Politecnico di Milano
Dip. Elettronica e Informazione
Milano, Italy

Unsupervised Learning and Data Mining for Intrusion Detection

Stefano Zanero

Ph.D. Student, Politecnico di Milano

CTO & Founder, Secure Network S.r.l.

CanSecWest – Vancouver, 21-23/04/2004

Presentation Outline

- A case for Intrusion Detection Systems
- Intrusion Detection Systems, not Software !
- A brief taxonomy of intrusion detection systems
- State of the art in intrusion detection techniques
- Learning algorithms, patterns, outliers
- Conclusions

Parallel landscapes: physical vs. digital

- A discomforting parallel between physical and digital security
- Since 9/11/2001 we are building impressive defensive fortifications
  - Cost
  - Distraction
  - Annoyance
- Are we more secure today than we were three years ago ? Does not seem so
  - The defender needs to plan for everything… the attacker needs just to hit one weak point
  - King Darius vs. Alexander Magnus, at Gaugamela (331 b.C.)
- Why are we failing? Because in most cases we are not acting *sensibly*
- “Beyond fear”, by Bruce Schneier: a must read!

Information Security Engagement rules

- We cannot really defend against everything… but we can behave *sensibly*:
  - We can try to display *defenses* in the most vulnerable areas (deterrence)
  - We can try to protect the systems, designing them to be secure (prevention)
- At the end of the day, we must keep in mind that every defensive system will, at some time, fail, so we must plan for failure
  - We must design systems to *withstand* attacks, and fail gracefully (failure-tolerance)
  - **We must design systems to be *tamper evident* (detection)**
  - We must design systems to be capable of recovery (reaction)

Murphy’s law on systems

- The only difference between systems that can fail and systems that cannot possibly fail is that, when the latter actually fail, they fail in a totally devastating and unforeseen manner that is usually also impossible to repair
- The mantra is: **plan for the worst** (and pray it will not get even worse than that) and act accordingly

Tamper evidence and Intrusion Detection

- An information system must be designed for *tamper evidence* (because it *will* be broken into, sooner or later)
- An IDS is a *system* which is capable of detecting intrusion attempts on an *information system*
  - An IDS is a system, not a software!
  - An IDS works on an information system, not on a network!
- The so-called IDS software packages are a *component* of an intrusion detection system
- An IDS system usually closes its loop on a human being (who is an essential part of the system)

Breaking some hard-to-kill myths

- An IDS is a system, not a software
  - A skilled human looking at logs is an IDS
  - A skilled network admin looking at TCPdump is an IDS
  - A company maintaining and monitoring your firewall is an IDS
  - A box bought by a vendor and plugged into the network is **not** an IDS by itself
- An IDS is not a panacea, it's a component
  - Does not substitute a firewall, nor it was designed to (despite what Gartner thinks)
  - It's the last component to add to a security architecture, not the first
- Detection without reaction is a no-no
  - Like burglar alarms with no guards!
- Reaction without human supervision is a dream
  - "Network, defend thyself !"

Terminology and taxonomies

- Different types of software involved in IDS
  - Logging and auditing systems
  - Correlation systems
  - So-called “IDS” software
  - Honeypots / honeytokens
- The logic behind an IDS is always the same: those who access a system for illegal purposes act differently than normal users
- Two main detection methods:
  - Anomaly Detection: we try to describe what is normal, and flag as anomalous anything else
  - Misuse Detection: we try to describe the attacks, and flag them directly

Anomaly vs. misuse

Anomaly Detection Model

- Describes normal behaviour, and flags deviations
- Uses statistical or machine learning models of behaviour
- Theoretically able to recognize any attack, also 0-days
- Strongly dependent on the model, the metrics and the thresholds
- Generates statistical alerts: “Something’s wrong”

Misuse Detection Model

- Uses a knowledge base to recognize the attacks
- Can recognize only attacks for which a “signature” exists in the KB
- When new types of attacks are created, the language used to express the rules may not be expressive enough
- Problems for polymorphism
- The alerts are precise: they recognize a specific attack, giving out many useful informations

Misuse detection alone is an awful idea

- Misuse detection systems rely on a knowledge base (think of the anti-virus example, if it's easier to grasp)
- Updates continuously needed, and not all the attacks become known (as opposed to viruses)
- Signature engineering problems (an anti-virus update, a couple of years ago, recognized *itself* as a virus...)
- Attacks are polymorphs, more than computer viruses: ADMutate, UTF encoding…
- Attacks are not atomic, and interleaving helps in avoiding detection!

Anomaly Detection, perhaps not better

- We must describe the behaviour of a system
  - Which features/variables/metrics do we use?
  - Which model do we choose to fit them?
- Thresholds must be chosen to minimize false positive vs. detection rate: a difficult process
- The base model is fundamental: if the attack shows up only in variables we discarded, the system is blind on it!
- Any type of alert is more or less equivalent to “hey, something’s wrong here”. What? Your guess!

Learning Algorithms for an IDS

- What is a learning algorithm ?
  - It is an algorithm whose performances grow over time
  - It can extract information from training data
- Supervised algorithms learn on labeled training data
  - “This is a good packet, this is not good”
  - Think of your favorite bayesian anti-spam filter
  - It is a form of generalized misuse detection, more flexible than signatures
- Unsupervised algorithms learn on unlabeled data
  - They can “learn” the normal behavior of a system and detect variations
  - How can they be employed on networks ?
- We are developing a network-based, anomaly-based intrusion detection system capable of *unsupervised learning*

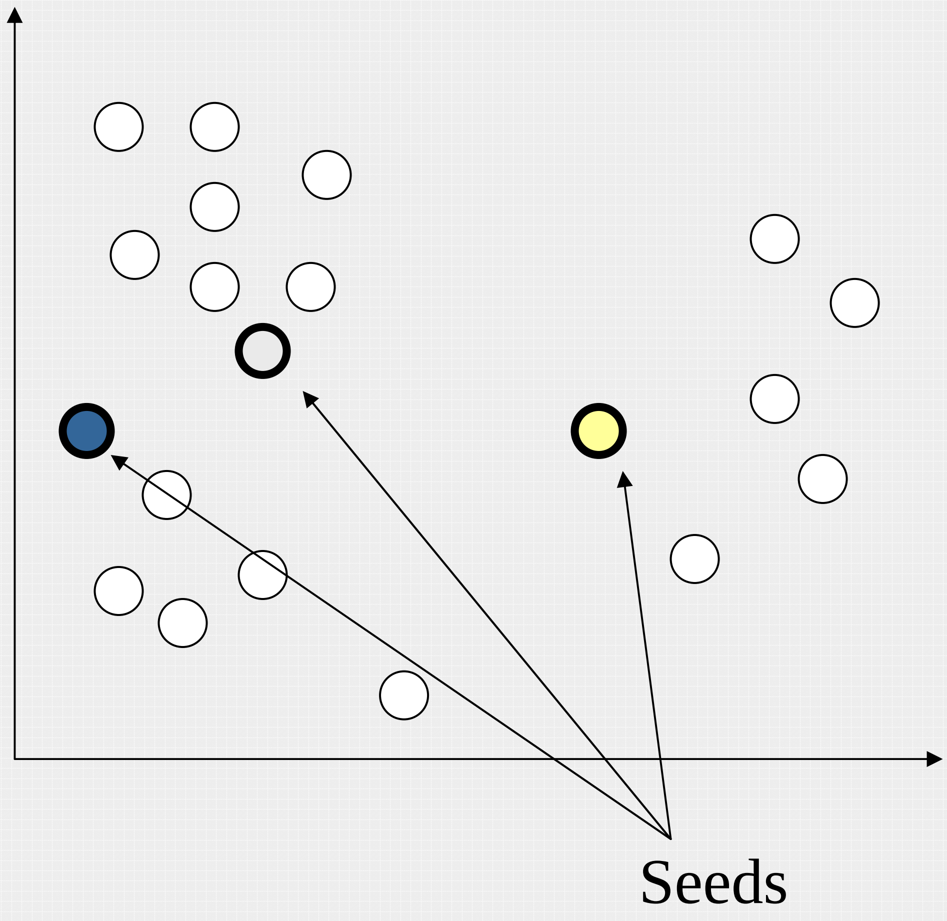
Unsupervised Learning Algorithms

- What are they used for:
  - Find natural groupings of X (X = human languages, stocks, gene sequences, animal species,…) in order to discovery hidden underlying properties
  - Summarize <data> for the past <time> in a visually helpful manner
  - Sequence extrapolation: predict cancer incidence in next decade; predict rise in antibiotic-resistant bacteria
- A general overview of methods:
  - Clustering (“grouping” of data)
  - Novelty detection (“meaningful” outliers)
  - Trend detection (extrapolation from multivariate partial derivatives)
  - Time series learning
  - Association rule discovery

What is clustering ?

- *Clustering is the grouping of pattern vectors into sets that maximize the intra-cluster similarity, while minimizing the inter-cluster similarity*
- What is a pattern vector (tuple)?
  - A set of measurements or attributes related to an event or object of interest:
  - E.g. a persons credit parameters, a pixel in a multi-spectral image, or a TCP/IP packet header fields
- What is similarity?
  - Two points are similar if they are "close"
- How is "distance" measured?
  - Euclidean
  - Manhattan
  - Matching Percentage

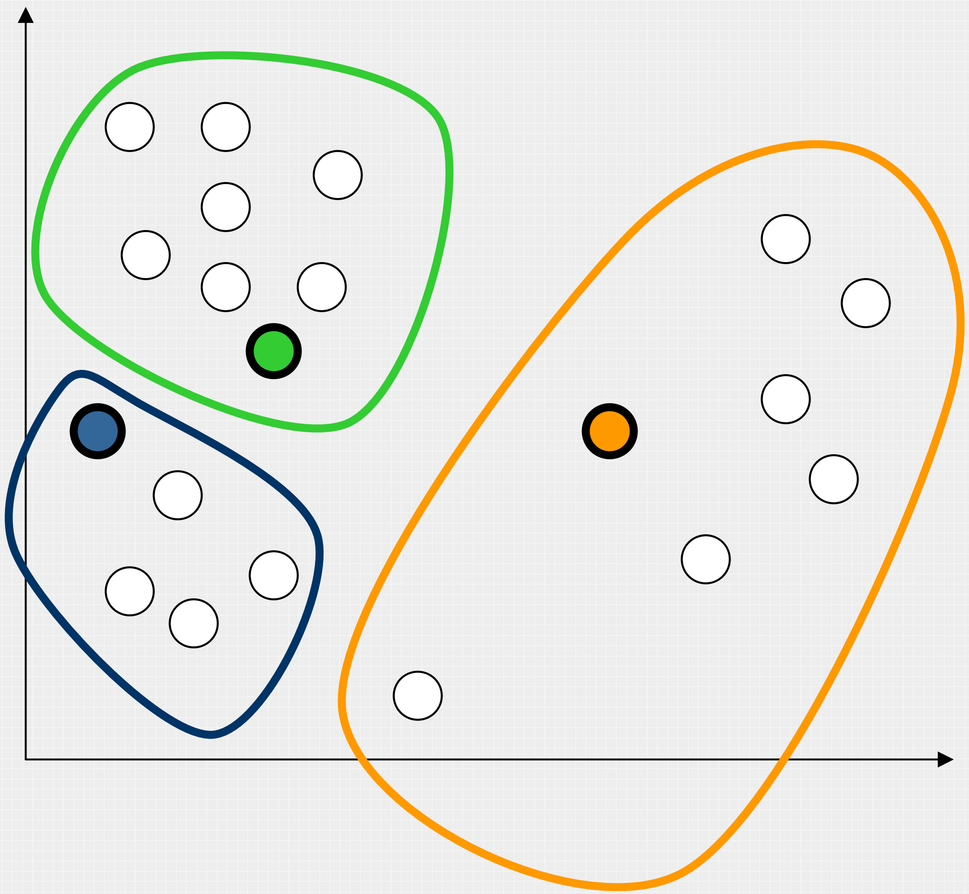
An example: K-Means clustering



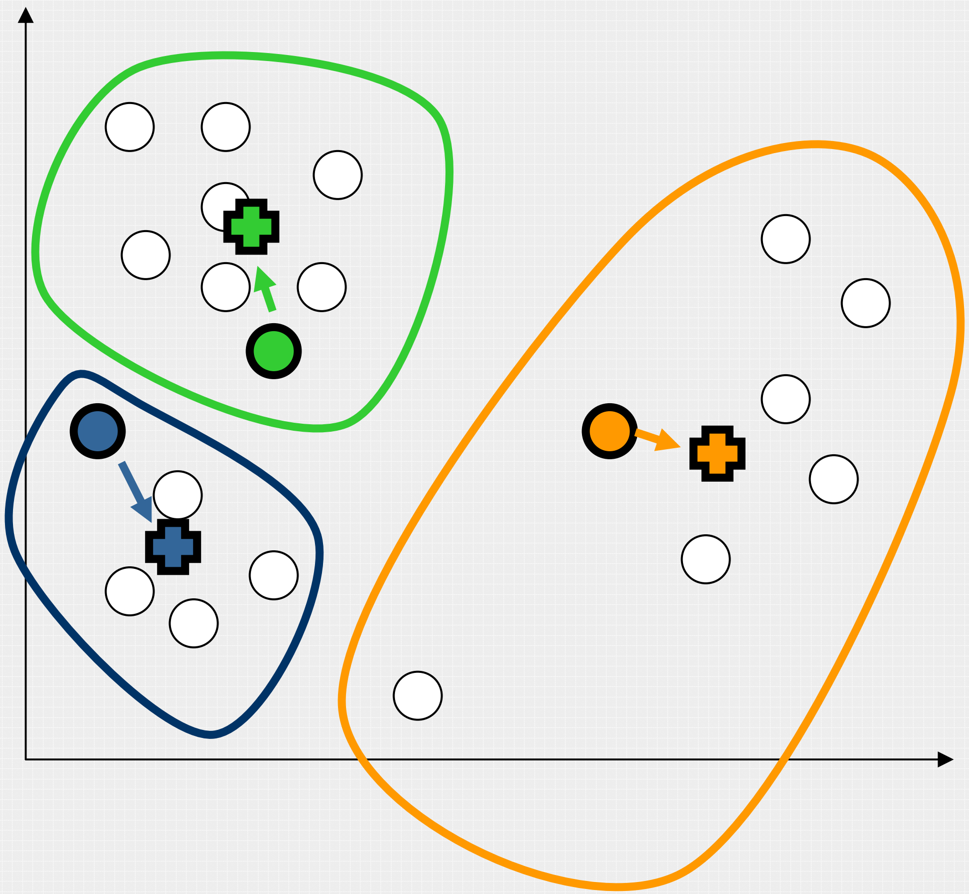
Predetermined number of clusters

Start with seed clusters of one element

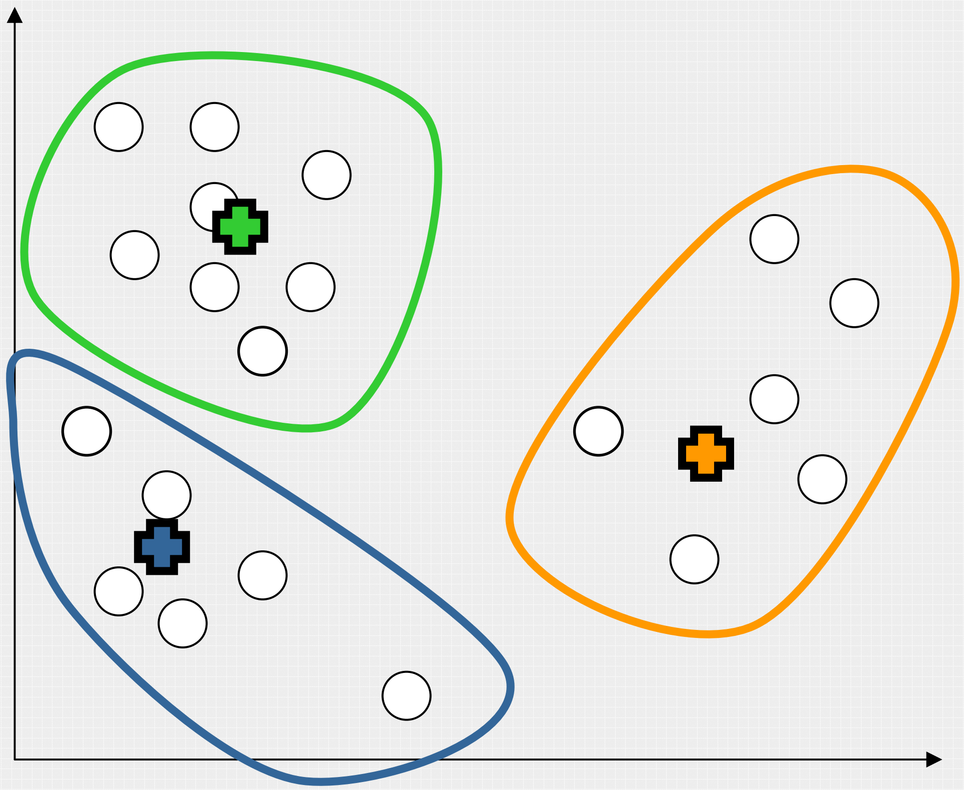
Assign Instances to Clusters



Find the new centroids



Recalculate clusters on new centroids



Which Clustering Method to Use?

- There are a number of clustering algorithms, K-means is just one of the easiest to grasp
- How do we choose the proper clustering algorithm for a task ?
  - Do we have a preconceived notion of how many clusters there should be?
  - How strict do we want to be?
    - Can a sample be in multiple clusters ?
    - Hard or soft boundaries between clusters
  - How well does the algorithm perform and scale up to a number of dimensions ?
- The last question is important, because data miners work in an offline environment, but we need speed!
  - Actually, we need speed in classification, but we can afford a rather long training

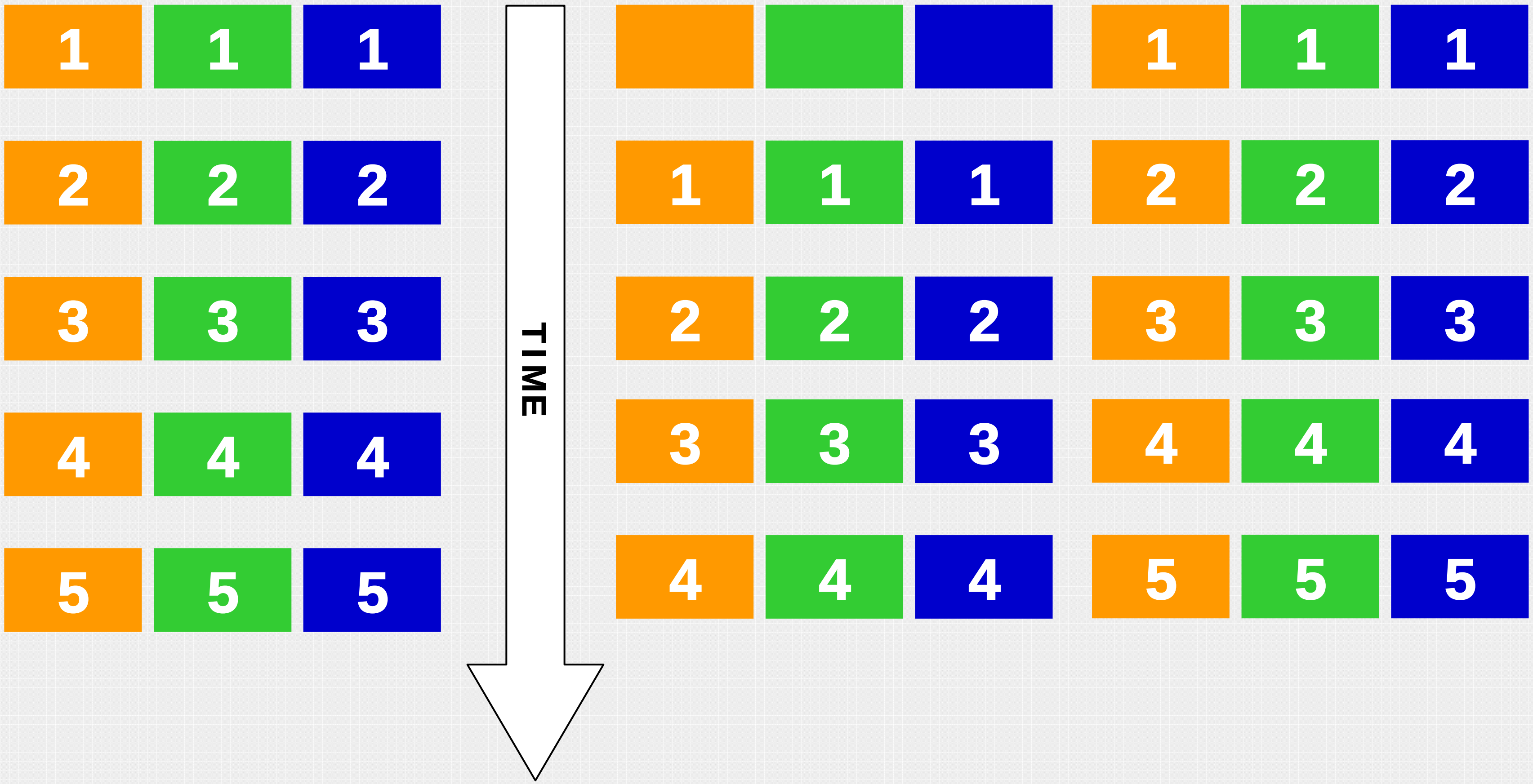
Outlier detection

- What is an outlier ?
  - It's an observation that deviates so much from other observations as to arouse suspicions that it was generated from a different mechanism
- If our observations are packets... attacks probably are outliers
  - If they are not, it's the end of the game for unsupervised learning in intrusion detection
- There is a number of algorithms for outlier detection
- We will see that, indeed, many attacks are outliers

Multivariate time series learning

- A time series is a sequence of observations on a variable made over some time
- A multivariate time series is a sequence of vectors of observations on multiple variables
- If a packet is a vector, then a packet flow is a multivariate time series
- What is an outlier in a time series ?
  - Traditional definitions are based on wavelet transforms but are often not adequate
- Clustering time series might also be an approach
  - We can transform time series into a sequence of vectors by mapping them on a rolling window

Mapping time series onto vectors



Association Rule Discovery

- The objective is to find rules that associate sets of events. E.g. X & Y=> Z
- We use 2 evaluation criteria:
  - Support (frequency): probability that an observation contains {X & Y & Z}
  - Confidence (accuracy): the conditional probability that an observation having {X & Y} also contains Z
- Used both in supervised and unsupervised manners
- Example: ADAM, Audit Data Analysis and Mining (supervised)

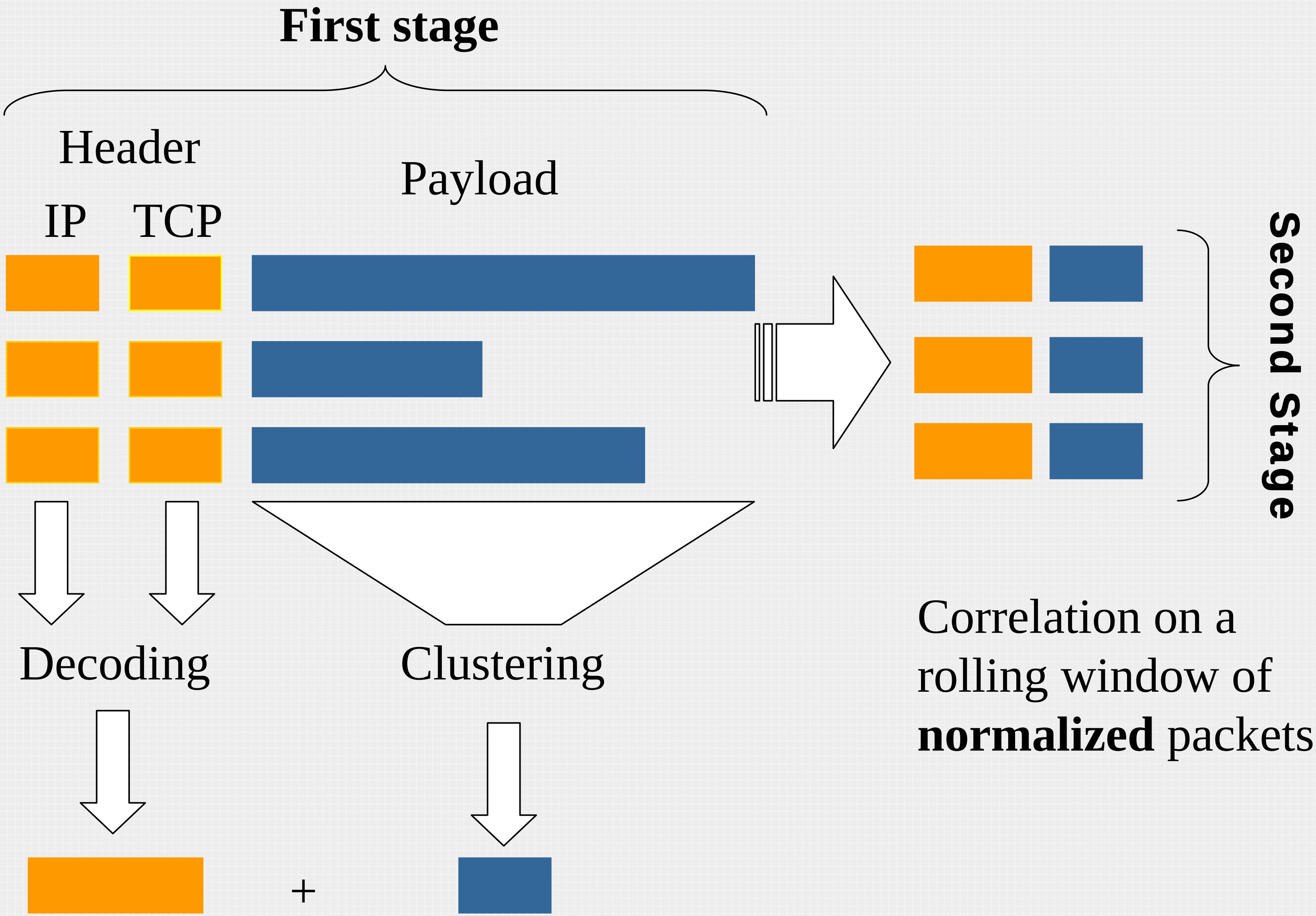
Selecting features

- Most learning algorithms do not scale well with the growth of irrelevant features
  - Training time to convergence may grow exponentially
  - Detection rate falls dramatically, from our experiments
- Computational efficiency gets lower when coordinates are higher
  - Some algorithms simply couldn’t handle too many dimensions in our tests
- Structure of data gets obscured with large amounts of irrelevant coordinates
- Run-time of the (already trained) inference engine on new test examples also grows

A hard problem, then…

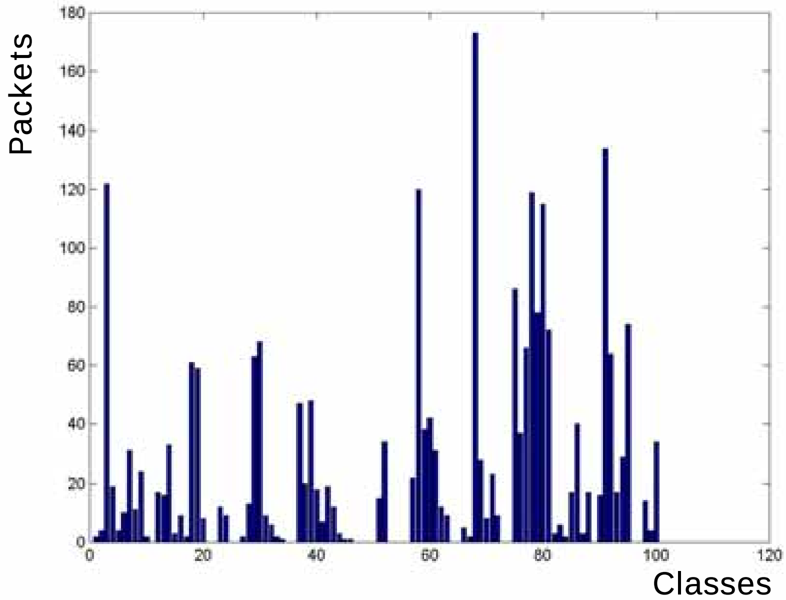
- A network packet carries an unstructured payload of data of varying dimension
- Learning algorithms like structured data of fixed dimension since they are vectorized
- A common solution approach was to *discard the packet contents.* Unsatisfying because many attacks are right there
- We used **two** layers of algorithms, prepending a clustering algorithm to another learning algorithm
- Published in S. Zanero, S. M. Savaresi, “Unsupervised Learning Techniques for an Intrusion Detection System”, Proc. of the 2004 ACM symposium on Applied computing, Nicosia, Cyprus

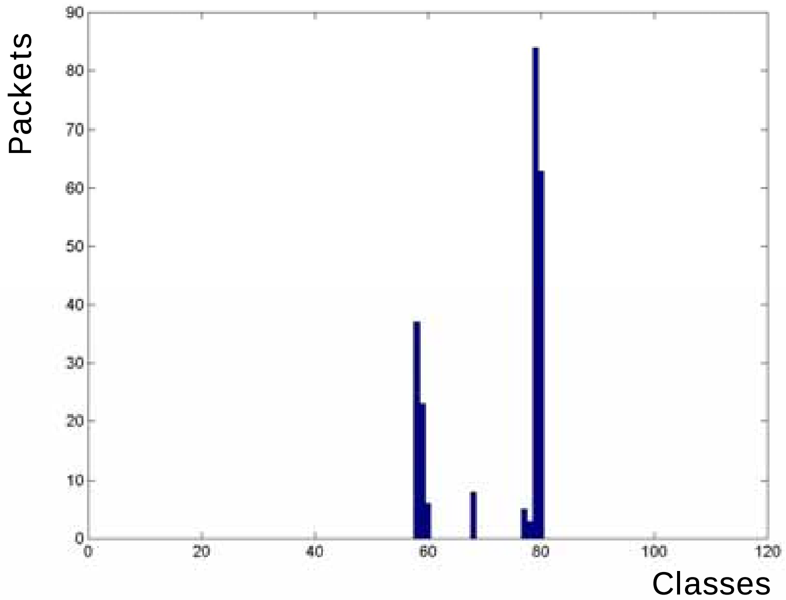
The overall architecture of the IDS



Correlation on a rolling window of **normalized** packets

An example of clustering results





- We clustered in 100 classes the packets of a test network on the left with a Self Organizing Map
- On the right, the classification of a window of packets towards TCP port 21
- As you can see, they are very well characterized!
- We experimented various attacks, and they fall outside this characterization: thus, they can be detected automatically

Attack detection, polymorphism resistance

- We have seen the classification of packets towards TCP port 21
- We experimented the "format string" vulnerability against wu-ftpd FTP server (CVE CAN-2000-0573)
  - We did NOT give to the system a sample of this attack forehand
  - The payload was classified in class 69, which is not commonly associated with FTP packets
  - Port 21, class 69 is an outlier, and is detected
- We also analyzed the globbing DoS attack,
  - It is inherently polymorph
  - The SOM classified a number of variants of the attack in the same class (97), which is also not commonly associated with FTP packets

Unsupervised learning at the second tier

- We are still experimenting with candidate algorithms for second tier learning
- Basically, any of the proposed algorithms found in the literature can be complemented by our clustering tier
- Our first results show that applying the additional stage can extend the range of detected attacks, improving average detection rate by as much as 75%
- False positive rate is also affected, but we are working to lower it

Conclusions & Future Work

- Conclusions:
  - We have introduced a two-tier architecture for applying unsupervised learning algorithms to network data for intrusion detection purposes
  - We have shown the feasibility of clustering TCP payloads to obtain meaningful results
  - We have shown that implementing the two-tier architecture improves the performance of existing systems
- Future developments:
  - We will study the best algorithm for second stage
  - We will carefully explore signal-to-noise ratio and false positive reduction techniques
  - We will study integration of our system in the architecture of Snort as a plugin

Thank you!

Any question?

Stefano Zanero
zanero@elet.polimi.it
www.elet.polimi.it/upload/zanero