Lecture #8: The Base-Rate Fallacy

UCalgary ENSF619

Elements of Software Security

Instructor: Lorenzo De Carli (lorenzo.decarli@ucalgary.ca)

What is software security about?

- Try to think of some typical activities and operations…

What is software security? - II

- Many things, but one of the most important is **detecting bad stuff** :-)
    - Or more precisely, **distinguishing** bad stuff from good stuff
    - "Stuff": users, processes, files, network flows, network packets, DNS domains, URLs, machines, etc.

Classification in software security

- Much work in software security literature is really work on **classification**
- **Typical problem:** you are given a set O of objects, s.t. $O = O_M \cup O_B$, where O_M represents the subset of **malicious objects** and O_B that of **benign objects**
- You must define a **classification algorithm (aka detection algorithm, or detector):**
  a : $O \rightarrow \{B, M\}$ that maps each element in O to one of two classes: malicious (M) and benign (B)

Classification in software security - II

- To be precise, oftentimes we need to distinguish between benign and multiple malicious classes:
  $O \rightarrow \{B, M_1, M_2, \ldots, M_n\}$
  - E.g. multiple network attacks, different malware families
- However, from the point of view of detection, **the high-order goal is to be able to identify malicious objects**, whatever their type is
- For the purpose of keeping the discussion simple, for the rest of this lecture we are going to assume only **one malicious object class**

”Benign” vs “Malicious”

- There are many ways to provide a **definition** of what constitutes a malicious object (more on this later in the lecture)
- Typically, a model is constructed representing either the **characteristics of benign or malicious objects**
  - In the first case, everything which is not benign is implicitly malicious; in the second case, everything which is not malicious is implicitly benign

“Benign” vs “Malicious” - II

- Can you think of some examples of specific techniques used to define the two object classes?

“Benign” vs “Malicious” - III

- Generally speaking, the nature of an object can be defined by specifying:
  - Some **specific intrinsic properties** of the objects (e.g. hash of a file)
  - Some **statistical properties** of the object (e.g., average flow duration when the network is not under attack)

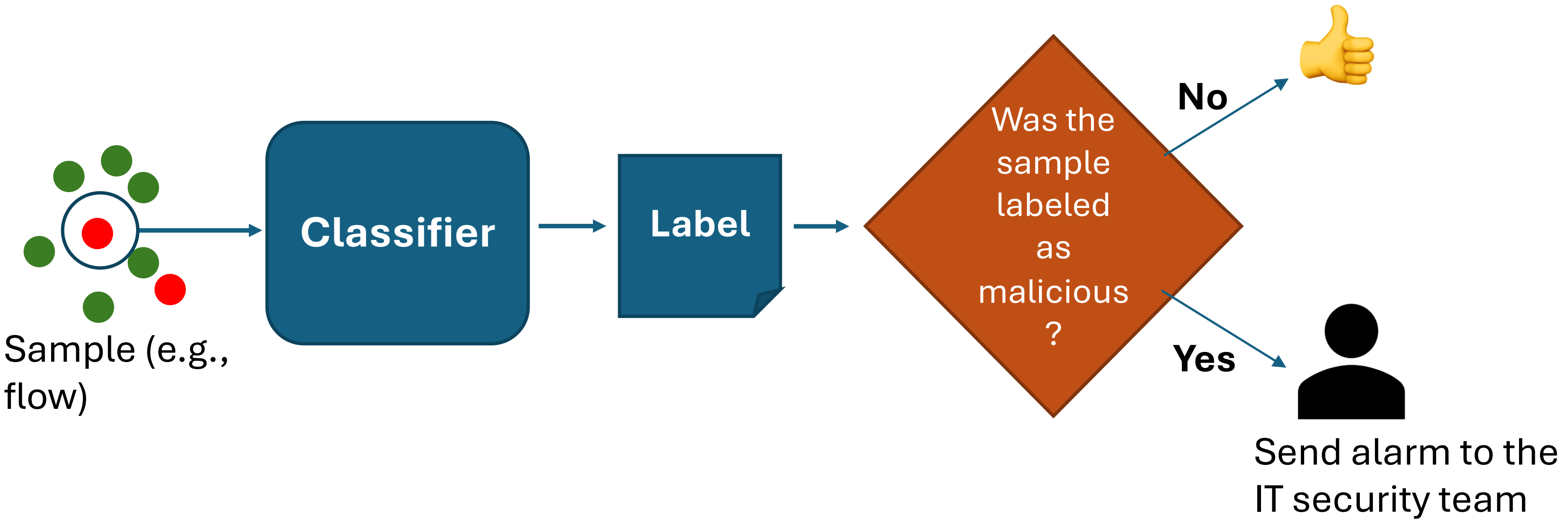
Why do we care about this?

- Oftentimes, detection problems in security are treated as standard classification problems **without much consideration or awareness of the underlying assumptions**
- This creates various problems, because security is a **fairly peculiar domain**:
  - Attacker may be able to control the sample “appearance”
  - Classes are not balanced
  - …

Focus & structure of today’s discussion

- We are going to review one of the issues arising when using classification algorithms without understanding the application domain: the **base-rate fallacy**
- Coming up next:
  - How is classifier effectiveness measured?
  - A brief review of Bayesian probability
  - “The Base-rate Fallacy and its Implications for the Difficulty of Intrusion Detection”

Workflow for detection


Sample (e.g., flow)
Classifier
Label
Was the sample labeled as malicious ?
No
Yes
Send alarm to the IT security team

Workflow for detection - II

- Typically, a detector only generates output **when malicious objects are identified**
  - Benign objects are silently ignored and allowed
- Throughout this lecture, I'll refer to the act of labeling an object as malicious as **generating an alarm**
- In many cases, **alarms are post-processed** (e.g. to merge together similar messages)
  - For simplicity, in this lecture we are going to assume **1 intrusion == 1 alarm**

Measures of classifier effectiveness

- A good classifier should:
  - Rarely fail to generate an alert in the presence of a malicious object (**high detection rate**)
  - Rarely generate an alert in the presence of a benign object (**low false alarm rate**)
- In many cases, the two properties are correlated: **a classifier which is sensitive enough to produce a high detection rate also tends to produce a high false alarm rate**
- However, our discussion today is valid even if we ignore this observation

Measures of classifier effectiveness

- Consider a **set of samples** $O = O_M \cup O_B$ and a **classifier** $O \rightarrow \{B, M\}$
- Given a sample o, there are **two possible errors:**
  - $o \in O_M$ and $o \rightarrow B$ (false negative)
  - $o \in O_B$ and $o \rightarrow M$ (false positive)
- Given a labeled set of samples O, we define:
  - TP = #true positives
  - FP = #false positives
  - TN = #true negatives
  - FN = #false negatives

What are those?

What do I get if I sum all of them?

Basic measures

- With $P = \#positives\ in\ dataset$ and $N = \#negatives\ in\ dataset$:
- **True positive rate:** $TPR = TP/P$ (#correctly identified positives over overall #positives)
- **True negative rate:** $TNR = TN/N$ (#correctly identified negatives over overall #negatives)
- **False positive rate:** $FPR = FP/N$
- **False negative rate:** $FNR = FN/P$
- Basic properties: $TPR + FNR = TNR + FPR = 1$

Accuracy

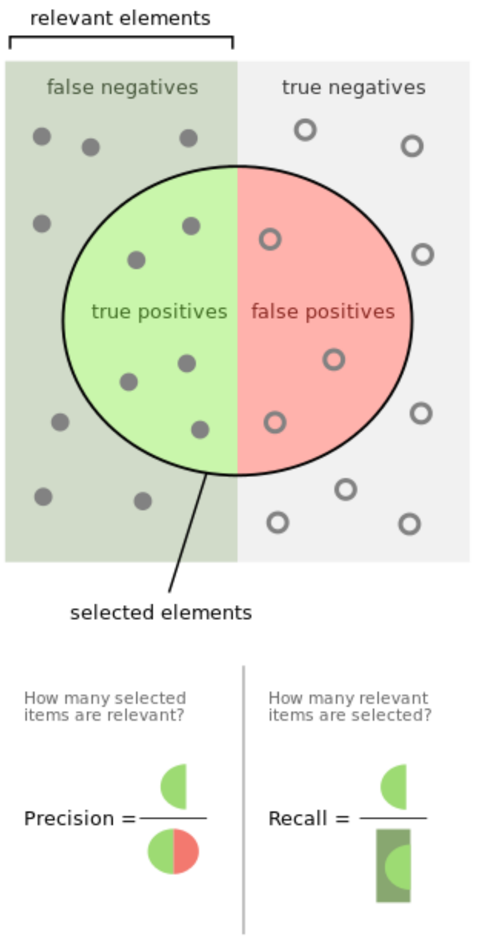
- $A = (TP + TN)/(TP+TN+FP+FN)$
- Aka “fraction correct”: **fraction of the population which is classified correctly**
- One of the most common measures of the ”quality” of a classifier

A reflection on accuracy

- The definition of accuracy **hides information**
  - Aggregates results concerning true negatives and true positives, which makes it impossible to tell them apart.
- E.g., if $|O_B| >> |O_M|$, a classifier that marks everything as "benign" has high accuracy
- Typically, when reporting performance results for a classifier it is best to accompany accuracy with other metrics (such as F1 score)

Precision & Recall

- **Precision:** $P = TP / (TP + FP)$
  - Which fraction of the elements marked as “malicious” is actually malicious?
- **Recall:** $R = TP / (TP + FN)$
  - Which fraction of the elements that are actually malicious is marked as “malicious”?



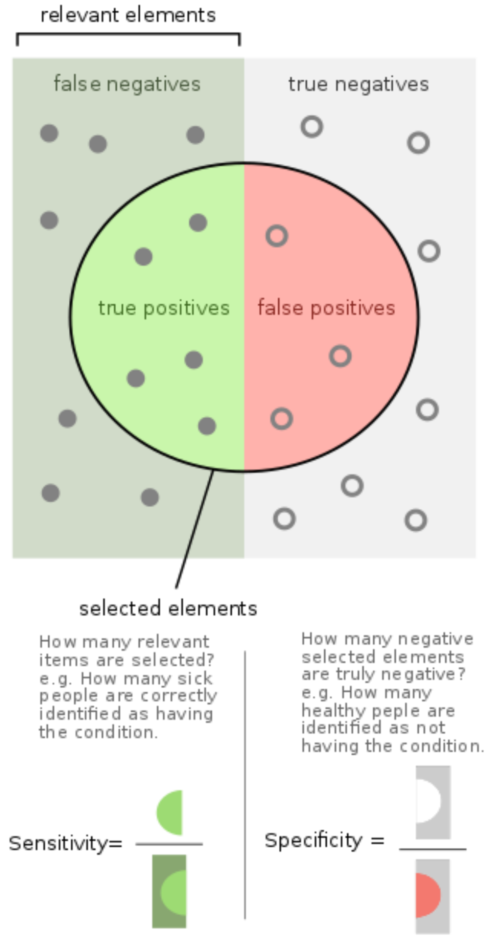
https://en.wikipedia.org/wiki/Precision_and_recall

F1 score

- $F_1 = 2(P \cdot R/P+R)$
- **Harmonic mean of precision and recall**
- Attempt to give a single scalar measure of classifier quality which is more informative than accuracy
- Example:
  - 1000 objects, 900 benign & 100 malicious
  - Classifier has 900 TN, 0 FP, 99 FN and 1 TP
  - **Classifier accuracy: 900 + 1 / 1000 = ~90%**
  - Precision = TP / (TP+FP) = 1 / 1 = 1
  - Recall = TP /(TP+FN) = 1/100 = 0.01
  - **Classifier F_1 score = 2((1 · 0.01)/(1+0.01)) ~= 0.02**

Sensitivity & Specificity

- **Sensitivity:** $Se = TP / (TP + FN)$
  - **Same as recall:** fraction of malicious objects that is detected
- **Specificity:** $Sp = TN / (TN + FP)$
  - Fraction of benign objects that are identified as benign
- Sensitivity & Specificity are common in the medical field, but rarely used in cybersecurity


relevant elements
false negatives
true negatives
true positives
false positives
selected elements
How many relevant items are selected? e.g. How many sick people are correctly identified as having the condition.
How many negative selected elements are truly negative? e.g. How many healthy peple are identified as not having the condition.
Sensitivity=
Specificity =

https://en.wikipedia.org/wiki/Sensitivity_and_specificity

A brief review of conditional probability

- What does “conditional probability” mean?
- Consider two events A and B
- $P(A)$ is the probability of event A occurring
- $P(A|B)$ is the probability of event A occurring, if it known that B has occurred (aka **conditional probability of $\boldsymbol{A}$ given $\boldsymbol{B}$**)
- Note that if A and B are **independent**, then $P(A|B) = P(A)$

Conditional probability definition

- $P(A|B) = \frac{P(A \cap B)}{P(B)}$ and $P(B|A) = \frac{P(B \cap A)}{P(A)}$
- By definition of joint probability, $P(A \cap B) = P(B \cap A)$

→ $P(A|B)P(B) = P(B|A)P(A)$

→ $P(A|B) = \frac{P(A)P(B|A)}{P(B)}$

Bayes' theorem

- $\boldsymbol{P(A|B) = \frac{P(A)P(B|A)}{P(B)}}$

- At this point, it is useful to remember that

$$P(B) = \sum_{i=1}^{n} P(A_i)P(B|A_i)$$

- Which allows us to reformulate the theorem as

$$P(A|B) = \frac{P(A)P(B|A)}{\sum_{i=1}^{n} P(A_i)P(B|A_i)}$$

Bayes theorem’s - II

- Why does the paper go to the trouble of deriving this formulation of the theorem?

$$P(A|B) = \frac{P(A)P(B|A)}{\sum_{i=1}^{n} P(A_i)P(B|A_i)}$$

- Because it allows us to easily calculate **posterior probabilities** from the characteristics of a detector!

Posterior probabilities

- It is typically easy to know the **likelihood** of a particular detector outcome (positive or negative) given the label of an object (malicious or benign)
- What is of interest here however (at least in the scope of the paper) are **posterior probabilities**, e.g.:
  - **What is the probability of an object being malicious given that it was labeled so?**
- Bayes' formulation given in the previous slide allows us to do so
- Let's come back to this later…

Rates vs Probabilities

- Typically, a detection algorithm is evaluated on a **test dataset**
- The results (e.g., true positive rate, false positive rate) are interpreted as a **probabilistic model of classifier behavior on unseen samples**
- E.g.:
  - $TPR = 90\% \leftrightarrow P(Alarm|Intrusion) = 0.9$
  - $FPR = 1\% \leftrightarrow P(Alarm|\neg Intrusion) = 0.01$

*(**Disclaimer:** I am going to use rates and probabilities interchangeably, but be fully aware that this is mathematically untoward w/o a philosophical discussion of the frequentist interpretation of probability)*

Now, a thought experiment…

- (From http://news.bbc.co.uk/2/hi/uk_news/magazine/8153539.stm)
- Imagine you have invented a scanner to detect terrorists at airports:

Positive (bad!)

Negative (good!)

- The scanner has an **accuracy of 90%**
- For simplicity, let's assume this means:

$$P(Terrorist|Alarm) = P(\neg Terrorist|\neg Alarm) = 0.9$$

A thought experiment - 2

- You are demonstrating your invention at an event in the house of parliament, in which 3000 people take part.
- Suddenly, you receive reliable information that a terrorist is attending the event:
  - A perfect opportunity to test your machine!
- Security seals all the exit and asks everyone to go through the scanner...

A thought experiment - 3

- Suddenly, as a man goes through the scanner, the scanner goes off! Police stops him at gunpoint.
- **Q:** What are the chances this man is a terrorist?
  - (a) 90%
  - (b) 10%
  - (c) 0.3%
- **A:** (c) 0.3%

A thought experiment - 4

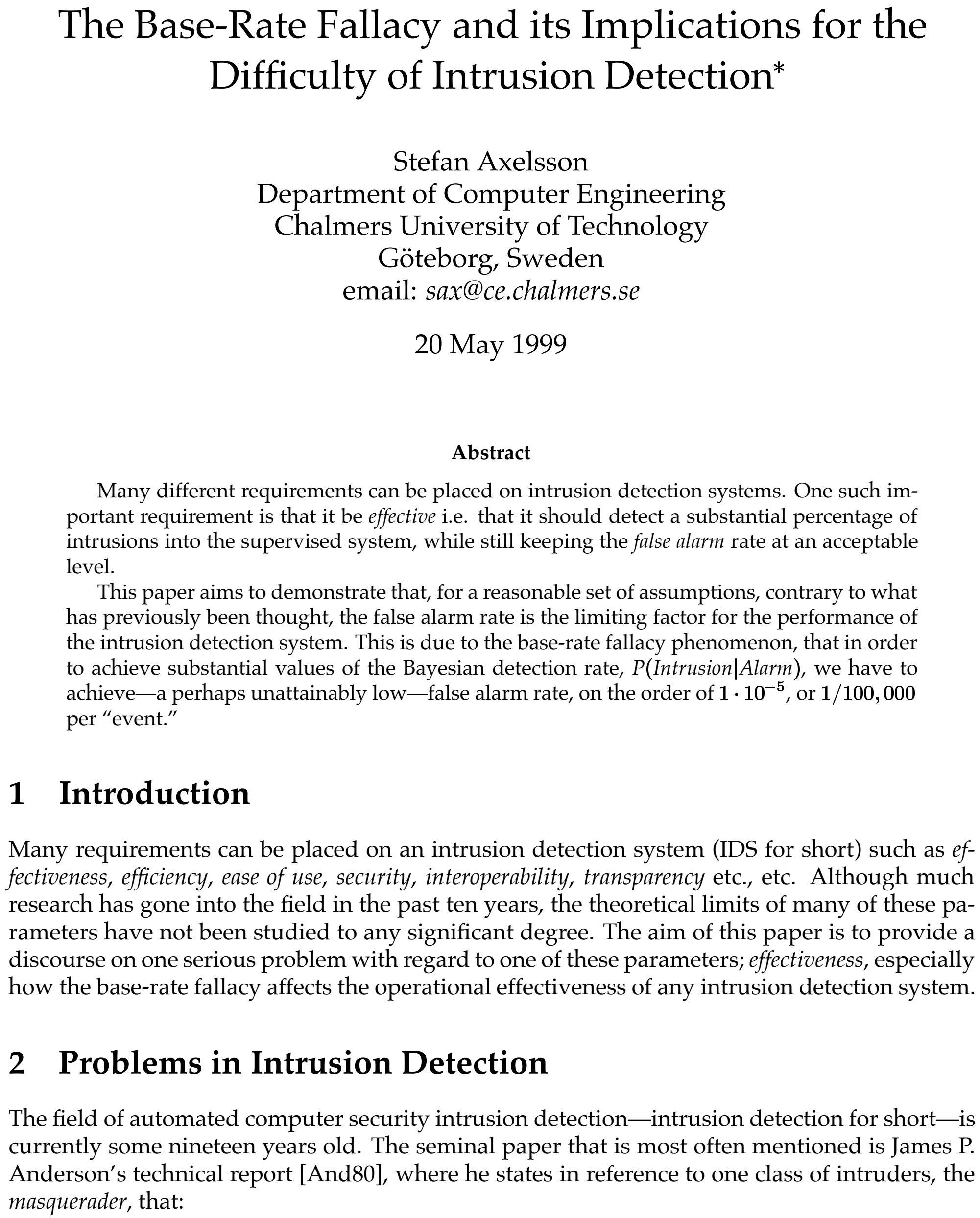
- What the experiment is really asking is: what is the **posterior probability** that a person is a terrorist, given knowledge that they were labeled as such?
- We can calculate it… we have the technology!
- Let's call T the event "subject is a terrorist" and A the event "detector raises an alarm"

$$P(T|A) = \frac{P(A|T)P(T)}{P(A|T)P(T) + P(A|\neg T)P(\neg T)} =$$

$$= \frac{0.9 \cdot ({}^{1}/_{3000})}{0.9 \cdot ({}^{1}/_{3000}) + 0.1 \cdot ({}^{2999}/_{3000})} = 0.003$$

Why is this experiment relevant?

- It forces us to reflect on the problems we face when attempting to detect **rare events**…
- And **most attacks in computer security are rare events**!
  - (Note: ”rare” refers to the ratio of attack events to benign events)



The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection*

Stefan Axelsson
Department of Computer Engineering
Chalmers University of Technology
Göteborg, Sweden
email: *sax@ce.chalmers.se*

20 May 1999

Abstract

Many different requirements can be placed on intrusion detection systems. One such important requirement is that it be *effective* i.e. that it should detect a substantial percentage of intrusions into the supervised system, while still keeping the *false alarm* rate at an acceptable level.

This paper aims to demonstrate that, for a reasonable set of assumptions, contrary to what has previously been thought, the false alarm rate is the limiting factor for the performance of the intrusion detection system. This is due to the base-rate fallacy phenomenon, that in order to achieve substantial values of the Bayesian detection rate, $P(Intrusion|Alarm)$, we have to achieve—a perhaps unattainably low—false alarm rate, on the order of $1 \cdot 10^{-5}$, or $1/100,000$ per "event."

1 Introduction

Many requirements can be placed on an intrusion detection system (IDS for short) such as *effectiveness*, *efficiency*, *ease of use*, *security*, *interoperability*, *transparency* etc., etc. Although much research has gone into the field in the past ten years, the theoretical limits of many of these parameters have not been studied to any significant degree. The aim of this paper is to provide a discourse on one serious problem with regard to one of these parameters; *effectiveness*, especially how the base-rate fallacy affects the operational effectiveness of any intrusion detection system.

2 Problems in Intrusion Detection

The field of automated computer security intrusion detection—intrusion detection for short—is currently some nineteen years old. The seminal paper that is most often mentioned is James P. Anderson's technical report [And80], where he states in reference to one class of intruders, the *masquerader*, that:

About the paper

- When this paper was published, the field of network intrusion detection was already ~10-year old
- However, as a largely applied line of research, there was **a lack of meta-analysis on the underlying assumptions and limits of detection techniques**
- This paper introduces a conversation about the **limits of intrusion detection** in the academic discourse
- The issues brought up by this paper are not easily solved (and in some cases, probably not solvable at all!)

Concepts from the paper

- **Example scenario:** small computer network
    - Intrusion detection is performed by analyzing logs of networks and host events
    - 1,000,000 log events/day; but each intrusion requires analysis of 10 events – so there are 100,000 objects to be analyzed per day
    - 2 intrusion attempts per day
    - 1 SSO (site security officer; network administrator)
    - Reasonable assumptions!

Concepts from the paper - 2

- Events of interest:
  - A: the intrusion detection system raises an alarm
  - I: an actual intrusion attempt occurs
- Given probabilities of interest:
  - $P(A|I)$: probability of alarm given an intrusion (**true positive rate**)
  - $P(A|\neg I)$: probability of alarm given no intrusion (**false positive rate**)
  - $P(\neg A|I)$: probability of no alarm given an intrusion (**false negative rate**)
  - $P(\neg A|\neg I)$: probability of no alarm given no intrusion (**true negative rate**)

Concepts from the paper - 3

- Given these probabilities, we are interested in computing various posterior probabilities using Bayes' theorem:
- **Bayesian detection rate:** $P(I|A)$: the probability that an alarm really indicates an intrusion
- $P(\neg I|\neg A)$: the probability that absence of alarm indicates that no intrusion is occurring
- A good design should attempt to **maximize both probabilities**

Concepts from the paper - 4

- Let’s compute the **Bayesian detection rate** for the example scenario (2 intrusions per day, 100,000 entries generated per day):

$$P(I|A) = \frac{P(I)P(A|I)}{P(I)P(A|I) + P(\neg I)P(A|\neg I)} =$$

$$= \frac{2 \cdot 10^{-5} P(A|I)}{2 \cdot 10^{-5} P(A|I) + 0.99998 \cdot P(A|\neg I)}$$

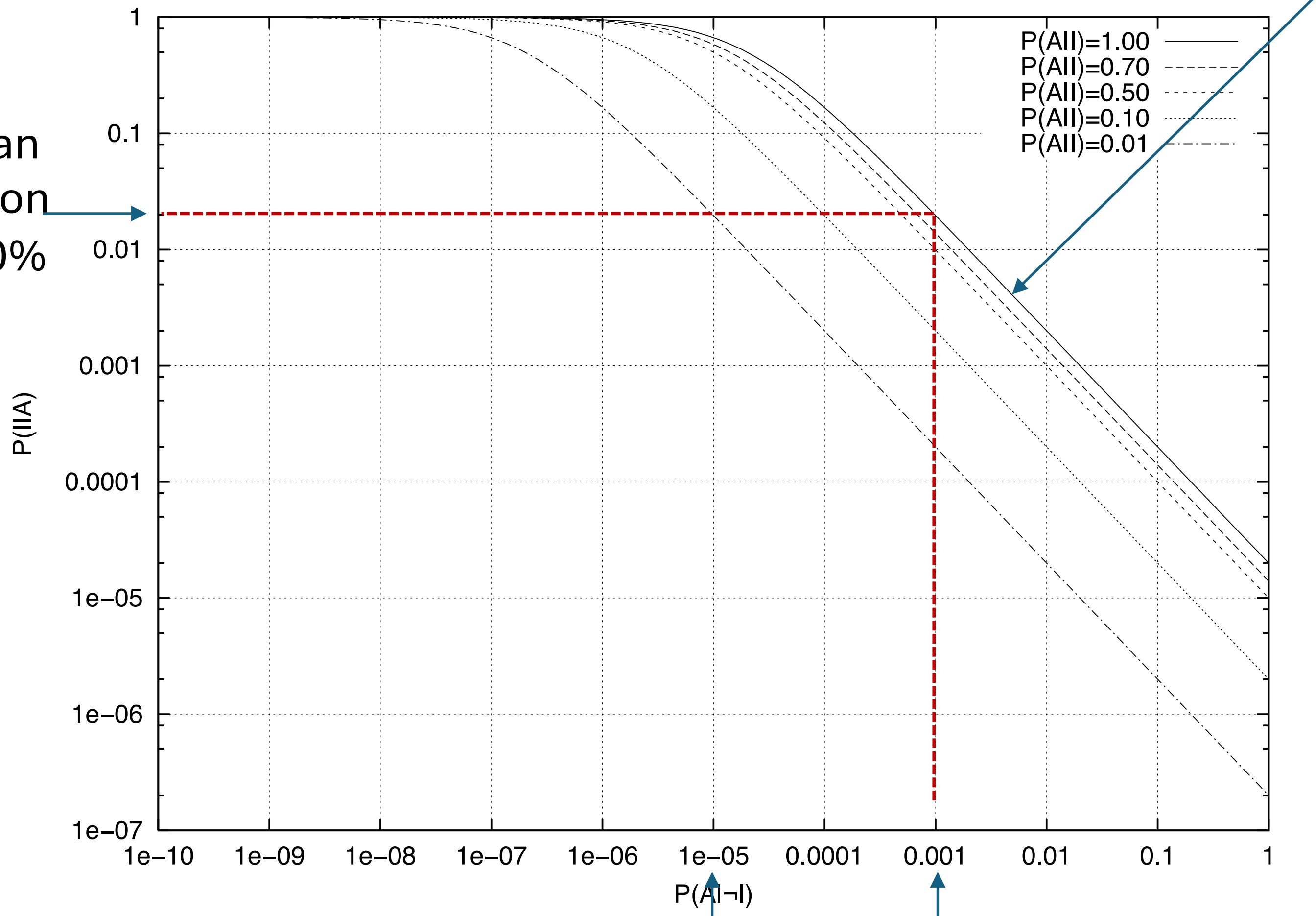
The BRF in a nutshell!

$$P(I|A) = \frac{P(I)P(A|I)}{P(I)P(A|I) + P(\neg I)P(A|\neg I)} =$$

$$= \frac{2 \cdot 10^{-5} P(A|I)}{2 \cdot 10^{-5} P(A|I) + \underbrace{0.99998 \cdot P(A|\neg I)}}$$

Even if $P(A|I)$ is orders of magnitude larger than $P(A|\neg I)$, this term still dominates the denominator and causes $P(I|A) \longrightarrow 0$ (**why?**)

Some concrete results



Let’s assume that our detector has 100% TP rate…

Bayesian detection rate: 20%

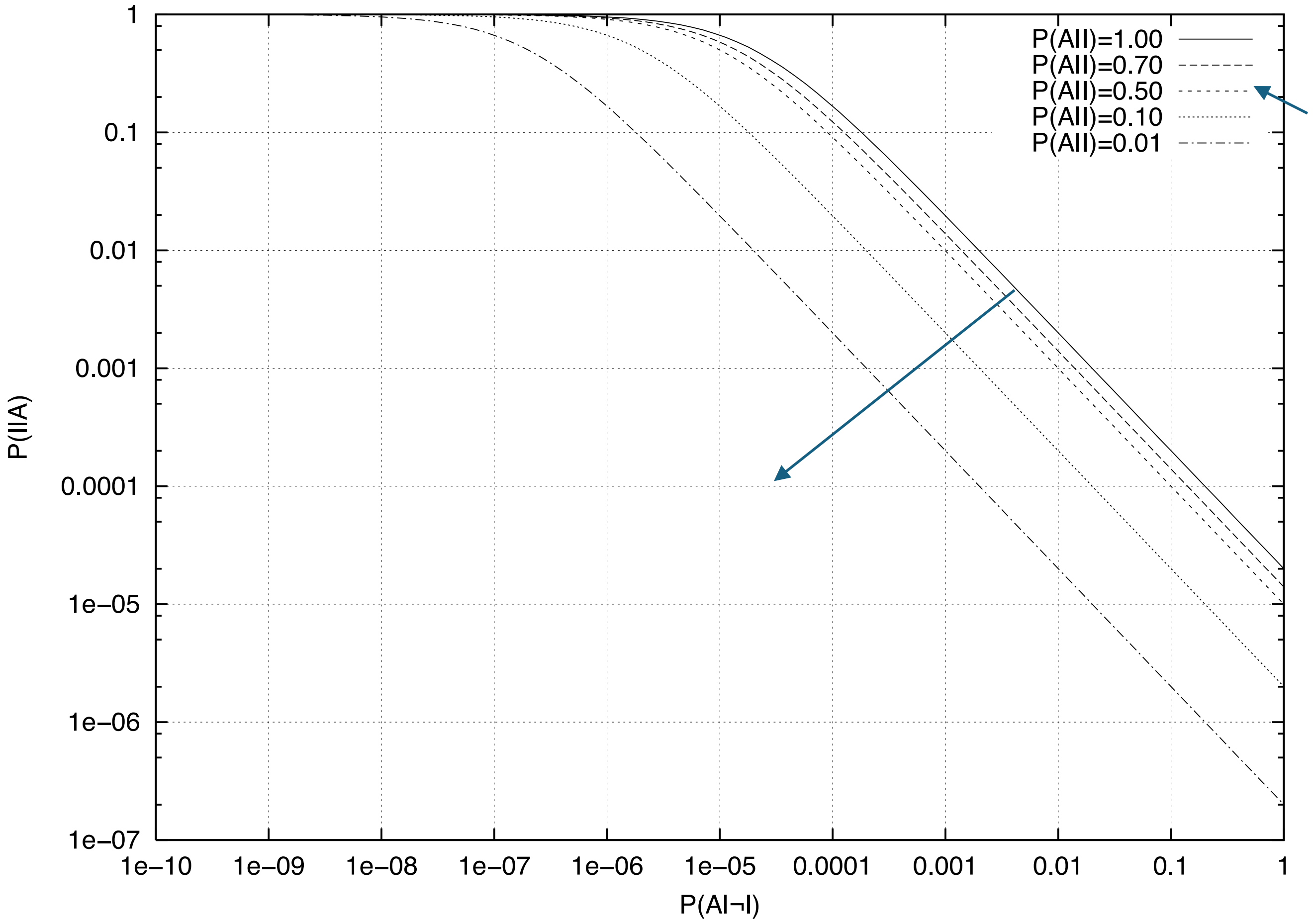
1 false alarm per day

100 false alarms per day (more realistic; still manageable)

Some concrete results - II

- Is a Bayesian detection rate of 20% good or bad?
  - Remember, the Bayesian detection rate represents the probability of an alarm representing an actual intrusion
- If a security officer dealing with 100 alarms per day were to investigate every alarm, she would need to **review an alarm every 5 minutes**
  - Would you do it, knowing that most of them are false alarms?

Some concrete results - III



What if P(A|I) is, realistically, < 1.0?

The situation gets even worse! (**why?**)

What are the implications?

- Why is it called “base-rate fallacy”?
- In considering an instrument to detect something harmful, we naturally focus on the ability of the instrument to alert us **when something harmful is indeed present**
- However, we fail to consider that in most scenarios **harmful situations occur rarely**

What are the implications? - II

- “Base-rate fallacy” represents a **failure to consider the rate at which the (harmful) situation of interest occurs**
- Since most of the time the instrument is analyzing something not harmful, the **probability of false alarm** becomes very important
- Unless the probability of false alarm can be kept extremely low, the instrument is, for all purposes, useless

The BRF and intrusion detection

- ”Network intrusion detection” refers to the task of detecting attempts to gain unauthorized access to a computer network
- Two main approaches:
  - Detect flows/processes/users/… whose characteristics differ from what is normally observed in the system (**anomaly-based intrusion detection**)
  - Detect flows/processes/users/… whose characteristics match an explicitly-specified attack signature or directly violates a policy (**specification- or policy-based i.d.**)
- **The BRF applies to both!**

The BRF and intrusion detection - II

- Where do false positives come from?
- **Anomaly-based** intrusion detection?
  - Regardless of the specific parameters being modeled, it is difficult to come up with statistical parameters that can reliably ignore benign actions and always identify intrusion situations (networks are complex, unpredictable objects)
- **Specification-/Policy-based** intrusion detection?
  - Attack signatures may lack specificity and match benign entities (e.g. a regular expression which captures attack packets but also some benign packets); policies may fail to represent how the network is actually used

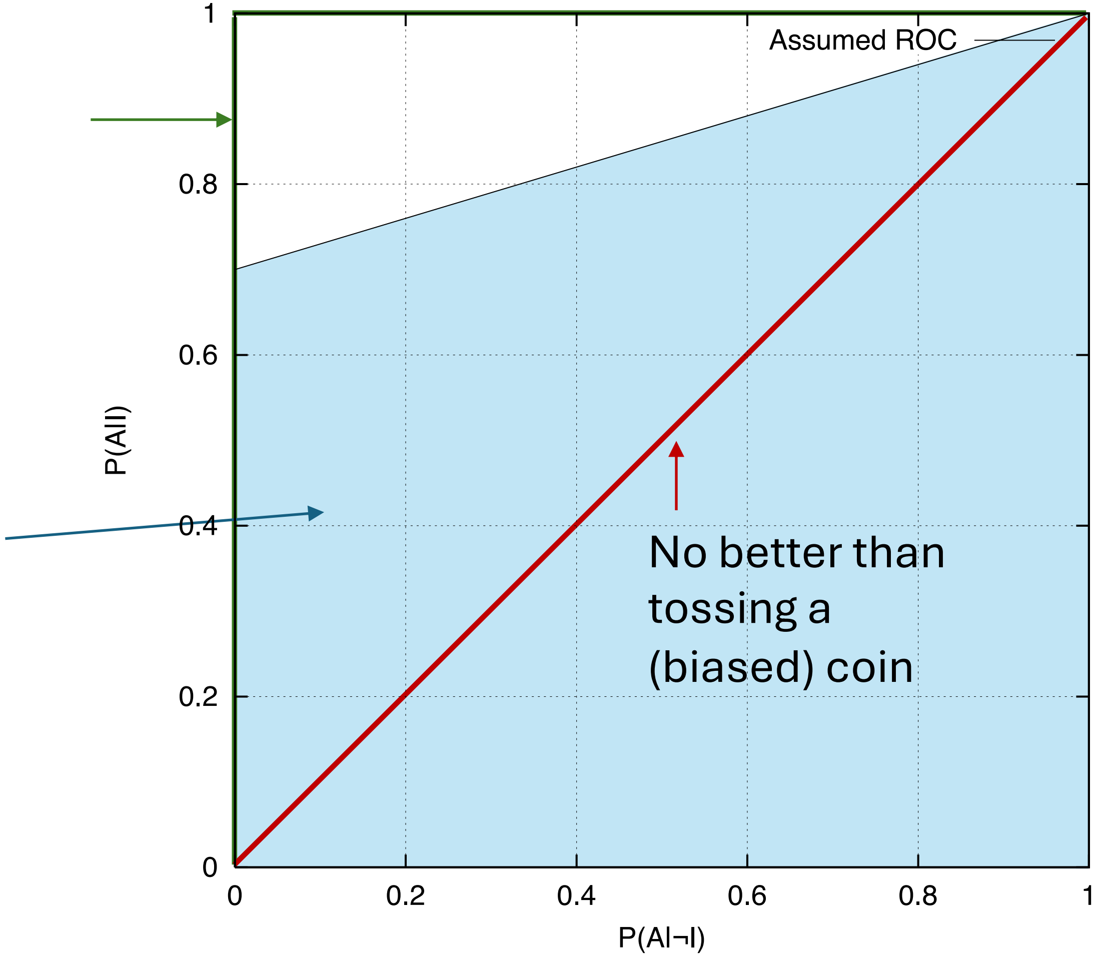
The BRF and intrusion detection - III

- The base-rate fallacy **does not provide guidance** on how to design intrusion detection systems
- It merely points out a **pitfall** for nearly every possible intrusion detection system
- Ignoring it most likely leads to an unusable system!

Evaluating trade-offs: ROC curves

- Designing a detector entails a trade-off between the **false positive rate** $P(A|\neg I)$ and the **true positive rate** $P(A|I)$
- Typically, **tuning a detector to achieve a higher true positive rate also causes it to exhibit a higher false positive rate**
- In other words, making a detector more sensitive to attacks also makes it more prone to consider benign situations as attacks
- A ROC curve allows to evaluate this trade-off

ROC curve: example



Perfect classifier

The area under the ROC curve (AUC) is oftentimes used as measure of the quality of a classifier

No better than tossing a (biased) coin

Dealing with the base-rate fallacy

- Each of you should think of two ways to reduce the impact of the base-rate fallacy

Dealing with the base-rate fallacy

- Designing classifiers with high AUC helps, but to a point
- **Possible approaches:**
    - Consider multiple independent intrusion signals
    - Merge similar alerts
    - Formulate a notion of classifier confidence and deal with clear-cut attacks automatically (only have human expert review ambiguous cases)
    - Assume intrusion will happen and have safeguards in place

In conclusion…

- The base-rate fallacy clarifies that **false positives are arguably a more sever problem than false negatives** for intrusion detection
- Design of algorithms for identifying intrusion attempts and malicious objects **must take the problem into account**
- **Several palliatives, but no full solution**