female	Silver	$300
Gardener	LA	Male	Bronze	$200
Programmer	DC	Male	White	$20
Teacher	DC	Female	White	$160

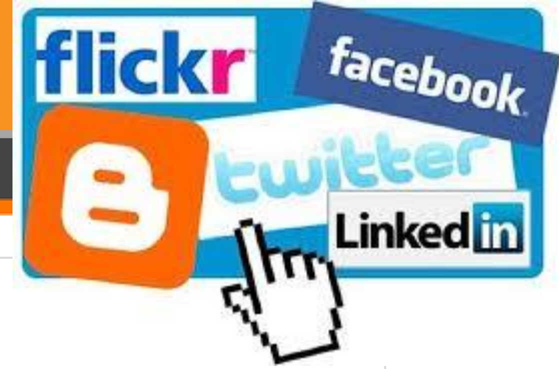
$$RS(1b) = \max\left(\frac{1}{2}, \frac{1}{2}, \frac{0.8}{3}, \frac{0.4}{2}, \frac{0.2}{1}, \frac{0.2}{3}\right) = \frac{1}{2}$$

Misuse detection in databases

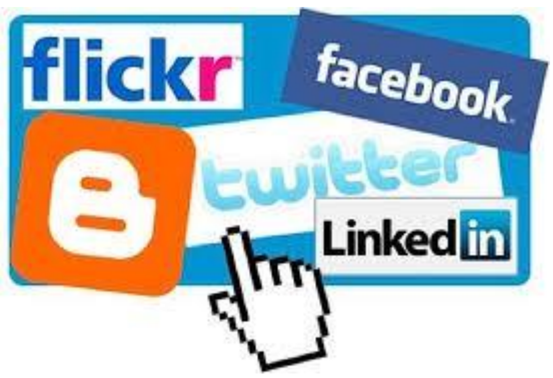
- The MScore
  - *r* - number of records
  - *x* – tradeoff parameter between the size of the data and quality of the data

$$\boldsymbol{MScore} = r^{1/x} \times RS = r^{1/x} \times \max_{0 \le i \le r} \left(\frac{RRS_i}{D_i} \right)$$

Social Networks Security Impact



National Security



Business Security

Individual Security

The Risk

- Researches shows that 36% of the personal information is shared with all 1 billion Facebooks users.
- 26% of the children studied in an European study had their online social network's profile set to "public".
- Currently a huge amount of information can be extracted by many different attacks like phishing, hacking, data mining etc.

Social Networks Security

Tens of Millions of Fake Profiles

Facebook estimates that 5%-6% of profiles in their social network are fake or duplicate profiles

Fake Profiles Identification

It is hard to distinguish fake profiles from real profiles

In some cases fake profiles clone real profiles.

.

Our Solution

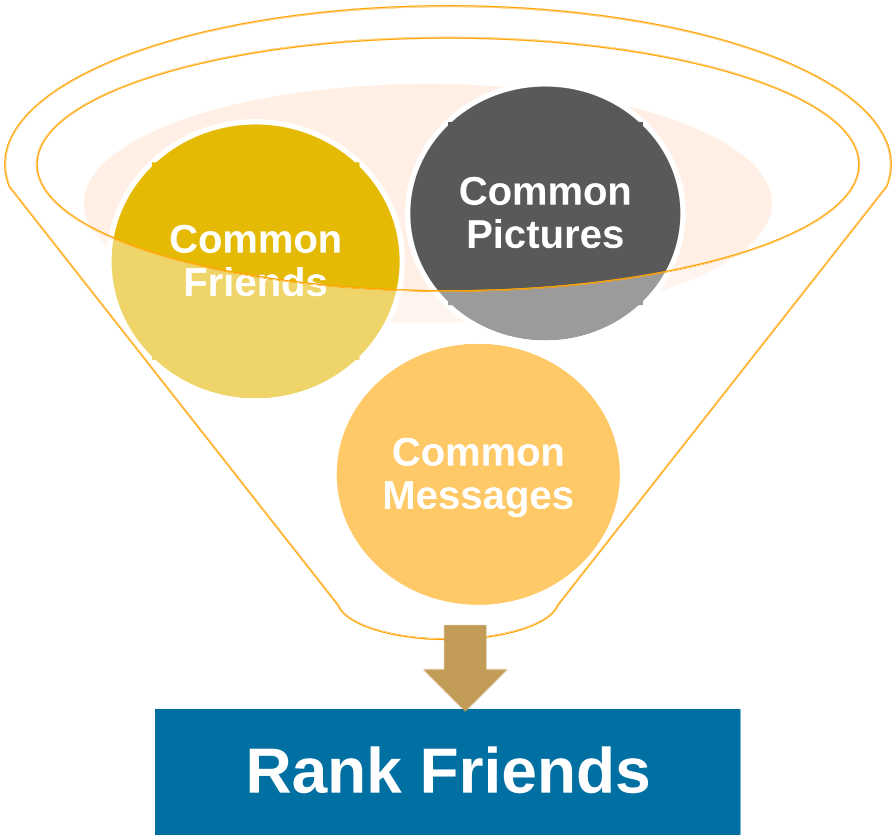
Social Privacy Protector for individuals

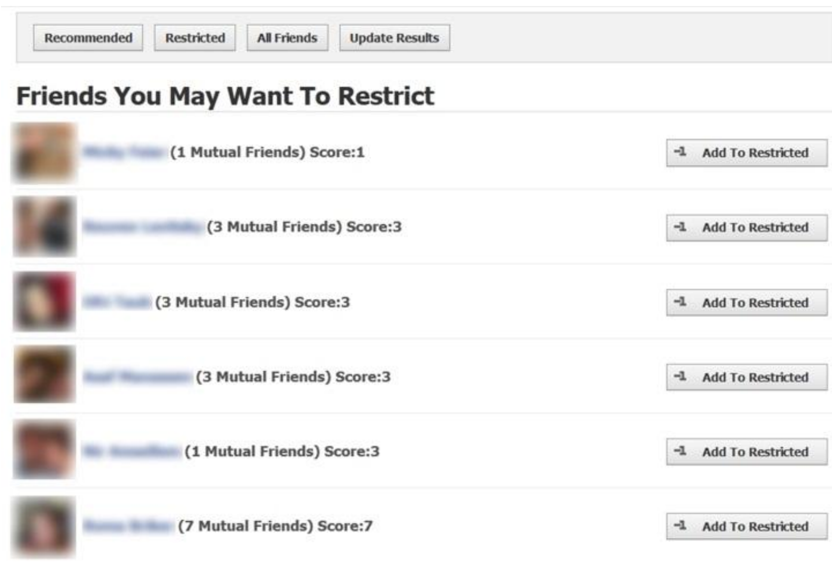
Recommend users to disconnect from other users.

Social Intrusion Detection For operators

Social Networks Security – Privacy Protector

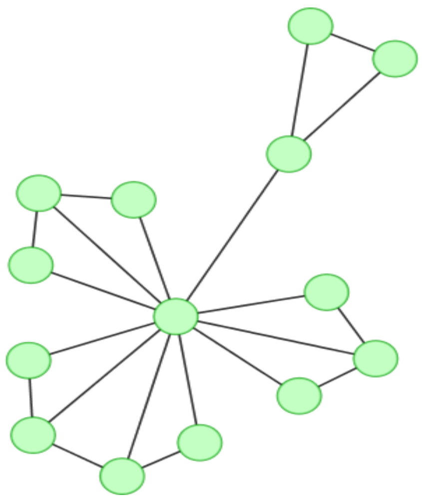
Protect social networks users' privacy by recommending removal of fake friends

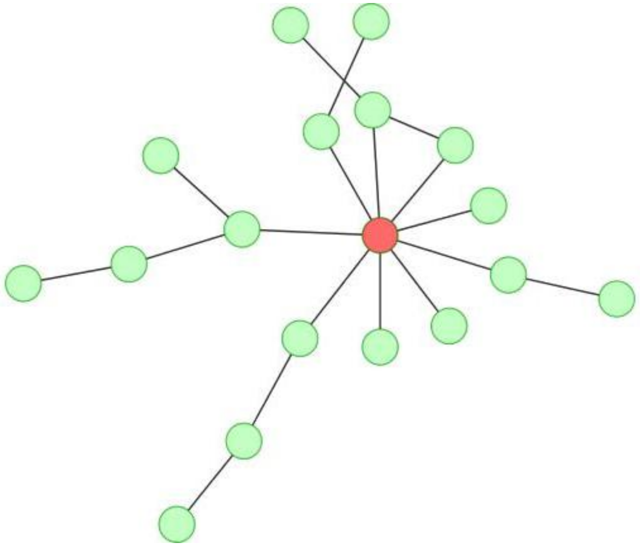




Identify Faked Profiles:
Communities Connection Anomaly

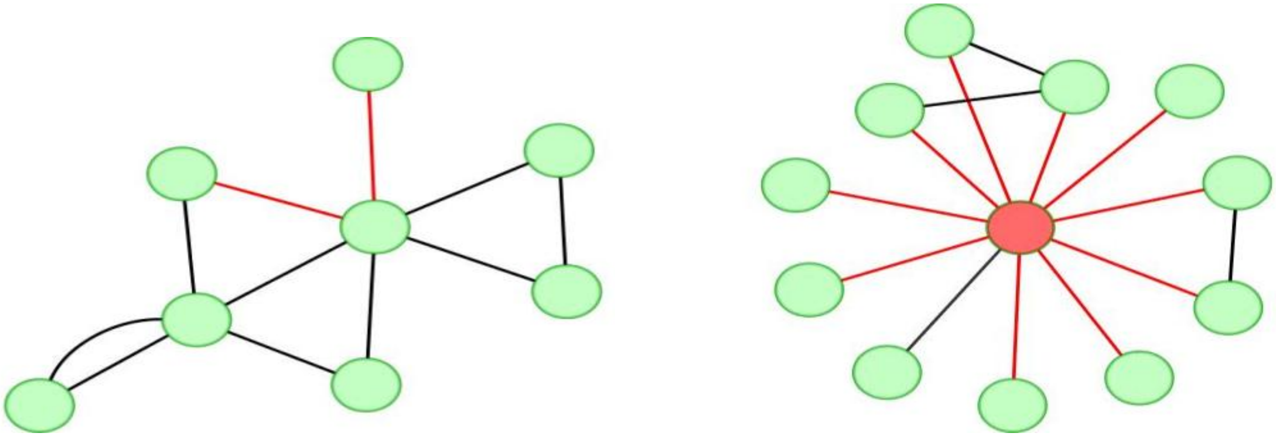
- Fake Profiles may look real but their social structure is usually different from real profiles.
- Fake Profiles tend to collect random users and connect to several communities.





Identify faked Profiles by Link Prediction

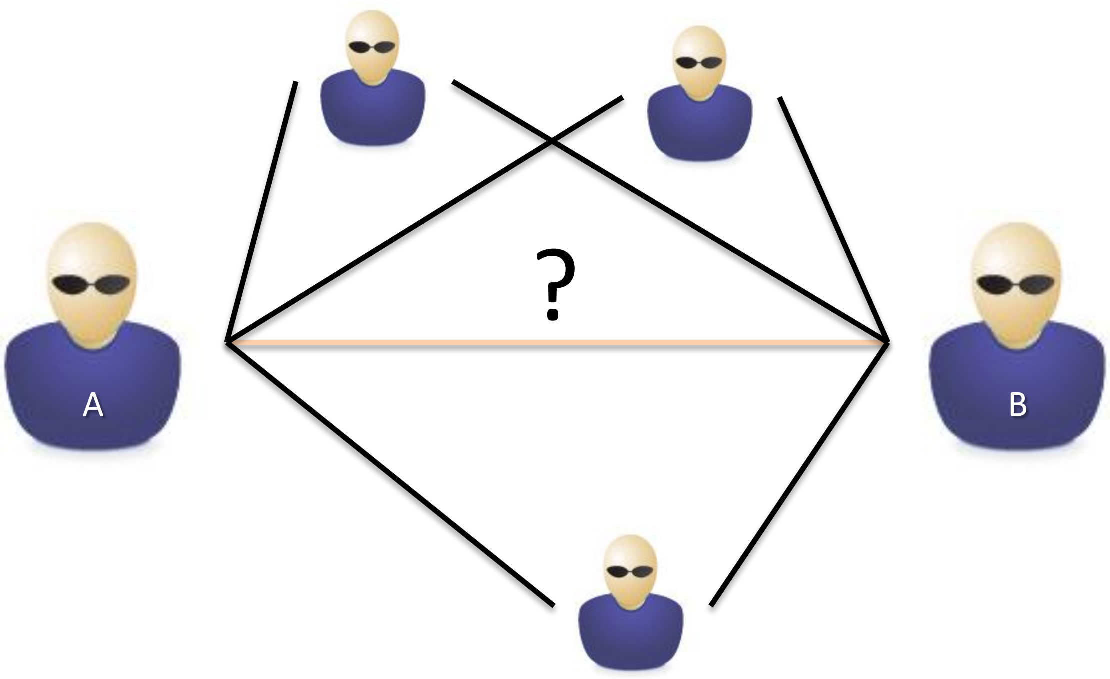
- Link prediction algorithms can estimate whether two users in a social network are connected.
- Users with many connections that cannot be supported by link predication algorithms may deemed to be faked.



Michael Fire, Lena Tenenboim, Ofrit Lesser, Rami Puzis, Lior Rokach, Yuval Elovici, "Computationally Efficient Link Prediction in Variety of Social Networks", ACM Transactions on Intelligent Systems and Technology, Volume 5 Issue 1, December 2013:1-25,

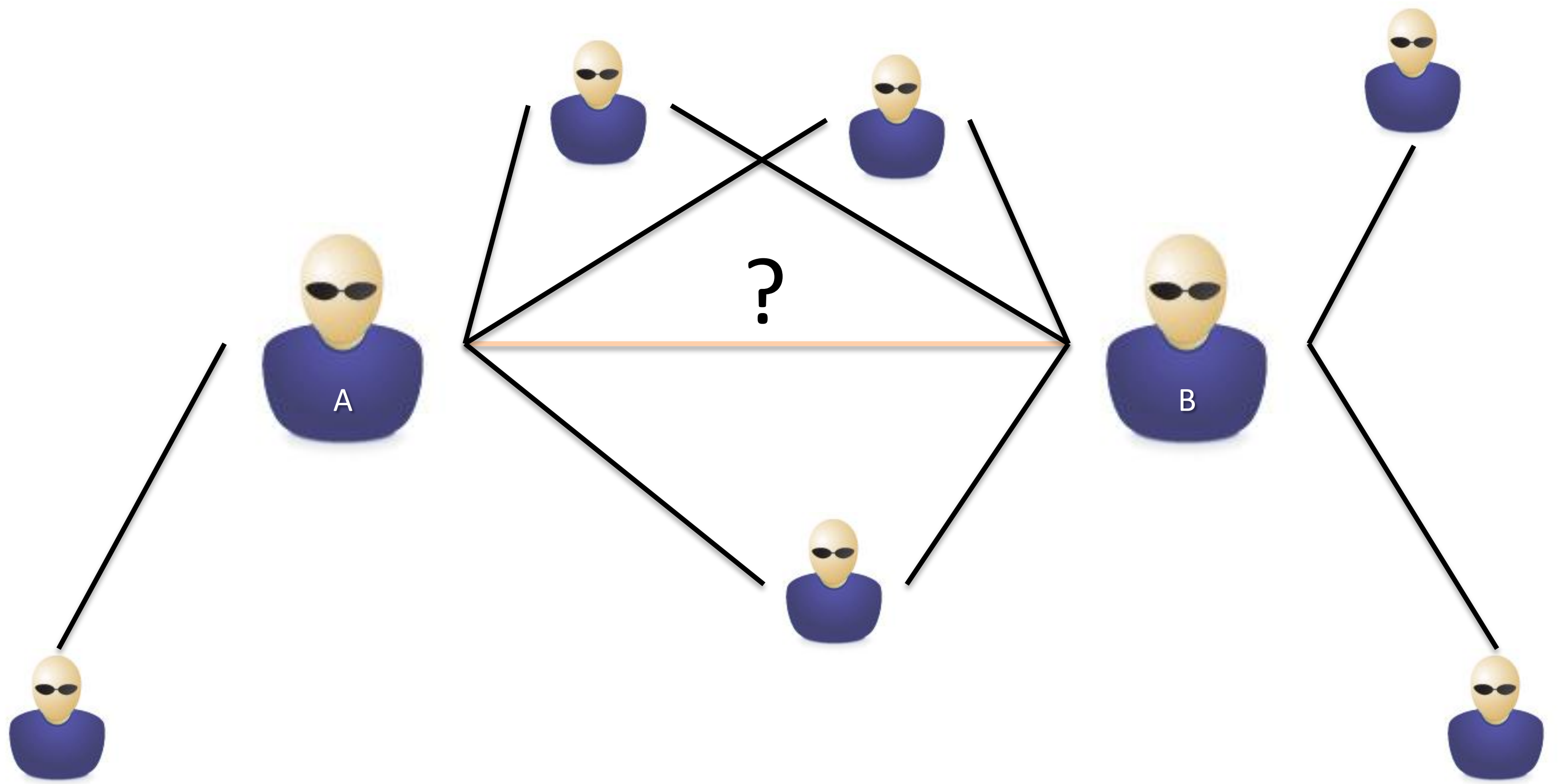
Link Prediction

- Number of common friends (3)



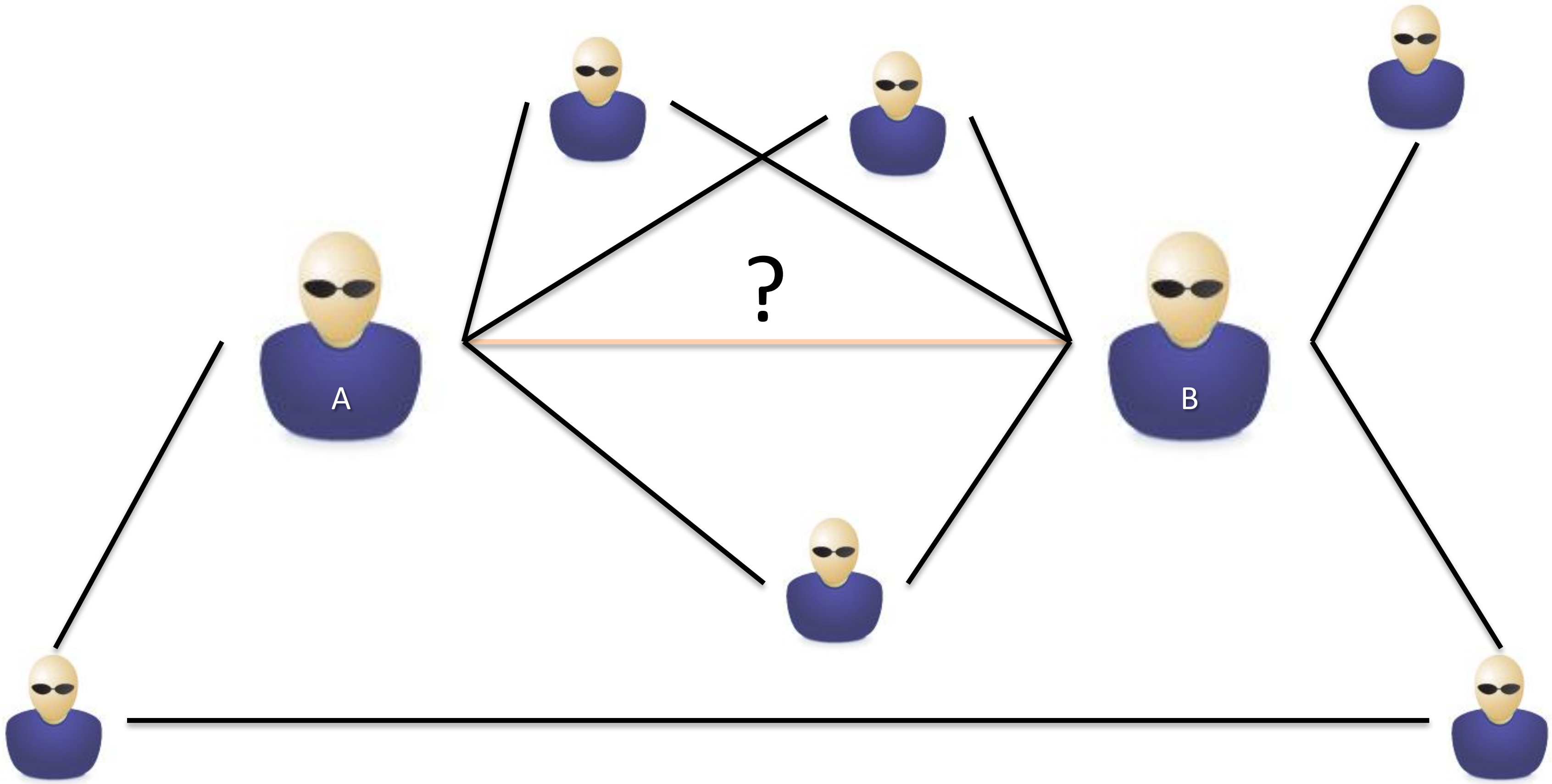
Link Prediction

- Jaccard coefficient (3 / 6)

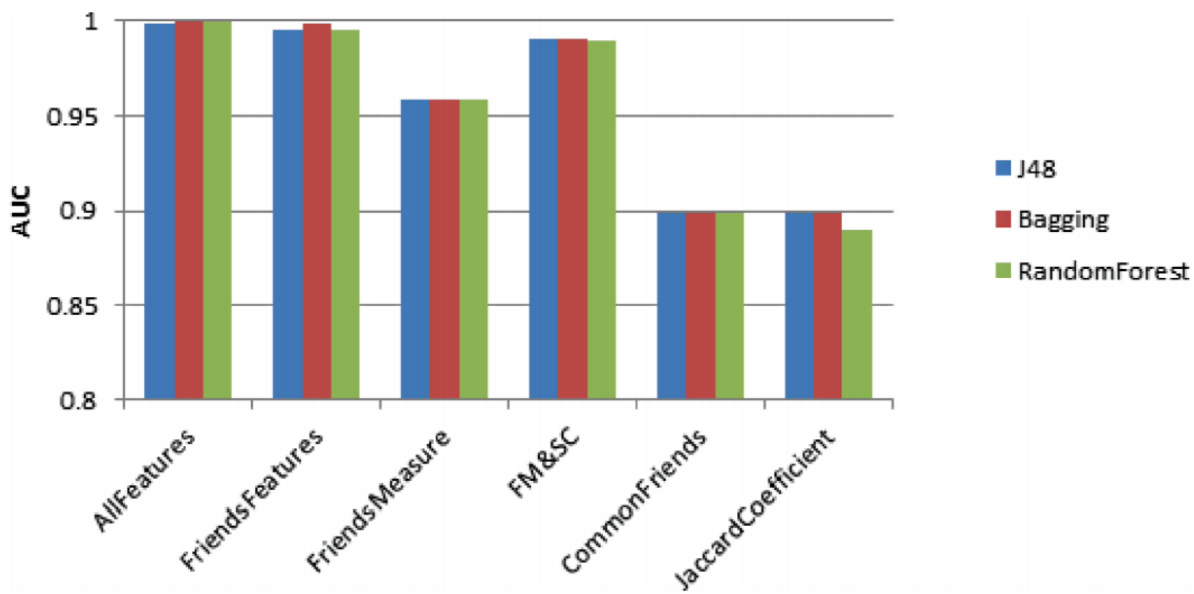


Link Prediction

- 2-3 path count ….



Link Prediction: Results



—*Friends-features subset* contains the following features: vertices *degree* features *Common friends*; *Total friends Preferential attachment score Same community*, and *Friends measure*. A total of 9 features for undirected networks and 16 features for directed networks were created.

—*Friends measure and Same community (FM & SM)* contains the *Friends-measure* and the *Same-community* features.

—*Common-friends subset* contains only the *Common-friends* feature.

—*Friends-measure subset* contains only the *Friends-measure* feature.

—*Jaccard's coefficient* contains only the *Jaccard's coefficient* feature.

—*Same community* contains only the *same-community* feature.

Existing Challenges

- Limited ground truth
- Class imbalance
- Adversarial Data Mining
- Feature engineering
- False positive
- Over-fitting to certain type of threat or environment configuration
- Big Data
- Concept Drift
- Limited explanation and attack attribution
- Curse of Dimensionality
- No free lunch
- Knowledge bottleneck

Addressing the Challenges

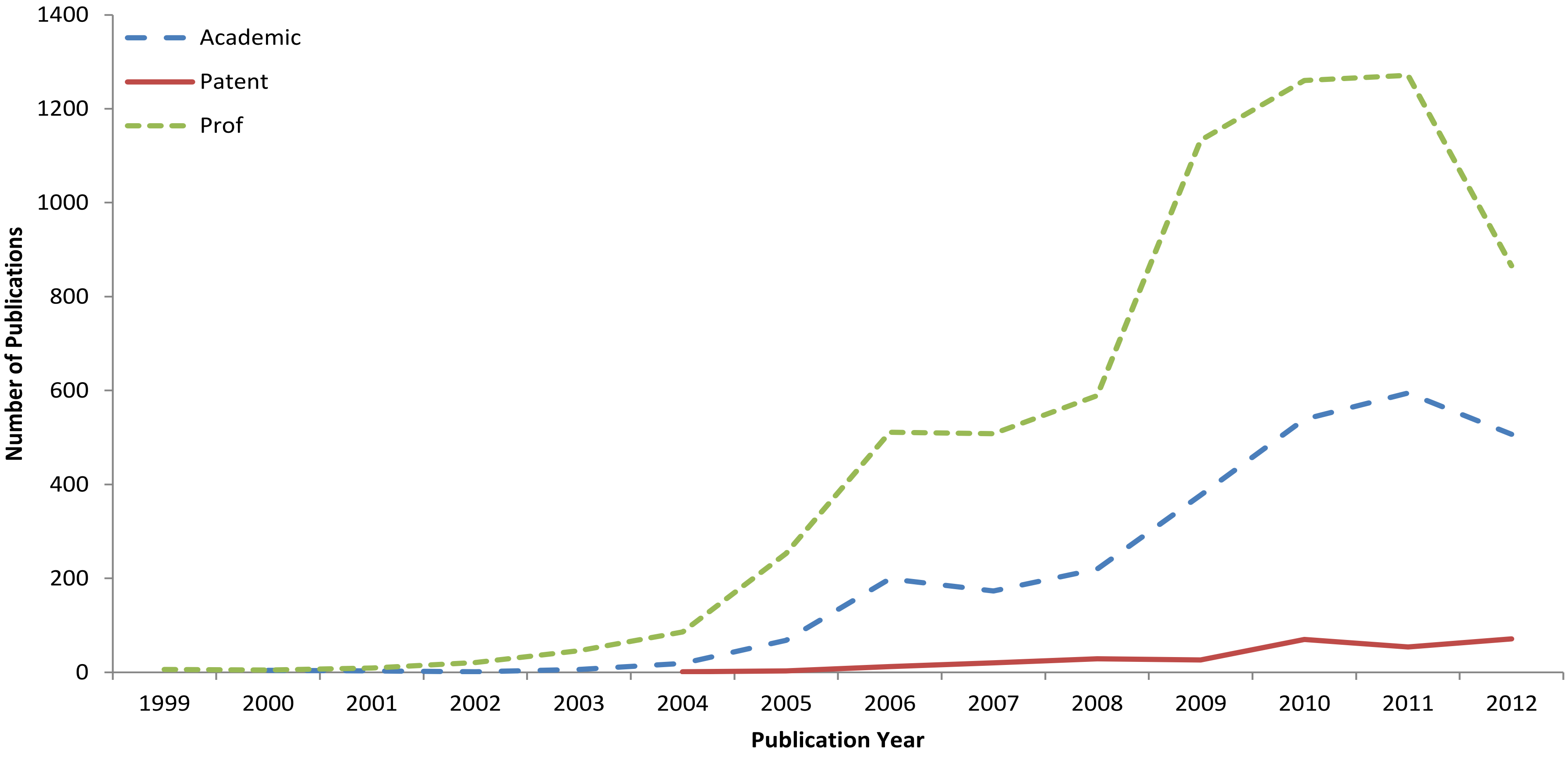
- Using Cutting Edge Big Data Technologies
- Using Modern Machine Learning Methods
  - Deep Learning
  - Active Learning
  - Transfer Learning
  - Ensemble Learning
- Incorporating ML Training in Cyber Security Curriculum
- Creating a common cyber security ontology
- Increasing collaboration and data sharing

Cyber Security Center Current Research Projects

1. MalSnap – Detection of Malware Presence in Private Clouds (VM) (including Ransomware Crypto-lockers.)
2. Sherlock – Closely track the mobile phones of dozens of users for 3 years to investigate the infection stage and out-of-context malicious usage.
3. Beehive – analysis the data of thousands of honeypots around the globe to study propagation patterns and who is next to be attacked.
4. Cyber-Med: Detection of Malware in Medical Devices.
5. Source Code Security Analysis – using RNN
6. USBWARE– Detection of USB based attacks.
7. Cyber Watson – Using IBM Watson for helping security analytics

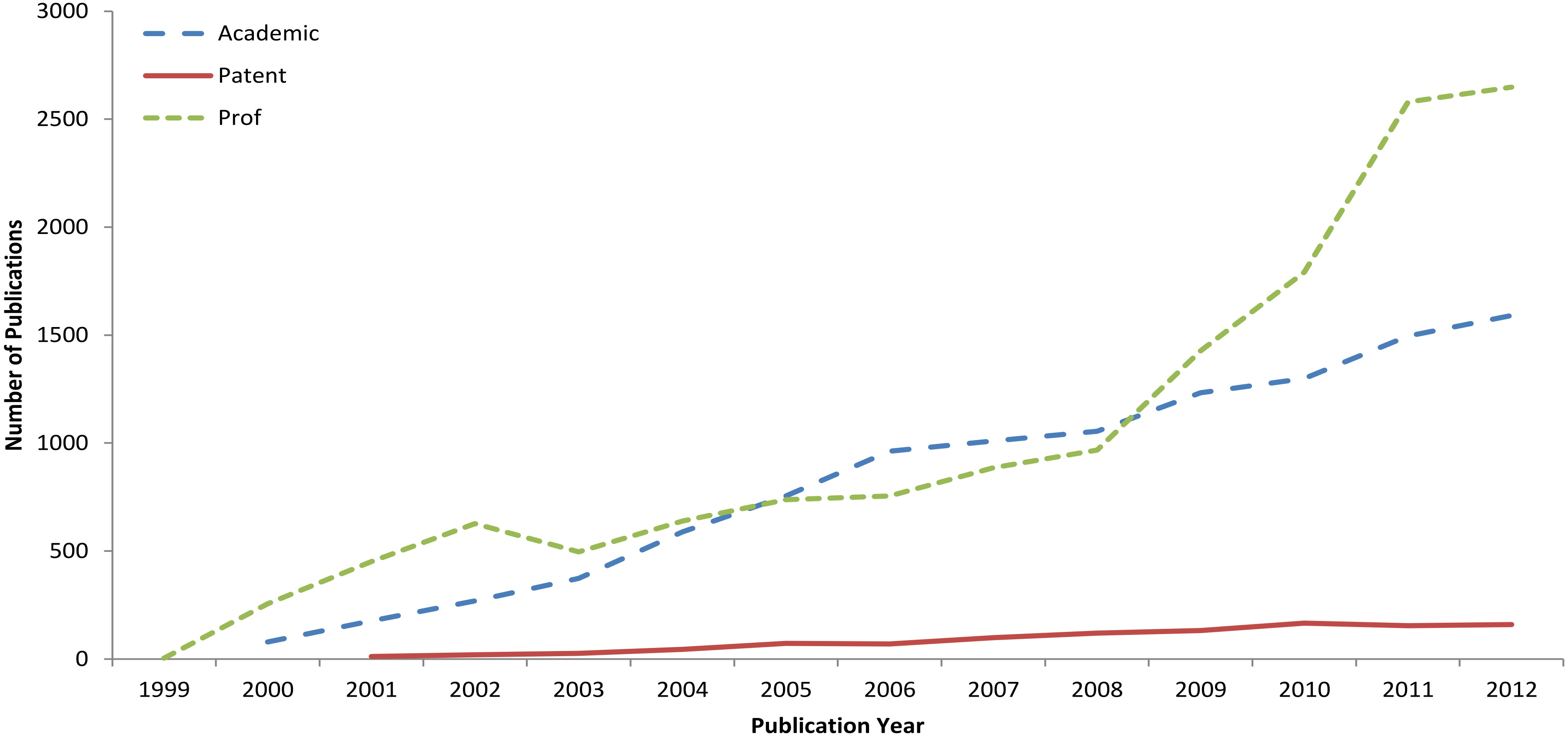
Academia as an Innovation Leader- BotNet Example

First reported (year)	First mentioned in a professional article	First scientific publication	First patent application
1999	1999	2000	2004



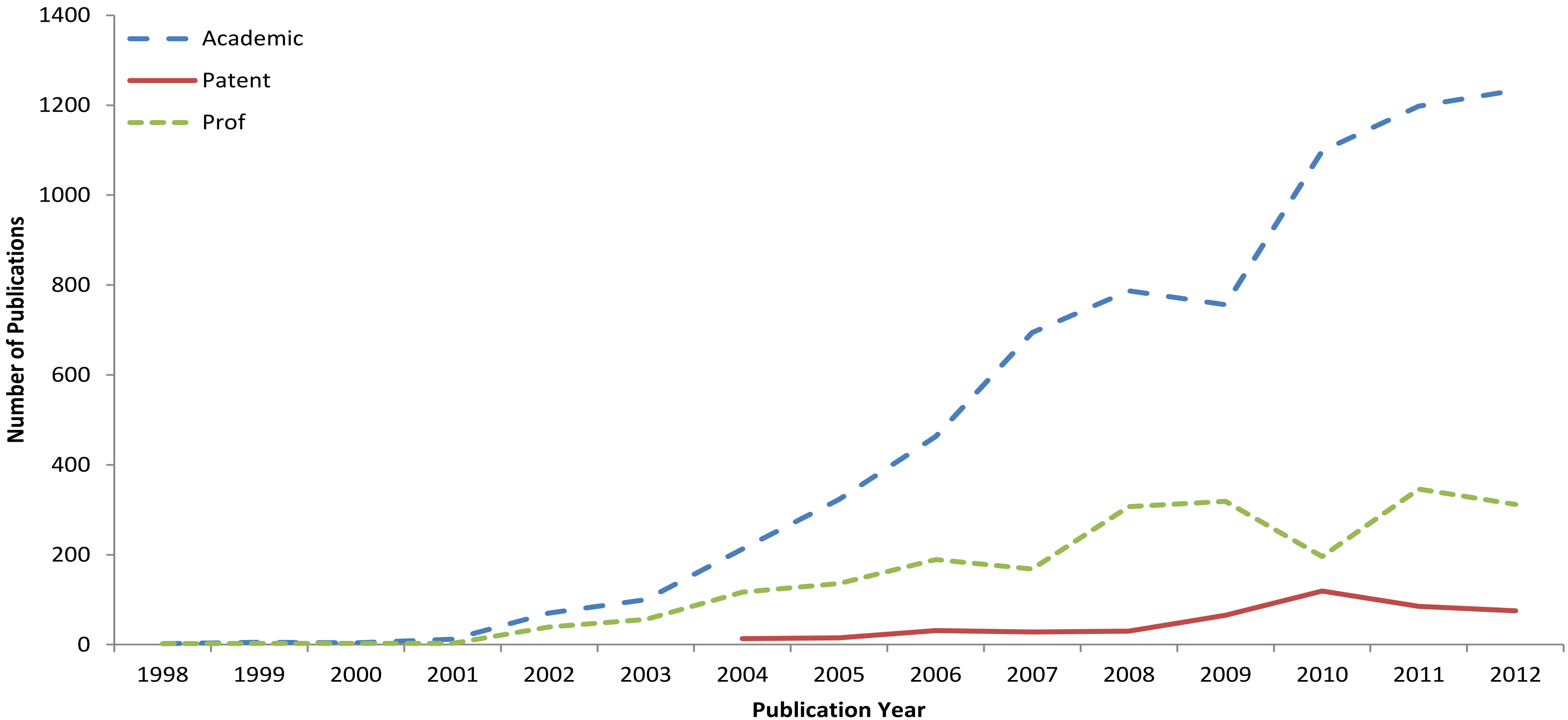
DDoS

First reported (year)	First mentioned in professional article	First scientific publication	First patent application
1999	1999	2000	2001



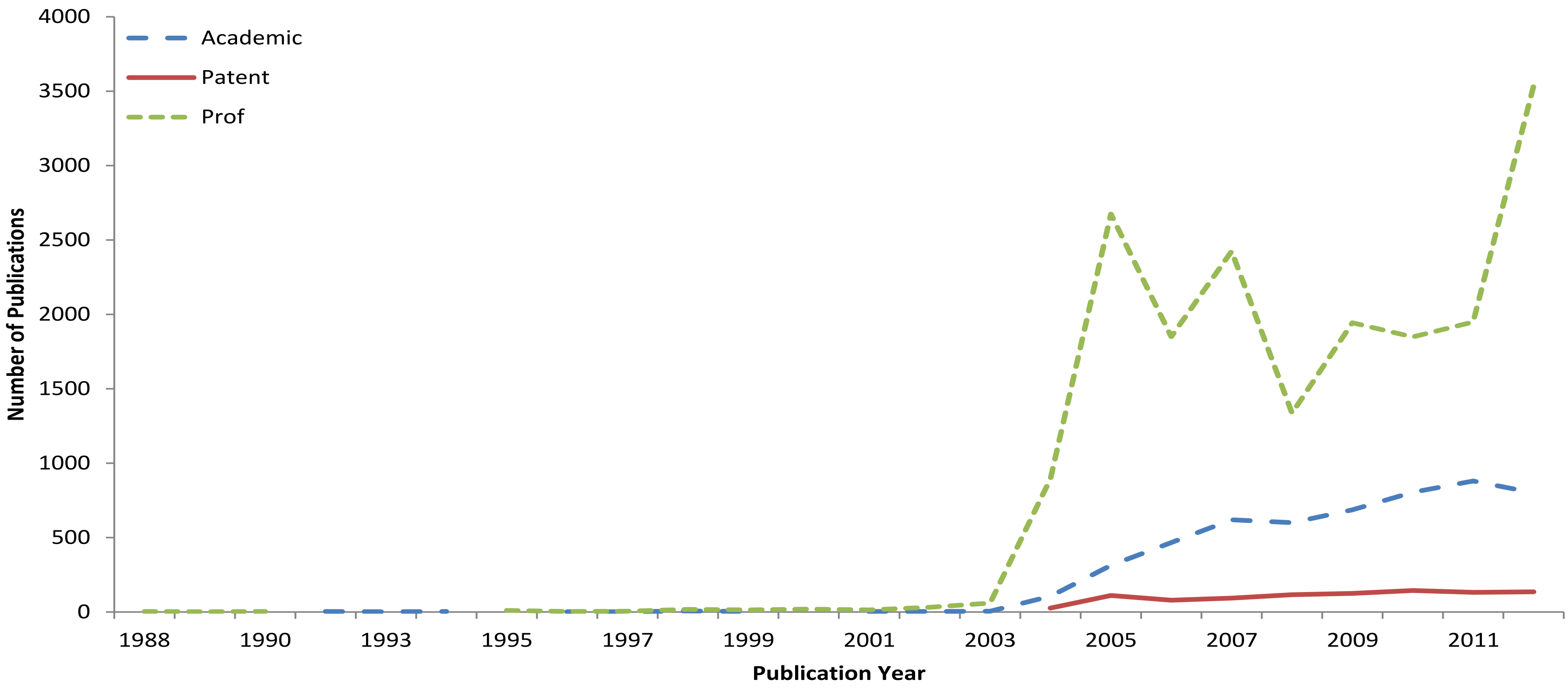
SQL Injection

First reported (year)	First mentioned in professional article	First scientific publication	First patent application
1998	1998	1998	2004



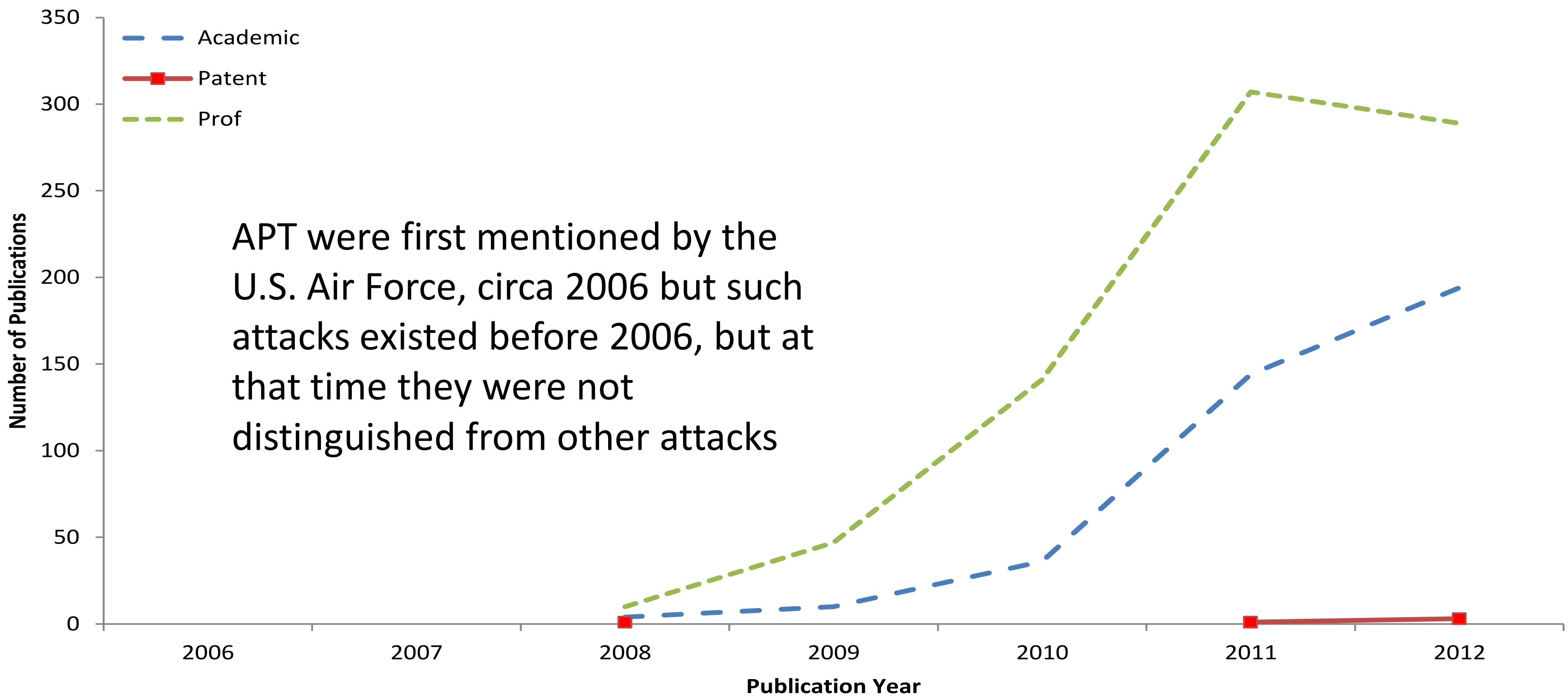
Phishing

First reported (year)	First mentioned in professional article	First scientific publication	First patent application
1987	1988	1988	2004



APT

Threat class	First reported (year)	First mentioned in professional article	First scientific publication	First patent application
APT	2006	2008	2008	2008



Summary

- Many current and emerging computer and network security challenges can be addressed by machine learning techniques.
- But, it is very important to employ machine learning techniques in the **right way**, in particular:
  – Carefully select the training corpora,
  – Feature engineering
  – Effective feature selection for reducing dimensionally reduction
  – Valid evaluations on a representative corpora.

ICSML

- International Summer School for Graduate Students in Beer-Sheva.
- Students from all over the world:
  - USA
  - Europe (Mainly Germany and Italy)
  - Asia (Mainly china and India)
- Rich curriculum which includes 180 hours.
- Practical and hand-on sessions using Machine-Learning methods for Cyber Security Applications.
- Mostly paid by the Israeli Ministry of Education.
- 30 out of 120 applicants are selected.