Intrusion Detection Systems: Core Elements

Kostas Papagiannopoulos
University of Amsterdam
kostaspap88@gmail.com //kpcrypto.net

Contents

Introduction

Introduction

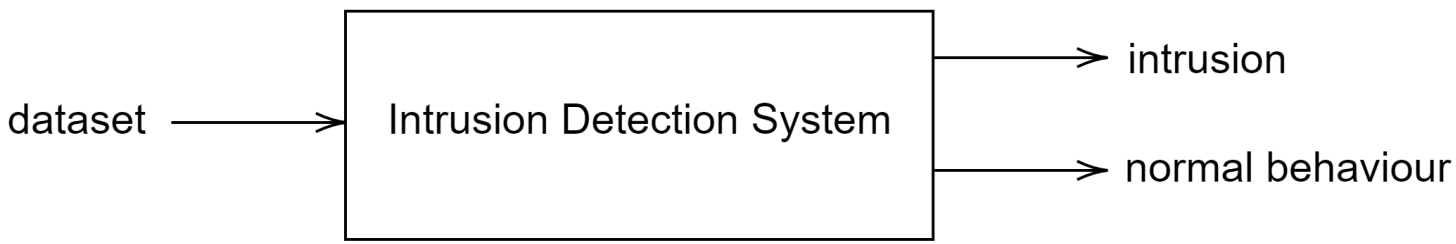
- **Intrusion** is any violation of the security policy of a system or network, such as a breach of confidentiality, integrity or availability of the system. It can originate from an external or internal adversary.

Introduction

- **Intrusion** is any violation of the security policy of a system or network, such as a breach of confidentiality, integrity or availability of the system. It can originate from an external or internal adversary.
- **Intrusion detection** is the process of collecting data about a computer system or network and analyzing them for signs of intrusion

Introduction

- **Intrusion** is any violation of the security policy of a system or network, such as a breach of confidentiality, integrity or availability of the system. It can originate from an external or internal adversary.
- **Intrusion detection** is the process of collecting data about a computer system or network and analyzing them for signs of intrusion
- **Intrusion Detection System (IDS)** is a software or hardware component that analyzes data and decides whether the system behavior is normal or whether an intrusion is happening



Introduction

IDS categories

- **Host-based** IDSs monitor a device directly for suspicious activity and are able to get a view of its internals

Introduction

IDS categories

- **Host-based** IDSs monitor a device directly for suspicious activity and are able to get a view of its internals
- **Network-based** IDSs monitor the activity of a larger network segment that may offer a bigger picture but could lack particular device information

Introduction

IDS categories

- **Host-based** IDSs monitor a device directly for suspicious activity and are able to get a view of its internals
- **Network-based** IDSs monitor the activity of a larger network segment that may offer a bigger picture but could lack particular device information

  e.g. the antivirus software that monitors a computer is a host-based IDS

  e.g. the sysadmin going over all network logs looking for suspicious behavior is a network-based IDS

Introduction

IDS categories

- **Host-based** IDSs monitor a device directly for suspicious activity and are able to get a view of its internals
- **Network-based** IDSs monitor the activity of a larger network segment that may offer a bigger picture but could lack particular device information

  e.g. the antivirus software that monitors a computer is a host-based IDS

  e.g. the sysadmin going over all network logs looking for suspicious behavior is a network-based IDS

- **Signature-based** IDSs use certain rules to detect an intrusion. They try to match their observations to a database of existing attack patterns.

Introduction

IDS categories 入侵

- **Host-based** IDSs monitor a device directly for suspicious activity and are able to get a view of its internals
- **Network-based** IDSs monitor the activity of a larger network segment that may offer a bigger picture but could lack particular device information

  e.g. the antivirus software that monitors a computer is a host-based IDS

  e.g. the sysadmin going over all network logs looking for suspicious behavior is a network-based IDS

- **Signature-based** IDSs use certain rules to detect an intrusion. They try to match their observations to a database of existing attack patterns.
- **Anomaly-based** IDSs try to model the normal system behaviour and then identifies divergences from normality in a probabilistic manner

Introduction

IDS categories

- **Host-based** IDSs monitor a device directly for suspicious activity and are able to get a view of its internals
- **Network-based** IDSs monitor the activity of a larger network segment that may offer a bigger picture but could lack particular device information

  e.g. the antivirus software that monitors a computer is a host-based IDS

  e.g. the sysadmin going over all network logs looking for suspicious behavior is a network-based IDS

- **Signature-based** IDSs use certain rules to detect an intrusion. They try to match their observations to a database of existing attack patterns.
- **Anomaly-based** IDSs try to model the normal system behaviour and then identifies divergences from normality in a probabilistic manner

  e.g. the antivirus software matching a binary file against a large database of known malware

  e.g. a machine learning model that profiles regular user behavior and tries to spot unexpected malicious behavior

Introduction

Requirements for an IDS

1. **Effectiveness**
   Does the IDS detect most intrusions, while keeping the number of false alarms reasonably low?
2. **Efficiency**
   What is the runtime performance? How much computing/storage resources does it need? Can it detect intrusions in real time? Is it resilient to stress?
3. **Ease of use and Flexibility**
   Can we operate it easily and deal quickly with alarms? Can we add new intrusion scenarios?

Introduction

Requirements for an IDS

4. **Security**
   Since it detects intrusions, the IDS is an attack target. Is it protected against attacks that try to disable it or circumvent it or tamper with it?
5. **Interoperability**
   Different IDSs may be deployed at different layers/part of a system or an organization. How can distributed IDSs collaborate while using different data formats and different detection algorithms? How do we secure their communication/coordination? Should we choose a centralized or decentralized architecture?
6. **Transparency and Privacy**
   How intrusive is the IDS? What type of data does it inspect and who has access to it? How long is the data stored?

Basic Elements of Intrusion Detection

IDS Basics

Example: Training and testing a supervised IDS

We are developing an IDS that monitors the time duration of TCP connections in our internal network. That is, long connection times indicate potential intrusions.

Training the IDS. Using the connection duration metric, we have captured two training datasets, one labeled as normal network behavior $\mathcal{D}_{\neg I}$ and labelled as abnormal network behavior $\mathcal{D}_I$

IDS Basics

Example: Training and testing a supervised IDS

We are developing an IDS that monitors the time duration of TCP connections in our internal network. That is, long connection times indicate potential intrusions.

Training the IDS. Using the connection duration metric, we have captured two training datasets, one labeled as normal network behavior $\mathcal{D}_{\neg I}$ and labelled as abnormal network behavior $\mathcal{D}_I$

- I stands for **intrusion** and $\neg I$ stands for **no intrusion**
- t_i^{label} stands for the connection time measurement number i with/without intrusion i.e. label $\in \{\neg I, I\}$

IDS Basics

Example: Training and testing a supervised IDS

We are developing an IDS that monitors the time duration of TCP connections in our internal network. That is, long connection times indicate potential intrusions.

Training the IDS. Using the connection duration metric, we have captured two training datasets, one labeled as normal network behavior $\mathcal{D}_{\neg I}$ and labelled as abnormal network behavior $\mathcal{D}_I$

- I stands for **intrusion** and $\neg I$ stands for **no intrusion**
- t_i^{label} stands for the connection time measurement number i with/without intrusion i.e. label $\in \{\neg I, I\}$

- $\mathcal{D}_{\neg I} = \{t_1^{\neg I}, t_2^{\neg I}, \ldots, t_n^{\neg I}\}$ is a train dataset that contains n connection timings that correspond to regular TCP connections $(\neg I)$
- $\mathcal{D}_I = \{t_1^I, t_2^I, \ldots, t_m^I\}$ is a train dataset that contains m connection timings that correspond to intrusive TCP connections (I)

- We will use the labeled datasets $\mathcal{D}_I, \mathcal{D}_{\neg I}$ to create an **anomaly-based IDS**

IDS Basics

- We assume that we can use a normal distribution to model the intrusions and the normal behavior

IDS Basics

- We assume that we can use a normal distribution to model the intrusions and the normal behavior
- We compute the mean μ and variance σ^2 for datasets $\mathcal{D}_{\neg I}, \mathcal{D}_I$

mean, std for no intrusion

$$\mu_{\neg I} = \frac{1}{n}\sum_{i=1}^{n} t_i^{\neg I} \qquad \sigma^2_{\neg I} = \frac{1}{n-1}\sum_{i=1}^{n}(t_i^{\neg I} - \mu_{\neg I})^2$$

mean, std for intrusion

$$\mu_I = \frac{1}{m}\sum_{i=1}^{m} t_i^{I} \qquad \sigma^2_{I} = \frac{1}{m-1}\sum_{i=1}^{m}(t_i^{I} - \mu_{I})^2$$

IDS Basics

- We assume that we can use a normal distribution to model the intrusions and the normal behavior
- We compute the mean μ and variance σ^2 for datasets $\mathcal{D}_{\neg I}, \mathcal{D}_I$

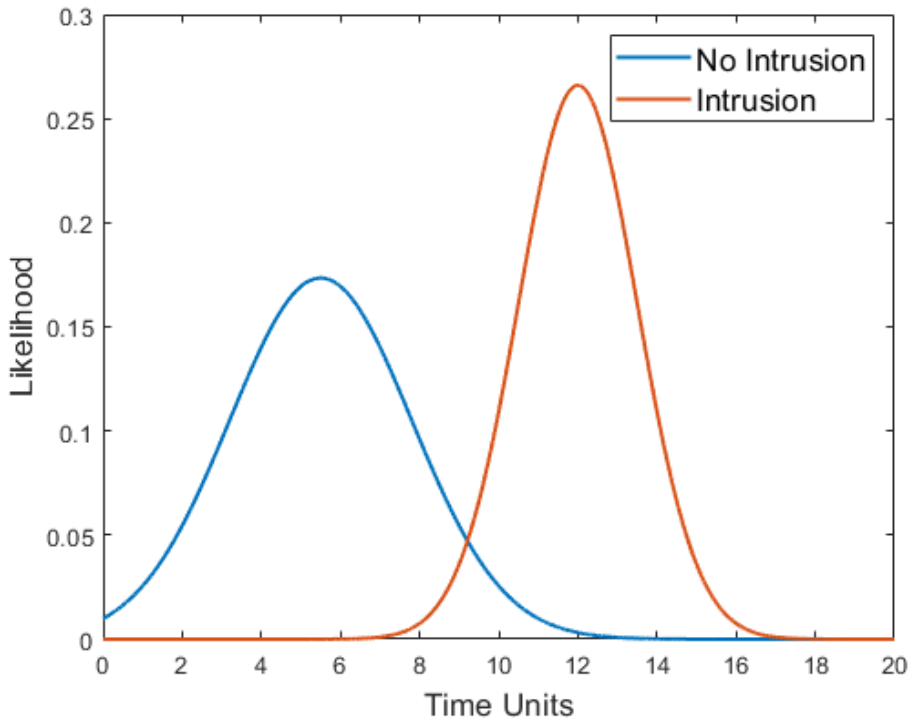
$$\mu_{\neg I} = \frac{1}{n}\sum_{i=1}^{n} t_i^{\neg I} \qquad \sigma^2_{\neg I} = \frac{1}{n-1}\sum_{i=1}^{n}(t_i^{\neg I} - \mu_{\neg I})^2$$

$$\mu_I = \frac{1}{m}\sum_{i=1}^{m} t_i^{I} \qquad \sigma^2_{I} = \frac{1}{m-1}\sum_{i=1}^{m}(t_i^{I} - \mu_{I})^2$$

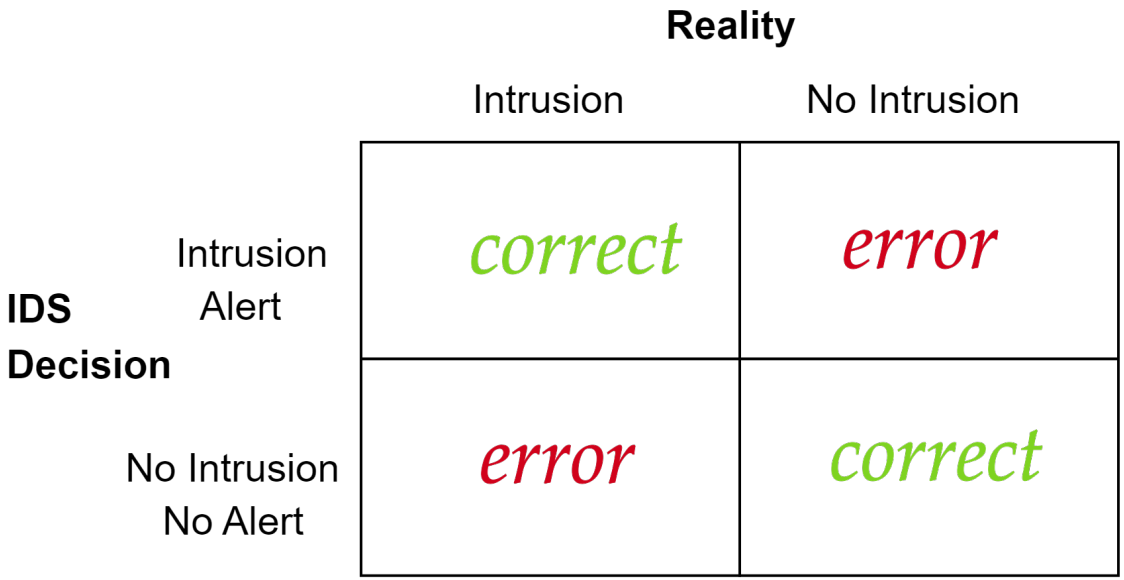
- The IDS consists of two normal distributions: $\mathcal{N}(\mu_{\neg I}, \sigma^2_{\neg I})$ and $\mathcal{N}(\mu_I, \sigma^2_I)$

IDS Basics

- Observe that the two distributions may overlap
- The IDS cannot always distinguish 'no intrusion' from 'intrusion'
- What types of mistakes can happen?

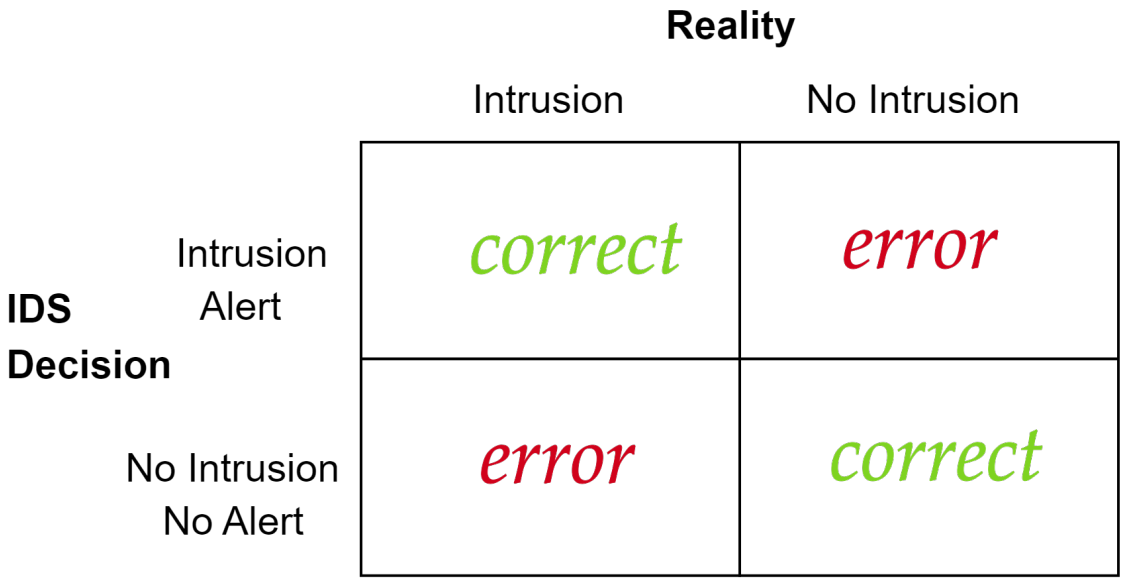


IDS Basics



1. Correct: decide 'intrusion' and raise an alert when there is an intrusion
2. Correct: decide 'no intrusion' when there is no intrusion

IDS Basics



1. Correct: decide 'intrusion' and raise an alert when there is an intrusion
2. Correct: decide 'no intrusion' when there is no intrusion
3. Error: decide 'intrusion' and raise an alert when there is no real intrusion
4. Error: decide 'no intrusion' when there is actually an intrusion

IDS Basics

positive → detect intrusion

IDS Decision	Reality: Intrusion	Reality: No Intrusion
Intrusion Alert	*True Positive*	*False Positive*
No Intrusion No Alert	*False Negative*	*True Negative*

false positive.

should be true detect as false.

1. True Positive TP
   $TP = P(\text{'alert'}|\text{'intrusion'}) = P(A|I)$

2. True Negative TN
   $TN = P(\text{'no alert'}|\text{'no intrusion'}) = P(\neg A|\neg I)$

3. False Positive FP
   $TP = P(\text{'alert'}|\text{'no intrusion'}) = P(A|\neg I)$

4. False Negative FN
   $FN = P(\text{'no alert'}|\text{'intrusion'}) = P(\neg A|I)$

IDS Basics

Testing the IDS. We have now captured two test/validation datasets: one for normal network behavior $\mathcal{T}_{\neg I}$ and one for abnormal network behavior $\mathcal{T}_I$

- $\mathcal{T}_{\neg I} = \{v_1^{\neg I}, v_2^{\neg I}, \ldots, v_w^{\neg I}\}$ is a test dataset that contains w connection timings that correspond to regular TCP connections ($\neg I$)
- $\mathcal{T}_I = \{v_1^I, v_2^I, \ldots, v_q^I\}$ is a test dataset that contains q connection timings that correspond to intrusive TCP connections (I)

IDS Basics

Testing the IDS. We have now captured two test/validation datasets: one for normal network behavior $\mathcal{T}_{\neg I}$ and one for abnormal network behavior $\mathcal{T}_I$

- $\mathcal{T}_{\neg I} = \{v_1^{\neg I}, v_2^{\neg I}, \ldots, v_w^{\neg I}\}$ is a test dataset that contains w connection timings that correspond to regular TCP connections ($\neg I$)
- $\mathcal{T}_I = \{v_1^I, v_2^I, \ldots, v_q^I\}$ is a test dataset that contains q connection timings that correspond to intrusive TCP connections (I)

- The IDS will use a likelihood score to identify intrusions based on the probability density function (pdf) of the normal distribution

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}$$

IDS Basics

Input: testset $\mathcal{T}_{\neg I}$
Input: distributions $\mathcal{N}(\mu_{\neg I}, \sigma^2_{\neg I})$, $\mathcal{N}(\mu_I, \sigma^2_I)$
Output: TN, FP rates

$TNcount = 0$; $FPcount = 0$
// iterate over the elements of the testset
for i=1 *until* w **do**

  // compute likelihood scores
  $score_{\neg I} = pdf(v_i^{\neg I}, \mu_{\neg I}, \sigma_{\neg I})$
  $score_I = pdf(v_i^{\neg I}, \mu_I, \sigma_I)$

  // count match/mismatch
  if $score_{\neg I} > score_I$ **then**
    $TNcount = TNcount + 1$
  else
    $FPcount = FPcount + 1$
  end
end
// compute the rates
$TN = TNcount/w$; $FP = FPcount/w$

w number of data sets.

IDS Basics

Input: testset $\mathcal{T}_l$
Input: distributions $\mathcal{N}(\mu_{\neg l}, \sigma^2_{\neg l})$, $\mathcal{N}(\mu_l, \sigma^2_l)$
Output: TP, FN rates

$TPcount = 0$; $FNcount = 0$
// iterate over the elements of the testset
for i=1 *until* q **do**

  // compute likelihood scores
  $score_{\neg l} = pdf(v_i^l, \mu_{\neg l}, \sigma_{\neg l})$
  $score_l = pdf(v_i^l, \mu_l, \sigma_l)$

  // count match/mismatch
  if $score_{\neg l} < score_l$ **then**
    $TPcount = TPcount + 1$
  else
    $FNcount = FNcount + 1$
  end
end
// compute the rates
$TP = TPcount/q$; $FN = FNcount/q$

IDS Basics

- We typically use the rates (TP, FP) to describe the performance of the IDS
- Additional metrics are available:

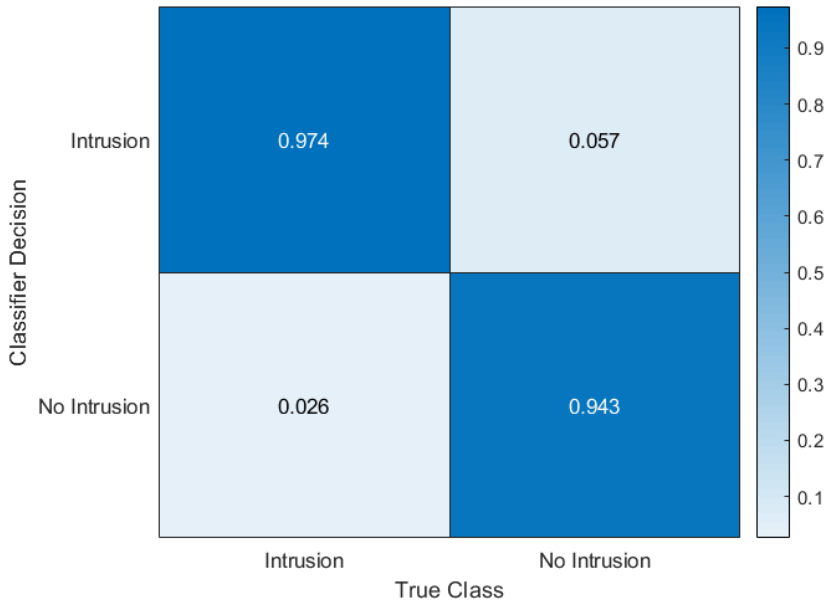
$$\text{precision} = \frac{TP}{TP + FP} \qquad \text{recall} = \frac{TP}{q}$$

$$\text{F-score} = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \qquad \text{specificity} = \frac{TN}{FP + TN}$$

$$\text{accuracy} = \frac{TP + TN}{q + w}$$

IDS Basics

- We can also create a confusion matrix that visualizes TP, TN, FP, FN



- Verify these steps using MATLAB code: `normal_ids.m`

IDS Basics

Example: Training and testing an IDS using only the normal behavior

So far we treated the IDS using a *supervised* learning approach with 2 classes ($\neg I$, I)

- This is recommended when the anomalies are plenty and well-known
  - e.g. a spam filter

IDS Basics

Example: Training and testing an IDS using only the normal behavior

So far we treated the IDS using a *supervised* learning approach with 2 classes ($\neg I$, I)

- This is recommended when the anomalies are plenty and well-known
  - e.g. a spam filter
- What if the intrusions are extremely rare events?
  - e.g. an airplane failure detection system may not have enough data to train a model for class I

IDS Basics

Example: Training and testing an IDS using only the normal behavior

So far we treated the IDS using a *supervised* learning approach with 2 classes ($\neg I$, I)

- This is recommended when the anomalies are plenty and well-known
  - e.g. a spam filter
- What if the intrusions are extremely rare events?
  - e.g. an airplane failure detection system may not have enough data to train a model for class I
- What if we don't know how to model all types of intrusions?
  - e.g. we want our airplane failure detection system to detect unforeseen problems

Thus many IDSs follow an approach that uses only a single class i.e. the 'normal' class ($\neg I$) for training

IDS Basics

Training the IDS. Using the connection duration metric, we now have only one dataset, labeled as normal network behavior $\mathcal{D}_{\neg I}$

IDS Basics

Training the IDS. Using the connection duration metric, we now have only one dataset, labeled as normal network behavior $\mathcal{D}_{\neg I}$

- We compute the mean μ and variance σ^2 for dataset $\mathcal{D}_{\neg I}$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} t_i^{\neg I} \qquad \sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(t_i^{\neg I} - \mu)^2$$

- We set a threshold *th* that delimits normal behavior from intrusions
- The IDS now consists of a normal distribution $\mathcal{N}(\mu, \sigma^2)$ and a specified threshold *th*

IDS Basics

Training the IDS. Using the connection duration metric, we now have only one dataset, labeled as normal network behavior $\mathcal{D}_{\neg I}$

- We compute the mean μ and variance σ^2 for dataset $\mathcal{D}_{\neg I}$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} t_i^{\neg I} \qquad \sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(t_i^{\neg I} - \mu)^2$$

- We set a threshold *th* that delimits normal behavior from intrusions
- The IDS now consists of a normal distribution $\mathcal{N}(\mu, \sigma^2)$ and a specified threshold *th*

Testing the IDS. Like before we have two test/validation datasets: one for normal network behavior $\mathcal{T}_{\neg I}$ and one for abnormal network behavior $\mathcal{T}_I$

- The IDS will use a likelihood score to identify intrusions based on the probability density function (pdf) of $\mathcal{N}(\mu, \sigma)$ and the threshold *th*

IDS Basics

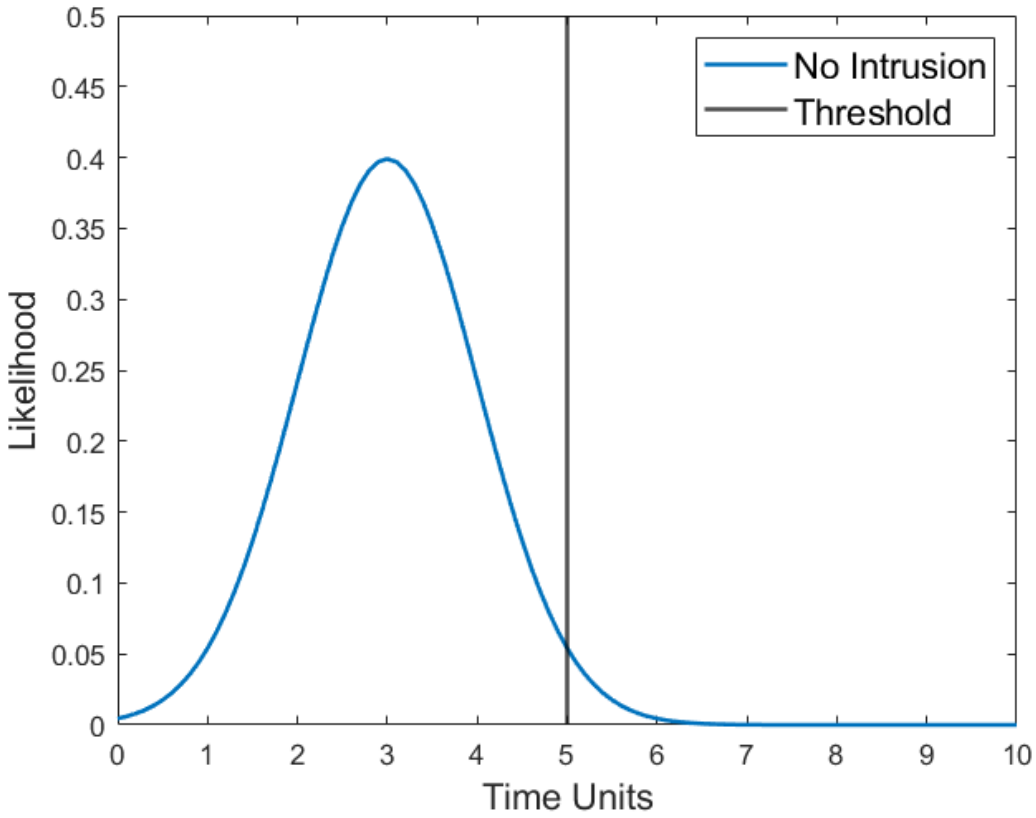
- We have trained a model for the 'normal' behavior $\mathcal{N}(\mu = 3, \sigma = 1)$
- We have set the threshold $th = 5$

IDS Basics

- We have trained a model for the 'normal' behavior $\mathcal{N}(\mu = 3, \sigma = 1)$
- We have set the threshold $th = 5$
- Any score $\geq th$ is classified as an intrusion
- Any score $< th$ is classified as 'no intrusion'

IDS Basics

- We have trained a model for the 'normal' behavior $\mathcal{N}(\mu = 3, \sigma = 1)$
- We have set the threshold $th = 5$
- Any score $\geq th$ is classified as an intrusion
- Any score $< th$ is classified as 'no intrusion'



IDS Basics

Input: testset $\mathcal{T}_{\neg I}$
Input: distribution $\mathcal{N}(\mu, \sigma^2)$, threshold th
Output: TN, FP rates

$TNcount = 0$; $FPcount = 0$
// iterate over the elements of the testset
for i=1 *until* w **do**

  // compute likelihood score
  $score = pdf(v_i^{\neg I}, \mu, \sigma)$
  // count match/mismatch
  if $score < th$ **then**
    $TNcount = TNcount + 1$
  else
    $FPcount = FPcount + 1$
  end

end
// compute the rates
$TN = TNcount/w$; $FP = FPcount/w$

IDS Basics

Input: testset $\mathcal{T}_l$
Input: distribution $\mathcal{N}(\mu, \sigma^2)$, threshold *th*
Output: *TP*, *FN* rates

$TPcount = 0$; $FNcount = 0$
`// iterate over the elements of the testset`
for *i=1 until q* **do**

`// compute likelihood score`
$score = pdf(v_i^l, \mu, \sigma)$
`// count match/mismatch`
if $score > th$ **then**
$TPcount = TPcount + 1$
else
$FNcount = FNcount + 1$
end

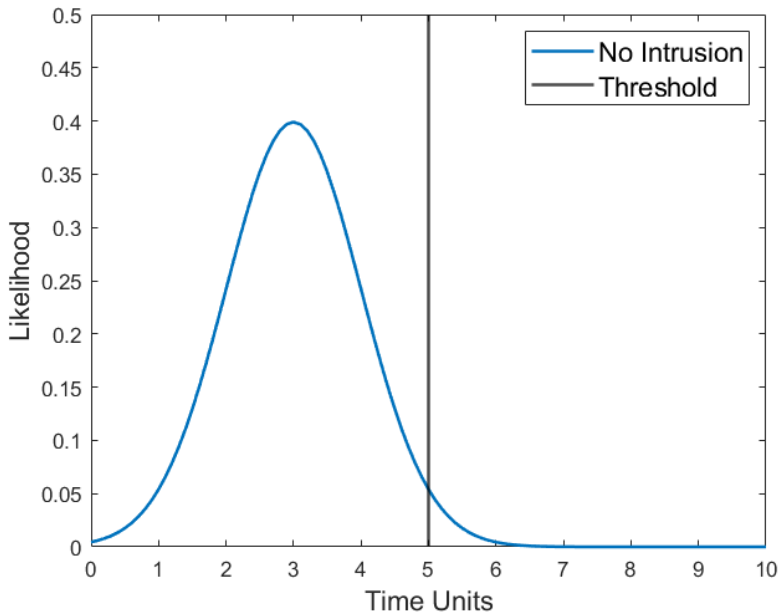
end
`// compute the rates`
$TP = TPcount/q$; $FN = FNcount/q$

- You can modify `normal_ids.m` to train the IDS with the 'normal' dataset only and then use a threshold during the test phase

IDS Basics

Example: Adjusting the threshold of an IDS

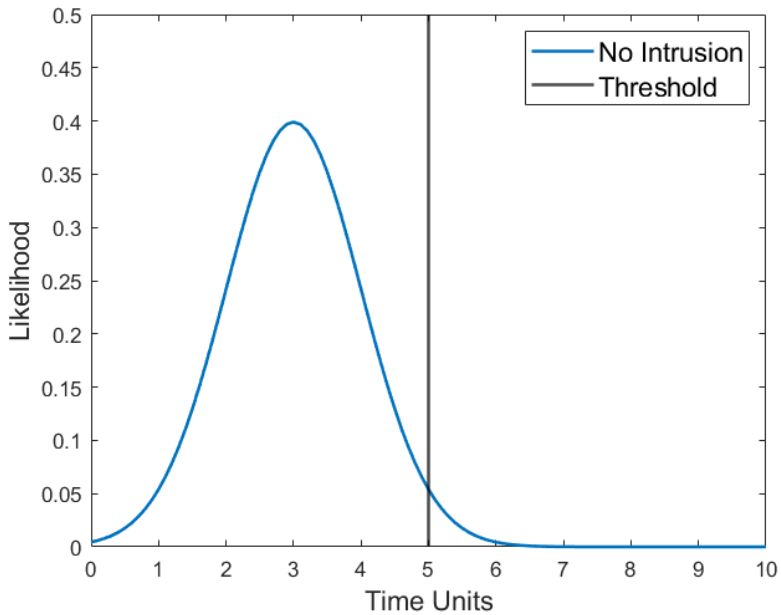
- Using the 'normal behavior' dataset we have trained a model $\mathcal{N}(\mu = 3, \sigma = 1)$
- We have set the threshold $th = 5$



IDS Basics

Example: Adjusting the threshold of an IDS

- Using the 'normal behavior' dataset we have trained a model $\mathcal{N}(\mu = 3, \sigma = 1)$
- We have set the threshold $th = 5$



1. Let r.v. $\mathbf{X} \sim \mathcal{N}(3, 1)$. Compute the false positive rate FP of the IDS

$$FP = P(\text{'alert'}|\text{'no intrusion'}) = P(A|\neg I) = P(\mathbf{X} \geq th) = 1 - P(\mathbf{X} < th) =$$

$$1 - P(\mathbf{X} \leq th) = 1 - cdf_{\mathcal{N}(3,1)}(5) = 1 \texttt{ - normcdf(5, 3, 1)} = 0.0228$$

IDS Basics

2. Find a new threshold th such that the false positive rate of the IDS is $FP = 0.01$

$$FP = 0.01 \iff P(A|\neg I) = 0.01 \iff 1 - P(\mathbf{X} \leq th) = 0.01 \iff$$

$$P(\mathbf{X} \leq th) = 0.99 \iff th = cdf^{-1}_{\mathcal{N}(3,1)}(0.99) \iff$$

$$th = \texttt{norminv(0.99, 3, 1)} \iff th = 5.3263$$

Thus, to ensure a FP of 1% we must shift the threshold from 5 to 5.3263

IDS Basics

2. Find a new threshold th such that the false positive rate of the IDS is $FP = 0.01$

$$FP = 0.01 \iff P(A|\neg I) = 0.01 \iff 1 - P(\mathbf{X} \leq th) = 0.01 \iff$$

$$P(\mathbf{X} \leq th) = 0.99 \iff th = cdf^{-1}_{\mathcal{N}(3,1)}(0.99) \iff$$

$$th = \texttt{norminv(0.99, 3, 1)} \iff th = 5.3263$$

Thus, to ensure a FP of 1% we must shift the threshold from 5 to 5.3263

3. Using the new threshold, compute the true negative rate TN

$$TN = P(\text{`no alert'}|\text{`no intrusion'}) = P(\neg A|\neg I) = P(\mathbf{X} < th) =$$

$$P(\mathbf{X} \leq th) = cdf_{\mathcal{N}(3,1)}(5.3263) = \texttt{normcdf(5.3263, 3, 1)} = 0.99$$

Notice that $TN = 1 - FP$

IDS Basics

4. Like before, let the 'normal behavior' be $\mathbf{X} \sim \mathcal{N}(3,1)$. We construct a different IDS that classifies as 'intrusion' values that are higher than $th_{up} = 6$ and lower than $th_{lo} = 0.5$ (i.e. double threshold)

   Compute the false positive rate FP of this new IDS

$$FP = P(A|\neg I) = P(\{\mathbf{X} \leq th_{lo}\} \cup \{\mathbf{X} \geq th_{up}\}) =$$

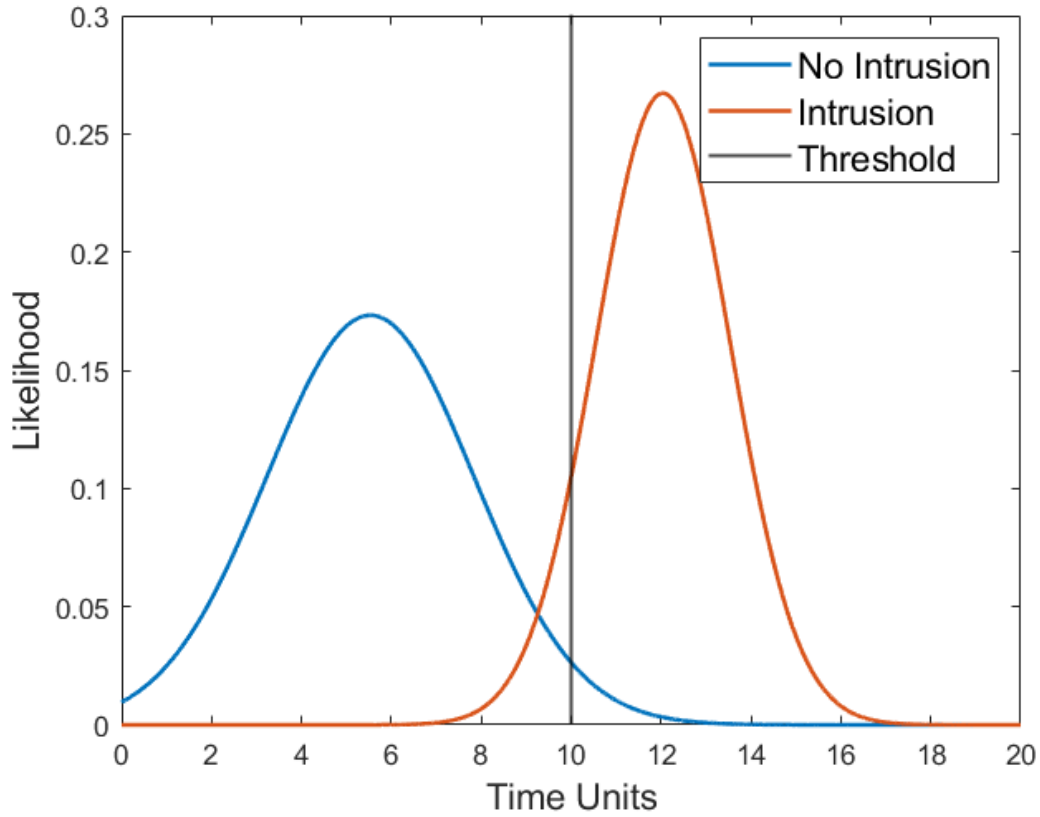
$$P(\mathbf{X} \leq th_{lo}) + P(\mathbf{X} \geq th_{up}) = cdf_{\mathcal{N}(3,1)}(th_{lo}) + (1 - cdf_{\mathcal{N}(3,1)}(th_{up})) =$$

$$\texttt{normcdf(0.5, 3, 1) + (1 - normcdf(6, 3, 1))} = 0.0491$$

IDS Basics

Example: Computing *TP* and *FP* without a testset

- This IDS has estimated models for both 'intrusion' and 'no intrusion'
- This produced r.v. $\mathbf{X} \sim \mathcal{N}(\mu = 5.5, \sigma = 2.3)$ and r.v. $\mathbf{Y} \sim \mathcal{N}(\mu = 12, \sigma = 1.5)$
- We have also set a threshold $th = 10$



IDS Basics

1. Compute the true positive rate TP

$$TP = P(A|I) = P(\mathbf{Y} \geq th) = 1 - cdf_{\mathcal{N}(12,1.5)}(th) =$$

$$\texttt{1 - normcdf(10, 12, 1.5)} = 0.9088$$

IDS Basics

1. Compute the true positive rate TP

$$TP = P(A|I) = P(\mathbf{Y} \geq th) = 1 - cdf_{\mathcal{N}(12,1.5)}(th) =$$

$$\texttt{1 - normcdf(10, 12, 1.5)} = 0.9088$$

2. Compute the false negative rate FN

$$FN = P(\neg A|I) = P(\mathbf{Y} \leq th) = cdf_{\mathcal{N}(12,1.5)}(th) = \texttt{normcdf(10, 12, 1.5)} = 0.0912$$

IDS Basics

1. Compute the true positive rate TP

$$TP = P(A|I) = P(\mathbf{Y} \geq th) = 1 - cdf_{\mathcal{N}(12,1.5)}(th) =$$

$$\texttt{1 - normcdf(10, 12, 1.5)} = 0.9088$$

2. Compute the false negative rate FN

$$FN = P(\neg A|I) = P(\mathbf{Y} \leq th) = cdf_{\mathcal{N}(12,1.5)}(th) = \texttt{normcdf(10, 12, 1.5)} = 0.0912$$

3. Compute the false positive rate FP

$$FP = P(A|\neg I) = P(\mathbf{X} \geq th) = 1 - cdf_{\mathcal{N}(5.5,2.3)}(th) =$$

$$\texttt{1 - normcdf(10, 5.5, 2.3)} = 0.0252$$

IDS Basics

1. Compute the true positive rate TP

$$TP = P(A|I) = P(\mathbf{Y} \geq th) = 1 - cdf_{\mathcal{N}(12,1.5)}(th) =$$

$$\texttt{1 - normcdf(10, 12, 1.5)} = 0.9088$$

2. Compute the false negative rate FN

$$FN = P(\neg A|I) = P(\mathbf{Y} \leq th) = cdf_{\mathcal{N}(12,1.5)}(th) = \texttt{normcdf(10, 12, 1.5)} = 0.0912$$

3. Compute the false positive rate FP

$$FP = P(A|\neg I) = P(\mathbf{X} \geq th) = 1 - cdf_{\mathcal{N}(5.5,2.3)}(th) =$$

$$\texttt{1 - normcdf(10, 5.5, 2.3)} = 0.0252$$

4. Compute the true negative rate TN

$$TN = P(\neg A|\neg I) = P(\mathbf{X} \leq th) = cdf_{\mathcal{N}(5.5,2.3)}(th) = \texttt{normcdf(10, 5.5, 2.3)} = 0.9748$$

IDS Basics

1. Compute the true positive rate TP

$$TP = P(A|I) = P(\mathbf{Y} \geq th) = 1 - cdf_{\mathcal{N}(12,1.5)}(th) =$$

$$\texttt{1 - normcdf(10, 12, 1.5)} = 0.9088$$

2. Compute the false negative rate FN

$$FN = P(\neg A|I) = P(\mathbf{Y} \leq th) = cdf_{\mathcal{N}(12,1.5)}(th) = \texttt{normcdf(10, 12, 1.5)} = 0.0912$$

3. Compute the false positive rate FP

$$FP = P(A|\neg I) = P(\mathbf{X} \geq th) = 1 - cdf_{\mathcal{N}(5.5,2.3)}(th) =$$

$$\texttt{1 - normcdf(10, 5.5, 2.3)} = 0.0252$$

4. Compute the true negative rate TN

$$TN = P(\neg A|\neg I) = P(\mathbf{X} \leq th) = cdf_{\mathcal{N}(5.5,2.3)}(th) = \texttt{normcdf(10, 5.5, 2.3)} = 0.9748$$

- Notice that $TP = 1 - FN$ and $TN = 1 - FP$
- Notice that sliding the threshold left and right will change the IDS making it more strict or more lenient towards certain type of errors

IDS Basics

Linking true positive to false negative. Show that the following holds:

$$TP = P(A|I) = 1 - P(\neg A|I) = 1 - FN$$

- From the law of total probability

$$P(I) = P((I \cap A) \cup (I \cap \neg A)) \iff P(I) = P(I \cap A) + P(I \cap \neg A)$$

- Dividing by $P(I)$ and applying Bayes rule

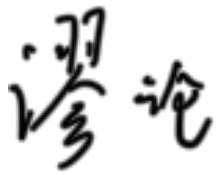
$$1 = \frac{P(I \cap A)}{P(I)} + \frac{P(I \cap \neg A)}{P(I)} \iff 1 = P(A|I) + P(\neg A|I) \iff$$

$$P(A|I) = 1 - P(\neg A|I)$$

- Try to prove in a similar way that $FP = 1 - TN$

The Base Rate Fallacy

Base Rate Fallacy



Base rate. When analyzing and IDS we are also interested in the base rate parameter $P(I) = B$ i.e. the probability of an intrusion

- It is often referred to as the 'environment' parameter
- It describes whether intrusions/anomalies occur often (e.g. spam emails) or rarely (e.g. airplane failure)
- It may even change over time

Base Rate Fallacy

Bayesian detection rate. An important metric is the Bayesian detection rate $P(I|A)$ a.k.a. positive predictive value PPV

$$PPV = P(I|A) = \frac{P(I,A)}{P(A)} = \frac{P(A,I)}{P(A)} = \frac{P(A|I)P(I)}{P(A)} =$$

$$\frac{P(A|I)P(I)}{\sum_{\mathbf{X}\in\{I,\neg I\}} P(A,\mathbf{X})} = \frac{P(A|I)P(I)}{\sum_{\mathbf{X}\in\{I,\neg I\}} P(A|\mathbf{X})P(\mathbf{X})} = \frac{P(A|I)P(I)}{P(A|I)P(I) + P(A|\neg I)P(\neg I)} \iff$$

$$PPV = \frac{TP * B}{TP * B + FP * (1 - B)}$$

Base Rate Fallacy

Bayesian detection rate. An important metric is the Bayesian detection rate $P(I|A)$ a.k.a. positive predictive value PPV

$$PPV = P(I|A) = \frac{P(I,A)}{P(A)} = \frac{P(A,I)}{P(A)} = \frac{P(A|I)P(I)}{P(A)} =$$

$$\frac{P(A|I)P(I)}{\sum_{\mathbf{X}\in\{I,\neg I\}} P(A,\mathbf{X})} = \frac{P(A|I)P(I)}{\sum_{\mathbf{X}\in\{I,\neg I\}} P(A|\mathbf{X})P(\mathbf{X})} = \frac{P(A|I)P(I)}{P(A|I)P(I)+P(A|\neg I)P(\neg I)} \iff$$

$$PPV = \frac{TP * B}{TP * B + FP * (1-B)}$$

- We have linked PPV to TP, B, FP using the Bayes rule and the law of total probability
- PPV close to 1 means that we need to take action after we see an alert, since an intrusion is likely
- PPV close to 0 means that we can ignore the alert, since an intrusion is unlikely
- Try to construct a similar formula for the negative predictive value $NPV = P(\neg I|\neg A)$

Base Rate Fallacy

Example: Base Rate Fallacy

We have trained a (fairly good) IDS that is capable of detecting intrusions by inspecting the TCP/IP packets in our internal network.
We test the IDS using a dataset that contains labelled regular and intrusive packets and obtain the following metrics:

- the true positive rate is quite high i.e. $TP = P(A|I) = 99\%$
- the false positive rate is fairly low i.e. $FP = P(A|\neg I) = 2\%$
- intrusions are not very common i.e. the base rate $B = P(I) = 10^{-3}\%$

Base Rate Fallacy

Example: Base Rate Fallacy

We have trained a (fairly good) IDS that is capable of detecting intrusions by inspecting the TCP/IP packets in our internal network.
We test the IDS using a dataset that contains labelled regular and intrusive packets and obtain the following metrics:

- the true positive rate is quite high i.e. $TP = P(A|I) = 99\%$
- the false positive rate is fairly low i.e. $FP = P(A|\neg I) = 2\%$
- intrusions are not very common i.e. the base rate $B = P(I) = 10^{-3}\%$

1. Compute the false negative rate FN

$$FN = P(\text{`no alert'}|\text{`intrusion'}) = P(\neg A|I) = 1 - P(A|I) =$$

$$1 - P(\text{`alert'}|\text{`intrusion'}) = 1 - TP = 1 - 0.99 = 0.01$$

Base Rate Fallacy

Example: Base Rate Fallacy

We have trained a (fairly good) IDS that is capable of detecting intrusions by inspecting the TCP/IP packets in our internal network.
We test the IDS using a dataset that contains labelled regular and intrusive packets and obtain the following metrics:

- the true positive rate is quite high i.e. $TP = P(A|I) = 99\%$
- the false positive rate is fairly low i.e. $FP = P(A|\neg I) = 2\%$
- intrusions are not very common i.e. the base rate $B = P(I) = 10^{-3}\%$

1. Compute the false negative rate FN

$$FN = P(\text{`no alert'}|\text{`intrusion'}) = P(\neg A|I) = 1 - P(A|I) =$$
$$1 - P(\text{`alert'}|\text{`intrusion'}) = 1 - TP = 1 - 0.99 = 0.01$$

2. Compute the true negative rate TN

$$TN = P(\text{`no alert'}|\text{`no intrusion'}) = P(\neg A|\neg I) = 1 - P(A|\neg I) =$$
$$1 - P(\text{`alert'}|\text{`no intrusion'}) = 1 - FP = 1 - 0.02 = 0.98$$

Base Rate Fallacy

3. Compute the Bayesian detection rate PPV (a.k.a. positive predictive value)

$$PPV = P(\text{`intrusion'}|\text{`alert'}) = P(I|A) = \frac{P(I)P(A|I)}{P(I)P(A|I) + P(\neg I)P(A|\neg I)} =$$

$$\frac{B * TP}{B * TP + (1 - B) * FP} = \frac{10^{-5} * 0.99}{10^{-5} * 0.99 + 0.99999 * 0.02} \approx 0.0004947600$$

Base Rate Fallacy

3. Compute the Bayesian detection rate PPV (a.k.a. positive predictive value)

$$PPV = P(\text{`intrusion'}|\text{`alert'}) = P(I|A) = \frac{P(I)P(A|I)}{P(I)P(A|I) + P(\neg I)P(A|\neg I)} =$$

$$\frac{B * TP}{B * TP + (1 - B) * FP} = \frac{10^{-5} * 0.99}{10^{-5} * 0.99 + 0.99999 * 0.02} \approx 0.0004947600$$

- Thus, whenever we see an alert, the probability of an actual intrusion happening is extremely low! Can a human cope with it?
- This happens in spite of the good IDS performance i.e. high TP and low FP

Base Rate Fallacy

3. Compute the Bayesian detection rate PPV (a.k.a. positive predictive value)

$$PPV = P(\text{`intrusion'}|\text{`alert'}) = P(I|A) = \frac{P(I)P(A|I)}{P(I)P(A|I) + P(\neg I)P(A|\neg I)} =$$

$$\frac{B * TP}{B * TP + (1 - B) * FP} = \frac{10^{-5} * 0.99}{10^{-5} * 0.99 + 0.99999 * 0.02} \approx 0.0004947600$$

- Thus, whenever we see an alert, the probability of an actual intrusion happening is extremely low! Can a human cope with it?
- This happens in spite of the good IDS performance i.e. high TP and low FP
- The culprit is the very low base rate B i.e. only a very small portion of the packets relate to an intrusion
- This counter-intuitive effect is called the **base-rate fallacy**

when detect an alert

very low prob that this is an intrusion

Base Rate Fallacy

3. Compute the Bayesian detection rate PPV (a.k.a. positive predictive value)

$$PPV = P(\text{`intrusion'}|\text{`alert'}) = P(I|A) = \frac{P(I)P(A|I)}{P(I)P(A|I) + P(\neg I)P(A|\neg I)} =$$

$$\frac{B * TP}{B * TP + (1 - B) * FP} = \frac{10^{-5} * 0.99}{10^{-5} * 0.99 + 0.99999 * 0.02} \approx 0.0004947600$$

- Thus, whenever we see an alert, the probability of an actual intrusion happening is extremely low! Can a human cope with it?
- This happens in spite of the good IDS performance i.e. high TP and low FP
- The culprit is the very low base rate B i.e. only a very small portion of the packets relate to an intrusion
- This counter-intuitive effect is called the **base-rate fallacy**
- Networks and computers get faster increasing the amount of packets/data that can be captured, thus the base rate can decrease even more
- We end up looking for a needle in a haystack!

Base Rate Fallacy

4. Compute the negative predictive value NPV

$$NPV = P(\text{`no intrusion'}|\text{`no alert'}) = P(\neg I|\neg A) = \frac{P(\neg I)P(\neg A|\neg I)}{P(\neg I)P(\neg A|\neg I) + P(I)P(\neg A|I)} =$$

$$\frac{(1-B)*TN}{(1-B)*TN + B*FN} = \frac{0.99999*0.98}{0.99999*0.98 + 10^{-5}*0.01} \approx 0.9999998979$$

- Whenever we don't see an alert, there is a very high chance of no actual intrusion
- The probability of no intrusion $P(\neg I)$ was already high
- The absence of an alert makes it marginally higher i.e. $P(\neg I|\neg A) > P(\neg I)$

Base Rate Fallacy

5. Assume a different IDS with base rate $B = 10^{-6}$
   For the IDS to be usable we require a Bayesian detection rate $PPV = 0.9$
   Find TP and FP pairs such that the PPV requirement is fulfilled

$$PPV = \frac{P(I)P(A|I)}{P(I)P(A|I) + P(\neg I)P(A|\neg I)} \iff PPV = \frac{B * TP}{B * TP + (1 - B) * FP} \iff$$

$$0.9 = \frac{10^{-6} * TP}{10^{-6} * TP + 0.999999 * FP}$$

Base Rate Fallacy

5. Assume a different IDS with base rate $B = 10^{-6}$
   For the IDS to be usable we require a Bayesian detection rate $PPV = 0.9$
   Find TP and FP pairs such that the PPV requirement is fulfilled

$$PPV = \frac{P(I)P(A|I)}{P(I)P(A|I) + P(\neg I)P(A|\neg I)} \iff PPV = \frac{B * TP}{B * TP + (1 - B) * FP} \iff$$

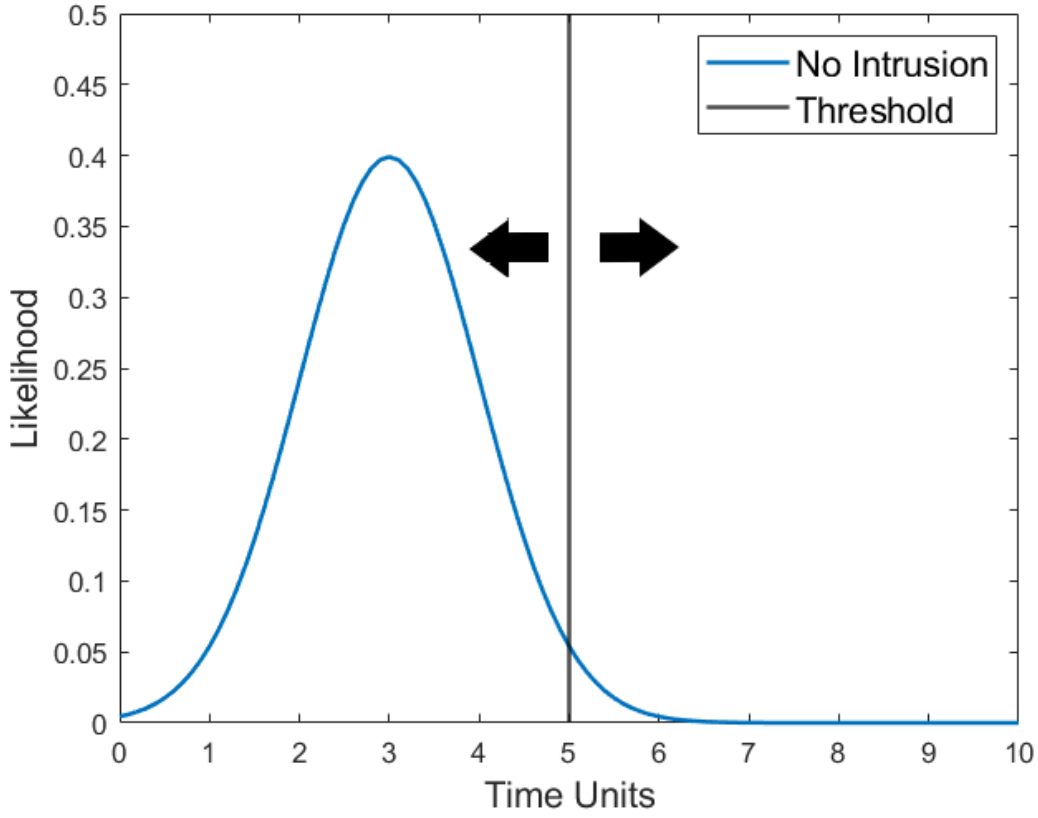
$$0.9 = \frac{10^{-6} * TP}{10^{-6} * TP + 0.999999 * FP}$$

- We can choose TP and solve for FP or vice-versa
- Choosing $TP = 1$ results in $FP \approx 0.1111112222 * 10^{-6}$
- Thus even a perfect (w.r.t. TP) system, requires an extremely low FP to reach the reasonable $PPV = 0.9$
- Any decent IDS must be very good at suppressing false alerts

Receiver Operator Characteristic (ROC) Curve

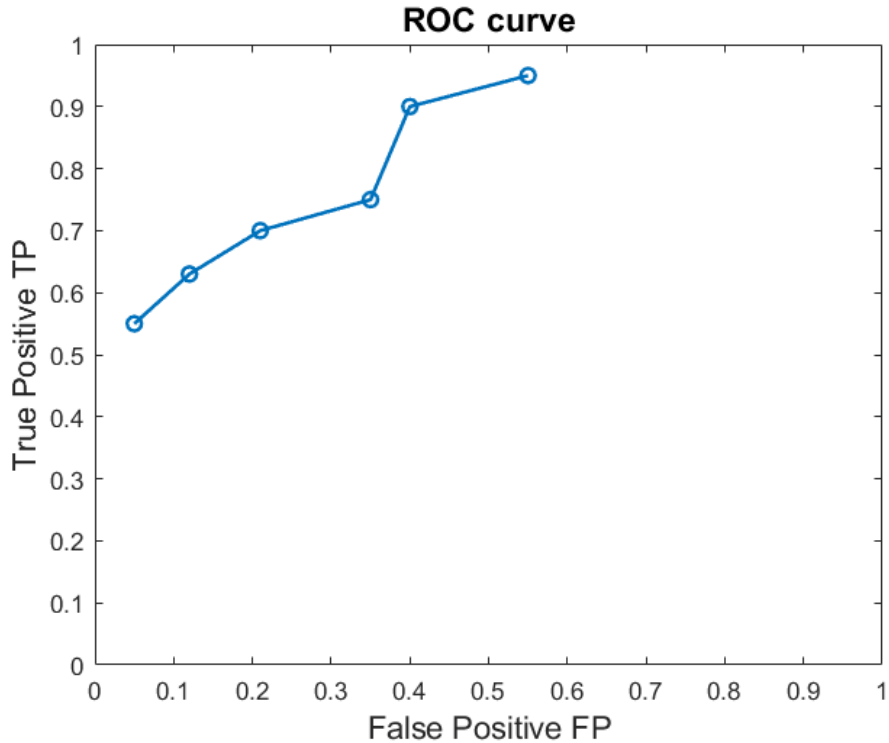
ROC

- Shifting the threshold left/right will result in a different IDS configuration
- Each threshold choice produces a new (TP, FP) pair
  i.e. a new IDS instance



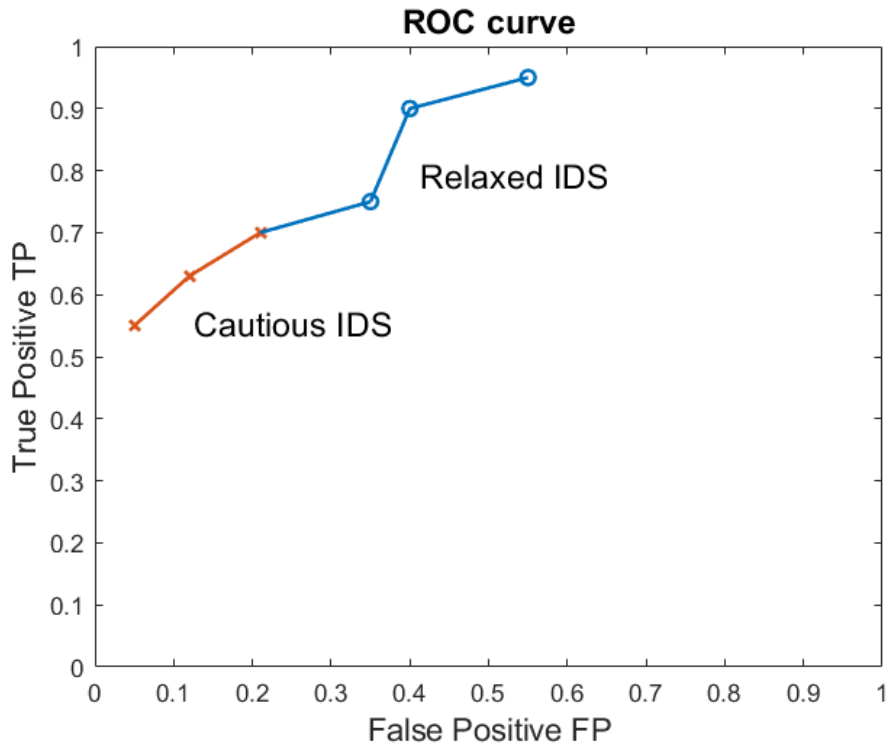
ROC

- We collect multiple IDS instances to create the ROC
- **ROC curve:** A 2-dimensional graph showing the tradeoff between TP (y-axis) and FP (x-axis)
- Both axes range in $[0, 1]$



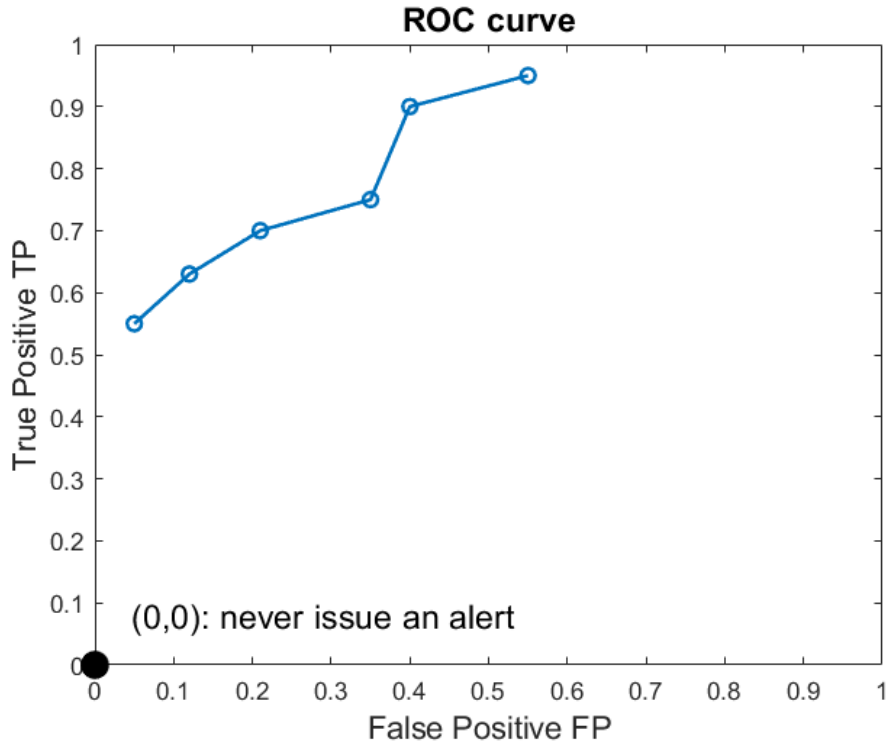
ROC

- ROC points with low *TP* and low *FP* are considered 'cautious': they will raise an alert if there is strong evidence
- ROC points with high *TP* and high *FP* are considered 'relaxed': they will raise an alert if there is weak evidence



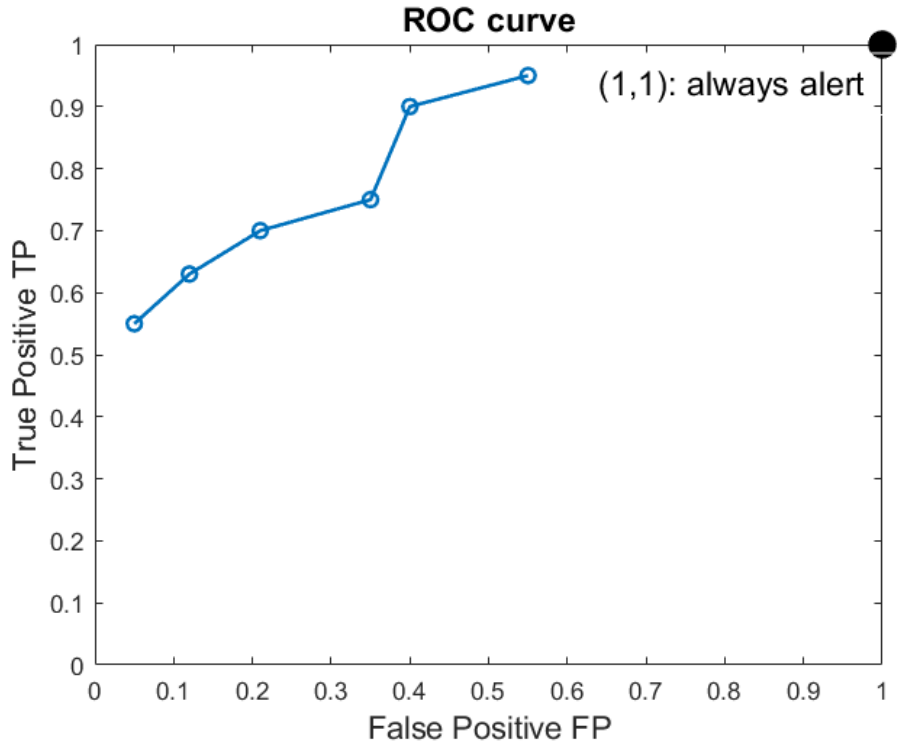
ROC

- The ROC point $(0, 0)$ represents an IDS that never raises an alert
- Thus the IDS does not err but it is not useful



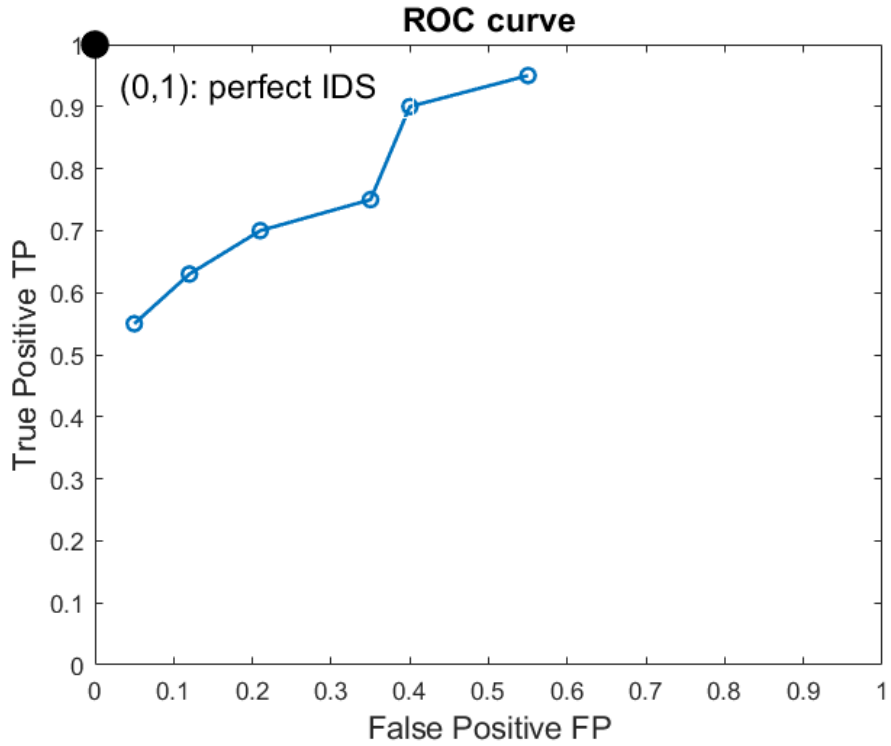
ROC

- The ROC point $(1, 1)$ represents an IDS that always raises an alert
- Thus it can detect all intrusions but it not useful



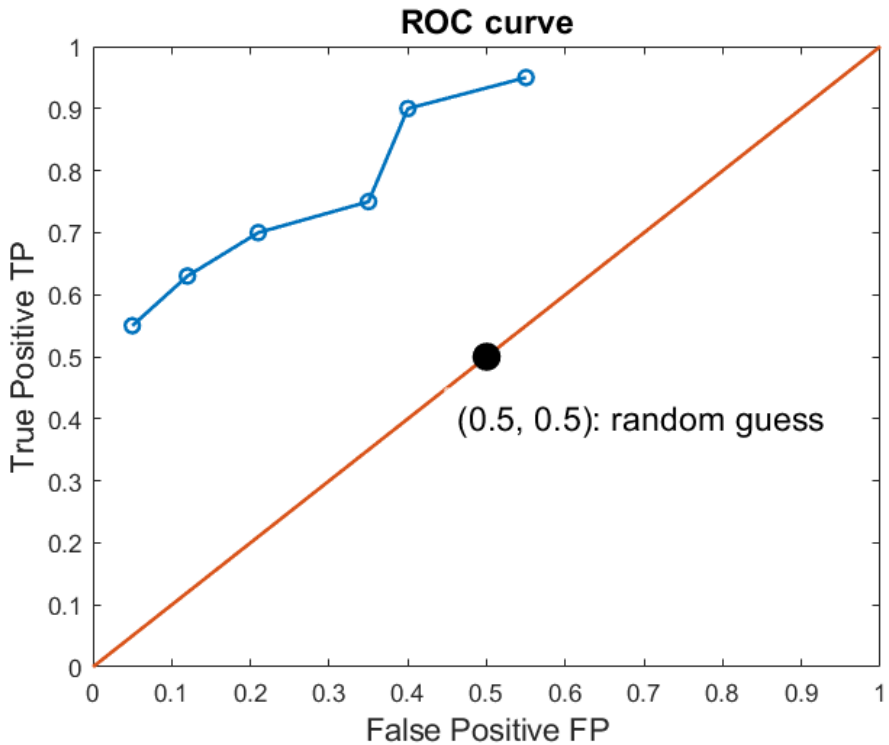
ROC

- The ROC point $(0, 1)$ represents the ideal IDS
- $TP = 1$ thus all intrusions are detected and $FP = 0$ thus there are no mistakes



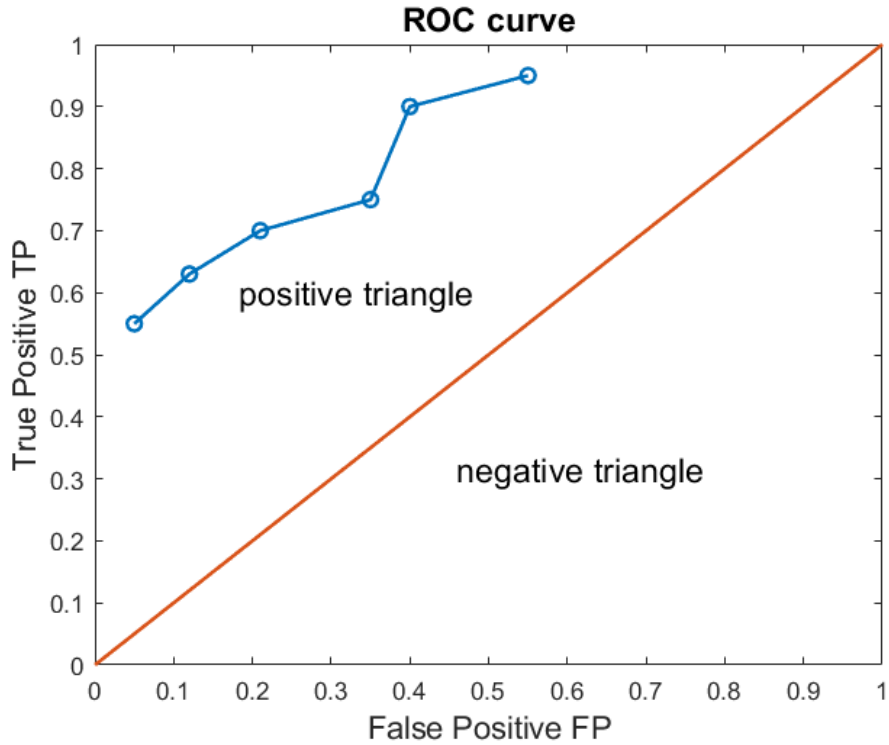
ROC

- The ROC point $(0.5, 0.5)$ represents an IDS that is guessing randomly
- Moving away from the diagonal means that the IDS is learning



ROC

- The upper triangle means that the IDS performs better than a random guess
- The lower triangle means that the IDS performs worse than random guess
- If we have an IDS in the lower (negative) triangle we can negate its classifier decision
  - i.e. map 'alert' $\rightarrow$ 'no alert' and 'no alert' $\rightarrow$ 'alert'



ROC

Generating the ROC: naive method

Input: testset $\mathcal{T}$, IDS test function $test\_IDS(\cdot)$
Output: Collection of ROC points (TP, FP)

```
// Test the dataset using multiple thresholds
```
$i = 1$
for $th = -\infty$ *until* $+\infty$ **do**
    `// Compute the ratio pair`
    $(TP(i), FP(i)) = \text{test\_IDS}(\mathcal{T}, th)$
    $i = i + 1$
end

ROC

Generating the ROC: naive method

Input: testset $\mathcal{T}$, IDS test function $test\_IDS(\cdot)$
Output: Collection of ROC points (TP, FP)

```
// Test the dataset using multiple thresholds
i = 1
for th= −∞ until +∞ do
    // Compute the ratio pair
    (TP(i), FP(i)) = test_IDS(T, th)
    i = i + 1
end
```

- Test a given dataset $\mathcal{T}$ multiple times using a different threshold th
- The testset $\mathcal{T}$ contains the datapoints and the respective labels (ground truth)
- The function test_IDS$(\cdot)$ depends on the IDS and it typically computes some score (see the examples in the IDS Basics section)
- For every chosen threshold th, the function $test\_IDS(\mathcal{T}, th)$ will return the respective (TP, FP) rates
- This method is slow to compute!

ROC

Generating the ROC: efficient method

Input: testset $\mathcal{T}$, score function $scoref(\cdot)$, sorting function $sort(\cdot)$, q intrusions in $\mathcal{T}$ and w normal datapoints in $\mathcal{T}$

Output: Collection of ROC points (TP, FP)

`// Compute the scores for all elements in testset` $\mathcal{T}$

for *all* t *in* $\mathcal{T}$ **do**

  $score(i) = scoref(t)$

end

`// Sort the elements of the testset in descending order of scores`

$(\mathcal{T}_{sorted}, score_{sorted}) = sort(\mathcal{T}, score)$

$score_{prev} = -\infty$

for $i = 1$ *until* $size(\mathcal{T}_{sorted})$ **do**

  if $score_{sorted}(i) \neq score_{prev}$ **then**

    $TP(j) = TPcount/q$; $FP(j) = FPcount/w$

    $j = j + 1$; $score_{prev} = score_{sorted}(i)$

  end

  if $\mathcal{T}_{sorted}(i)$ *is 'intrusion'* **then**

    $TPcount = TPcount + 1$

  else

    $FPcount = FPcount + 1$

  end

end

$TP(j) = TPcount/q$; $FP(j) = FPcount/w$

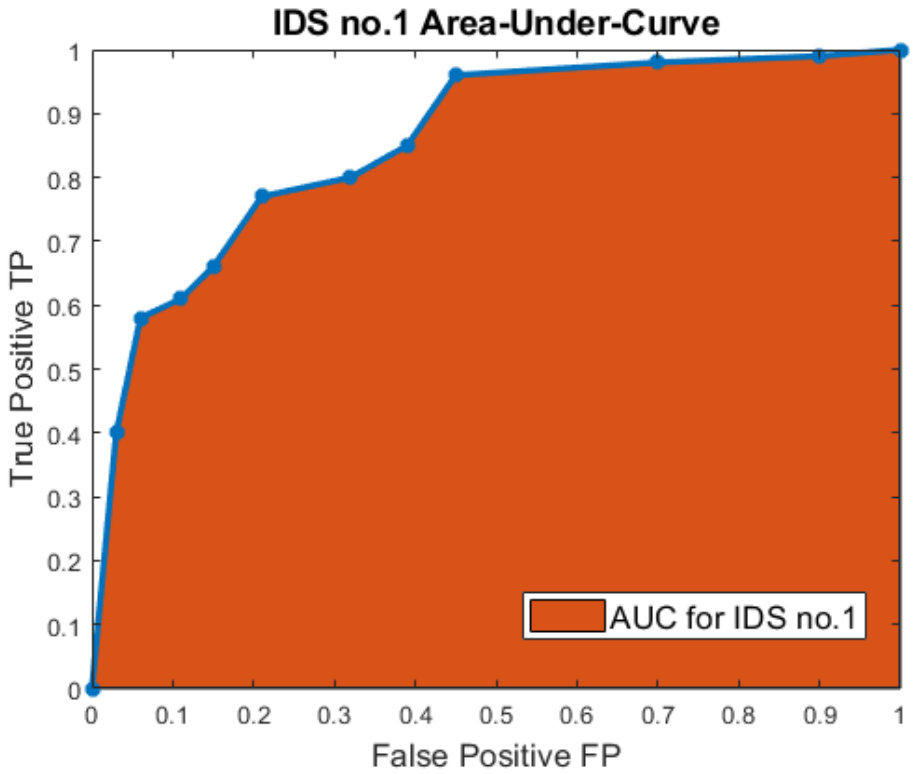
ROC

- This method sorts the scores and iterates over them to compute the TP and FP
- This method does not compute ROC points when the scores are equal to avoid over/under-estimating the IDS
- Fast to compute: $O(nlogn)$ for sorting and $O(n)$ for the iteration
- Verify these steps using MATLAB code: `generate_roc_naive_method.m` and `generate_roc_fast_method.m`

ROC

Area Under Curve (AUC)

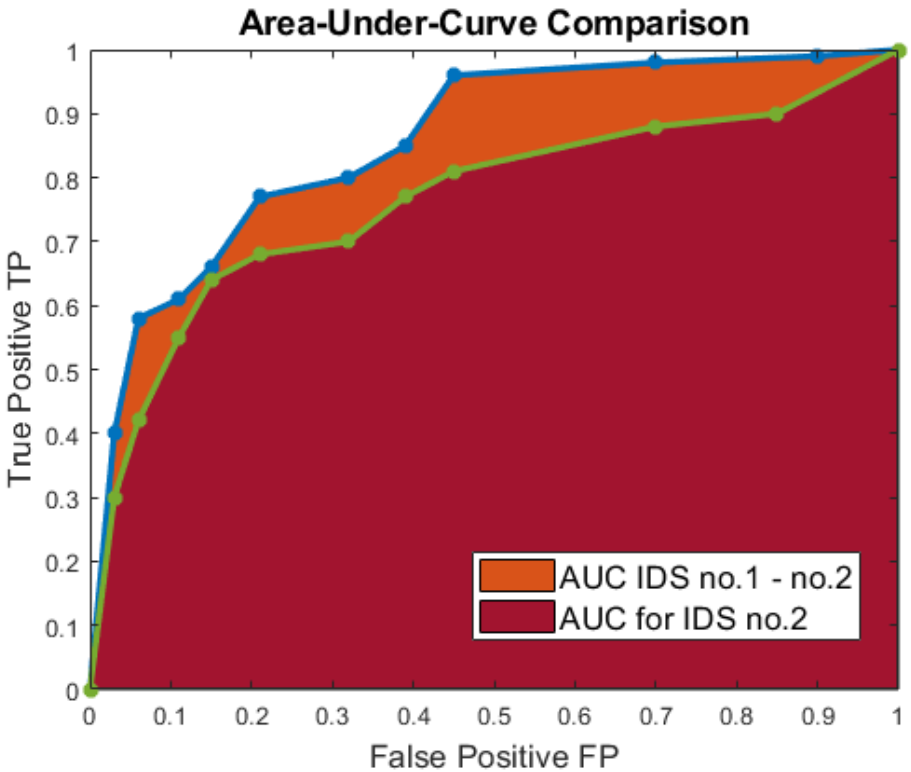
- AUC is a metric of IDS performance
- Since the AUC covers part of a square rectangle, $0 \leq \text{AUC} \leq 1$



ROC

Area Under Curve (AUC)

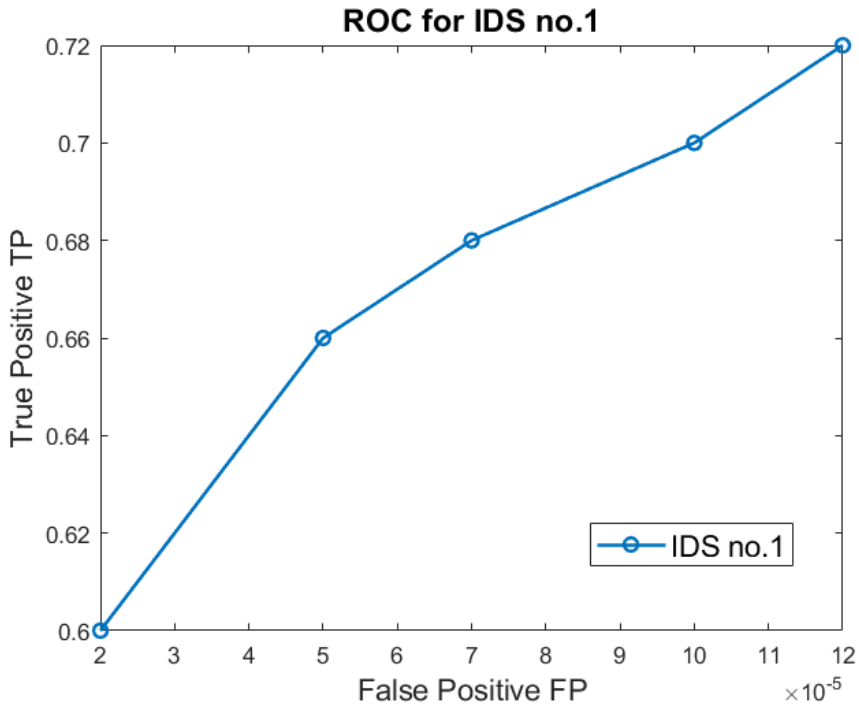
- AUC can be used to compare the performance of two IDSs



Evaluation of Intrusion Detection Systems

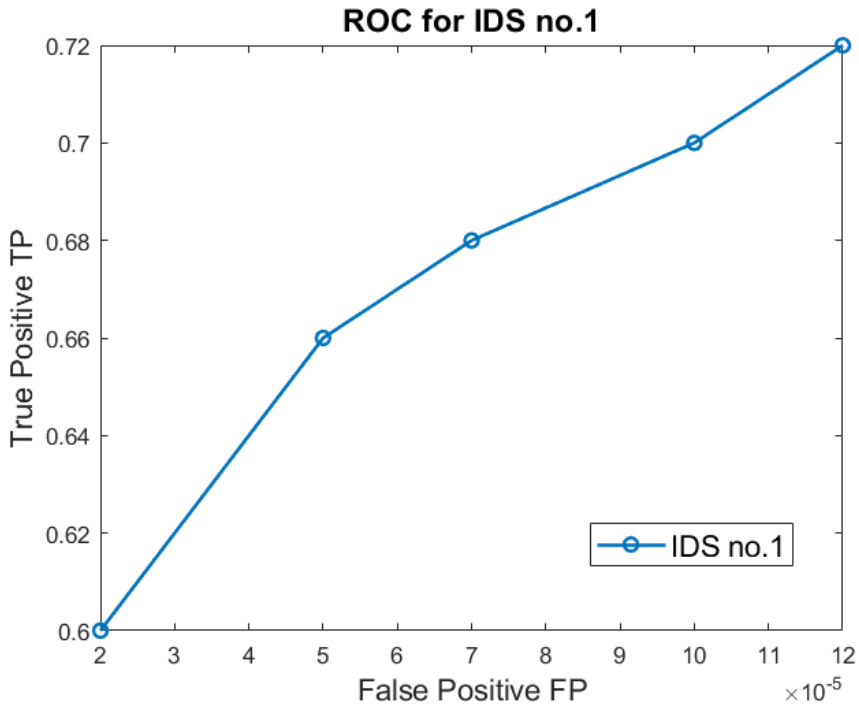
IDS Evaluation

- Which operating point (TP, FP) in the following ROC curve is optimal for the performance of the IDS?



IDS Evaluation

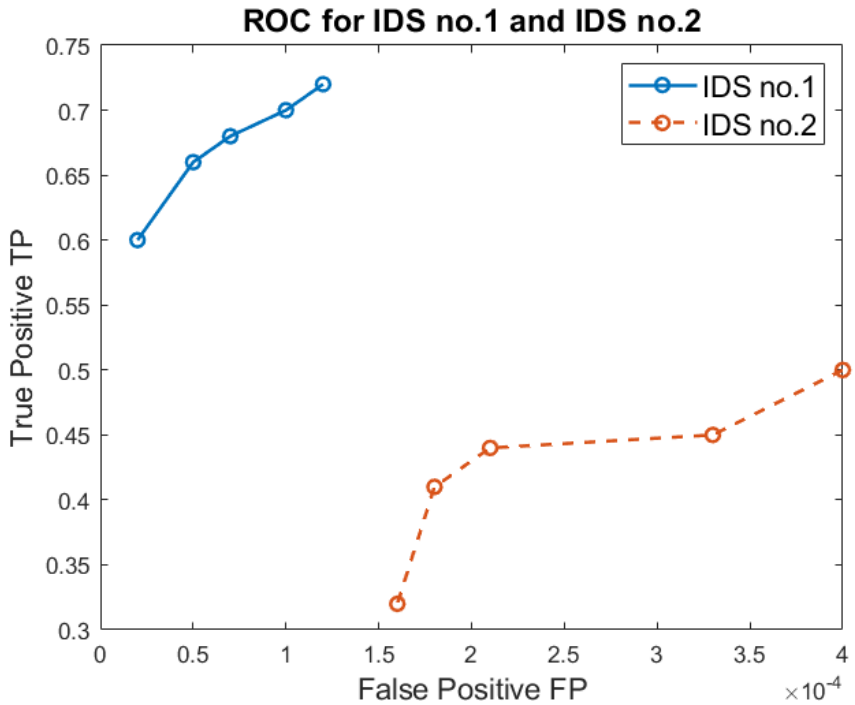
- Which operating point (TP, FP) in the following ROC curve is optimal for the performance of the IDS?



- We need to balance the tradeoff between TP and FP
- Can we combine them in a metric?

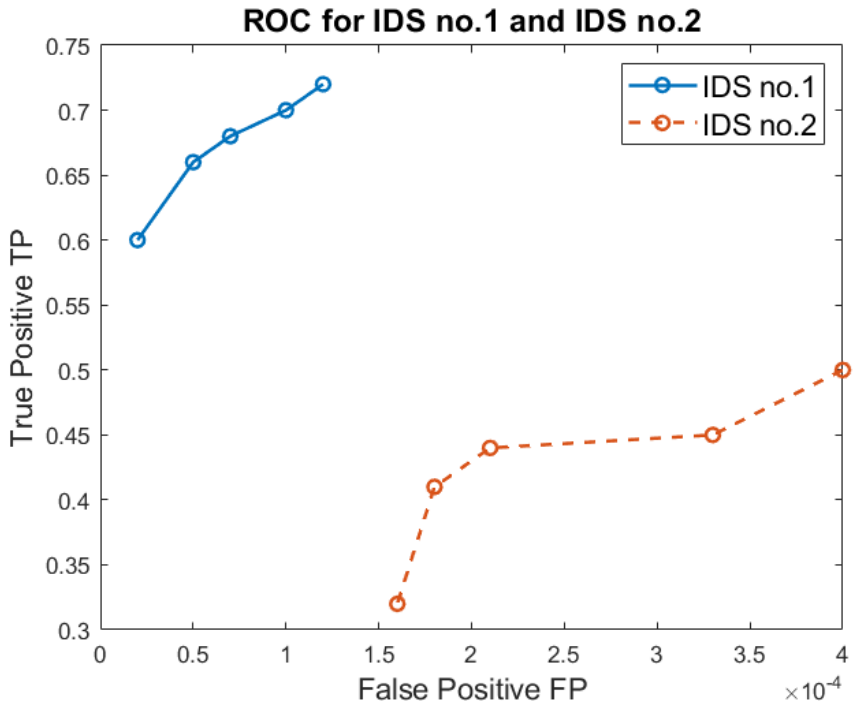
IDS Evaluation

- Can we use the ROC curves to compare the performance of 2 different IDSs?



IDS Evaluation

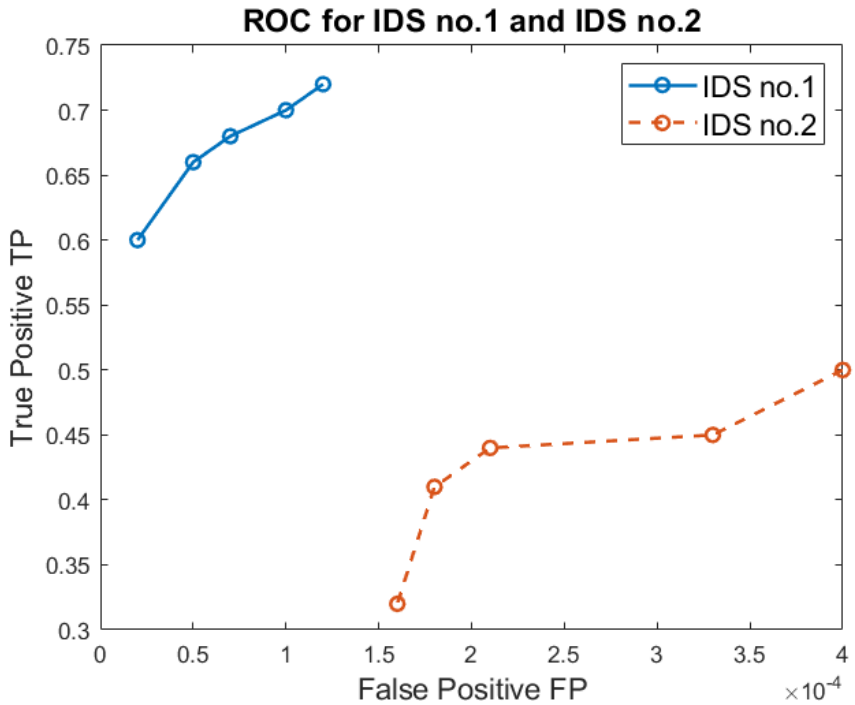
- Can we use the ROC curves to compare the performance of 2 different IDSs?



- ROC no.1 is above ROC no.2 thus $TP_1 > TP_2$ always
- ROC no.1 is to the left of ROC no.2 thus $FP_1 < FP_2$ always

IDS Evaluation

- Can we use the ROC curves to compare the performance of 2 different IDSs?



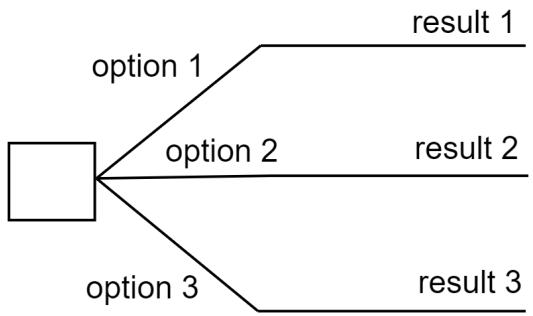
- ROC no.1 is above ROC no.2 thus $TP_1 > TP_2$ always
- ROC no.1 is to the left of ROC no.2 thus $FP_1 < FP_2$ always
- What if the ROCs 'cross'? Now the comparison is difficult!
- Measuring the AUC gives us the average performance yet integrates all (good and bad) operating points (TP, FP) and can be misleading

IDS Evaluation

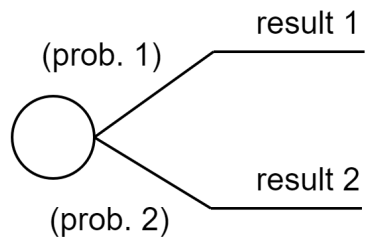
- To find the best operating point and to compare various IDSs we will use **decision trees** in conjunction with a **cost function**
- Decision trees are used to analyze a complex decision process that contains multiple factors and uncertainties

IDS Evaluation

- To find the best operating point and to compare various IDSs we will use **decision trees** in conjunction with a **cost function**
- Decision trees are used to analyze a complex decision process that contains multiple factors and uncertainties
- The tree contains two types of nodes: **decision/action nodes** (squares) and **event/chance nodes** (circles)
- At a decision node (square) you can select any one branch
- At an event node (circle) the branch is chosen in a probabilistic way



Decision Node



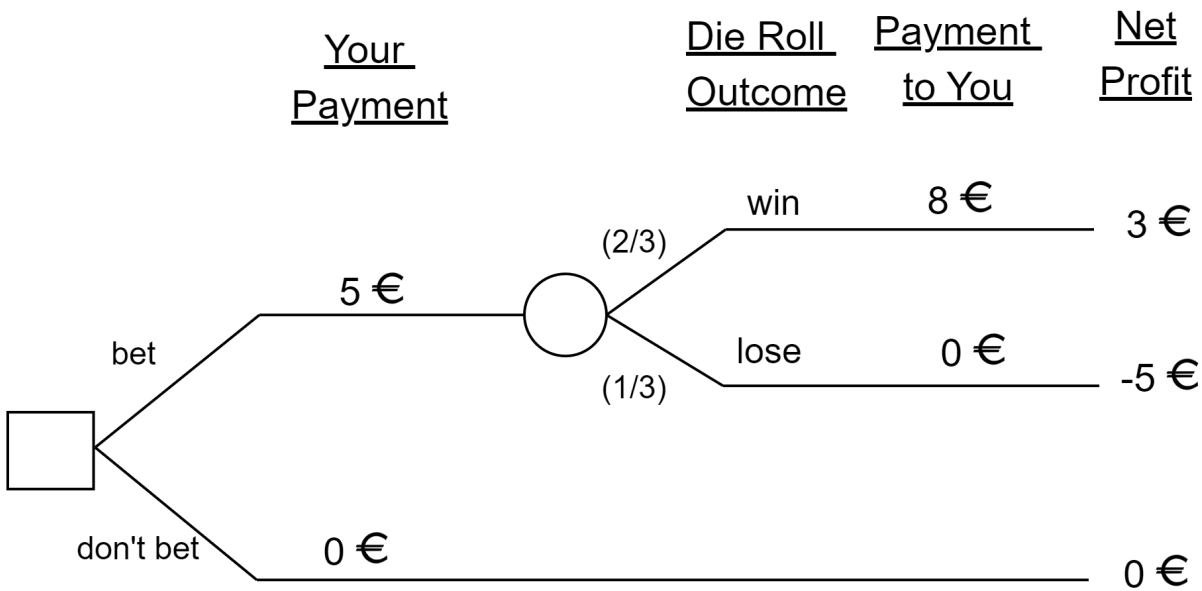
Event Node

IDS Evaluation

Example: Simple decision tree

A casino game works is played with dice. To roll a die you need to bet 5 €. If you roll 3, 4, 5 or 6 you win 8 euros. If you roll 1 or 2, you lose your bet. Should you bet on this game or not?

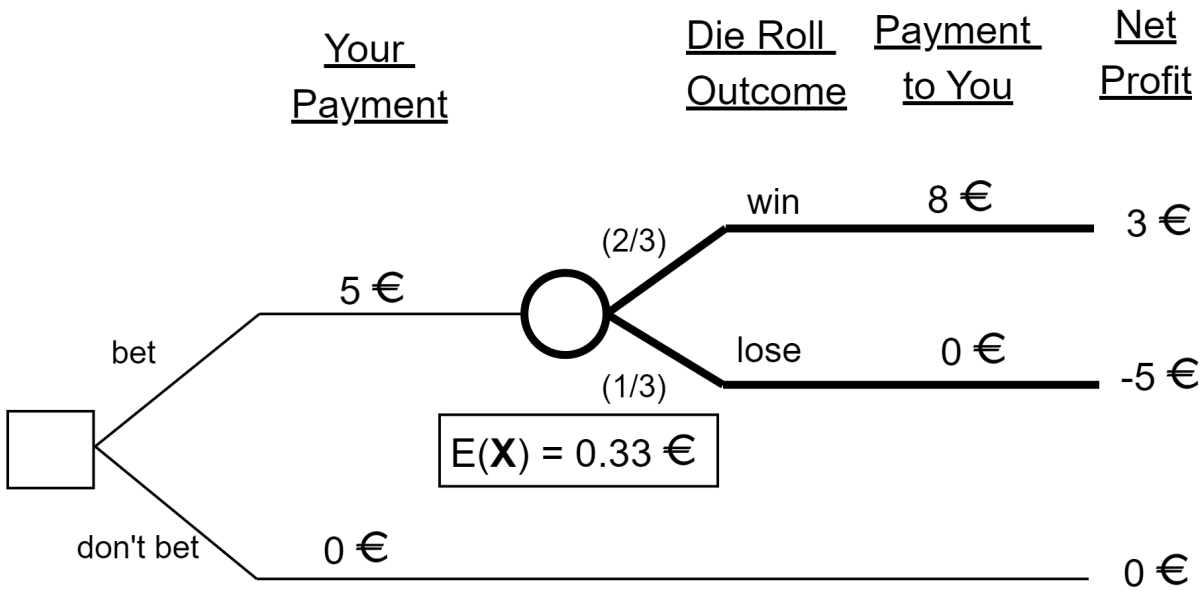
1. Describe your choices and their results using a decision tree



- You read the tree from left (root node) to right (endpoints)

IDS Evaluation

2. Perform a decision tree rollback (right to left)



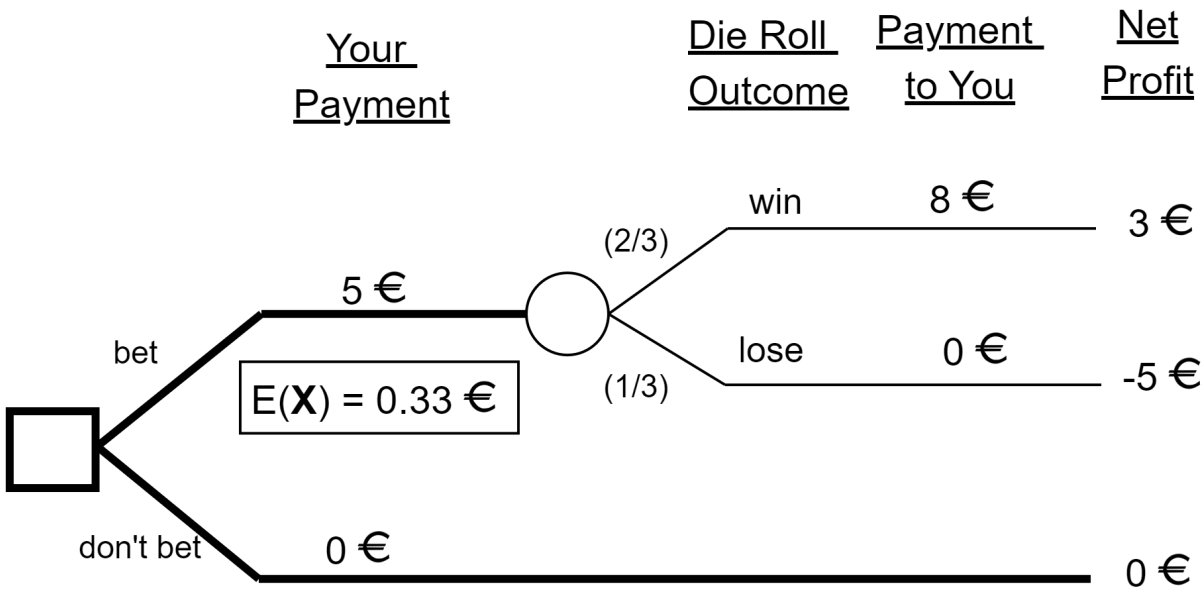
- We first encounter an event node. Let random variable $\mathbf{X}$ to describe the net profit after the die roll i.e. $\mathbf{X}$ takes values in $\{3, -5\}$
- We compute the expected net profit $E(\mathbf{X})$ for the event node

$$\text{Expected Net Profit} = E(\mathbf{X}) = \sum_{\mathbf{x} \in \{3,-5\}} \mathbf{x} P(\mathbf{x}) = 3 * P(3) + (-5) * P(-5) =$$

$$3 * P(\text{'win'}) - 5 * P(\text{'lose'}) = 3 * 2/3 - 5 * 1/3 = 0.33€$$

IDS Evaluation

2. Perform a decision tree rollback (right to left)



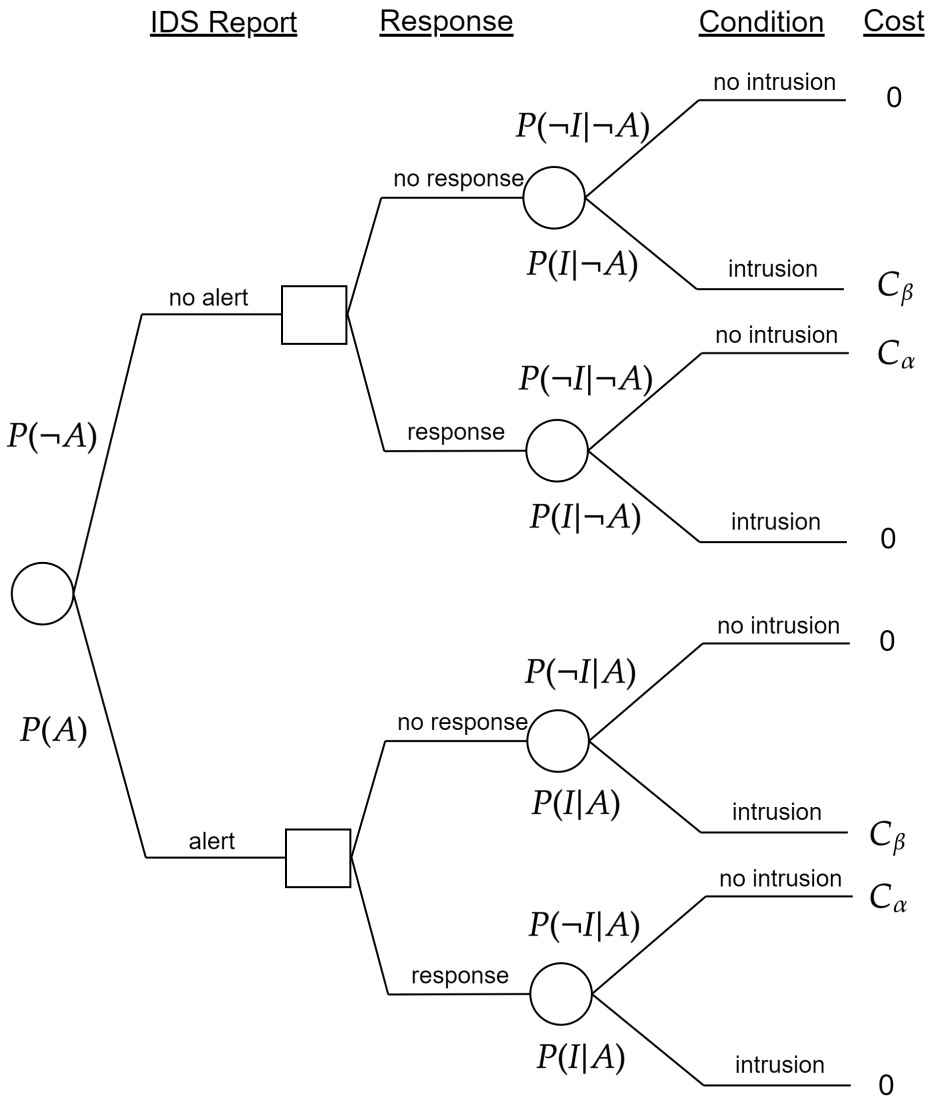
- We then encounter a decision node. At a decision node we choose the branch that maximizes the net profit

$$\text{Net Profit} = max\{E(\mathbf{X}), 0\} = 0.33€$$

- The net profit at the root node is positive thus, using the tree, we conclude that placing a bet is a good idea

IDS Evaluation

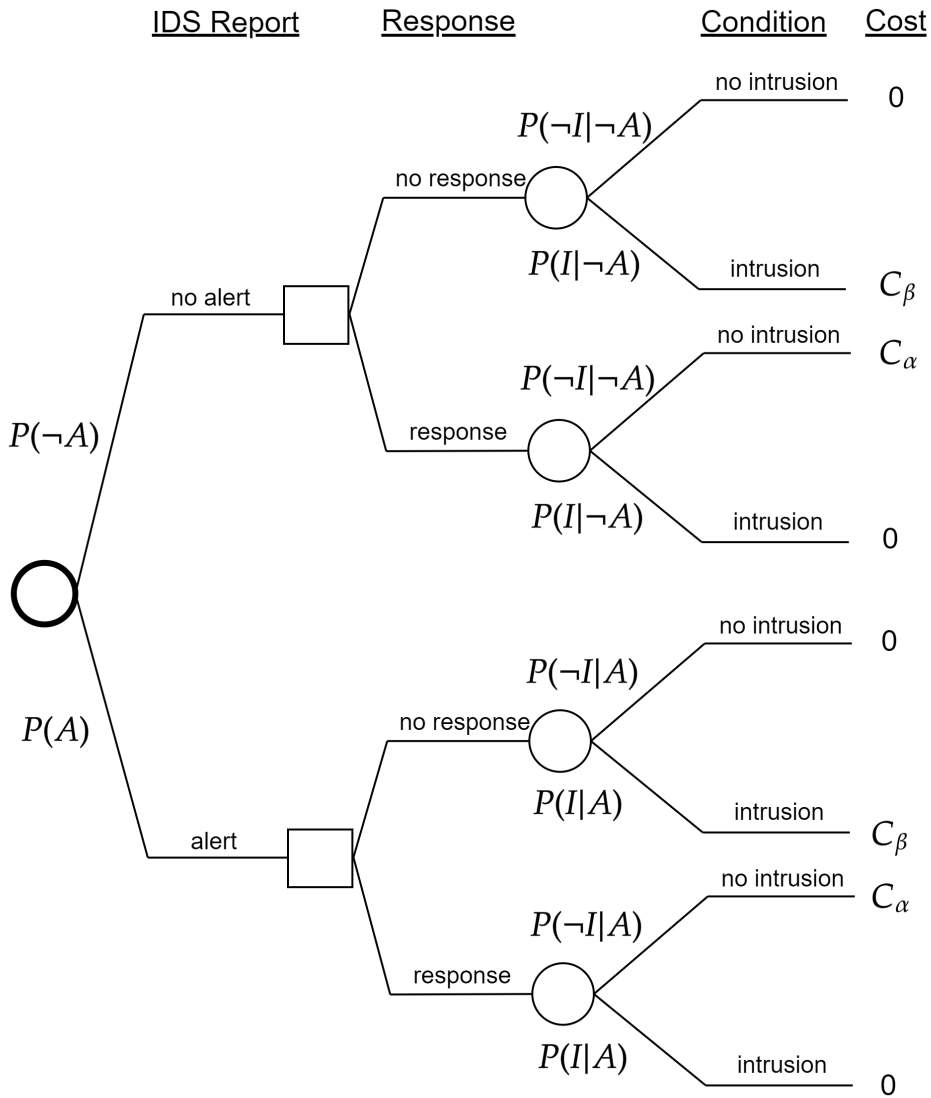
Example: IDS decision tree



- We will apply the decision tree in the IDS scenario

IDS Evaluation

IDS decision tree:



- **IDS report (event node)**

$$P(A) = \sum_{\mathbf{y} \in \{I, \neg I\}} P(A, \mathbf{y}) =$$

$$P(A, I) + P(A, \neg I) =$$

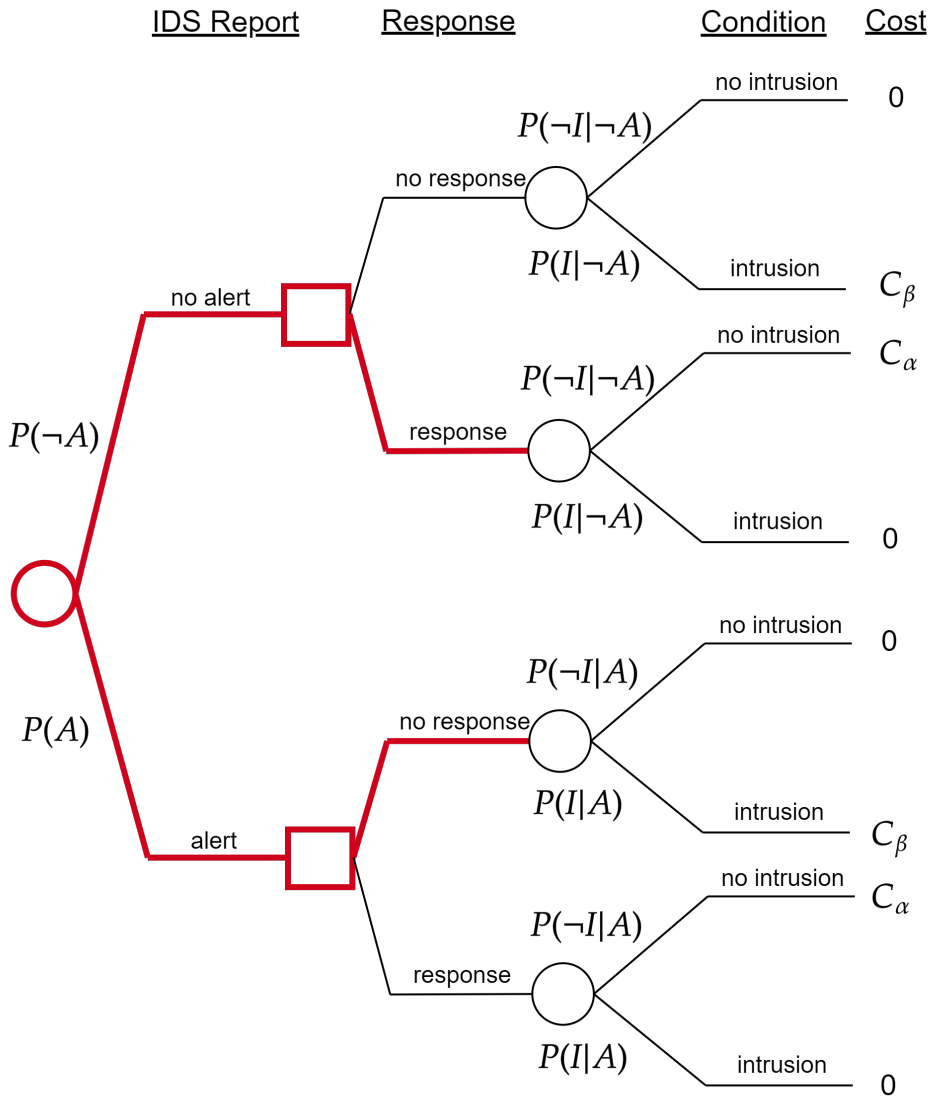
$$P(A|I)P(I) + P(A|\neg I)P(\neg I) =$$

$$TP * B + FP * (1 - B)$$

$$P(\neg A) = 1 - P(A)$$

IDS Evaluation

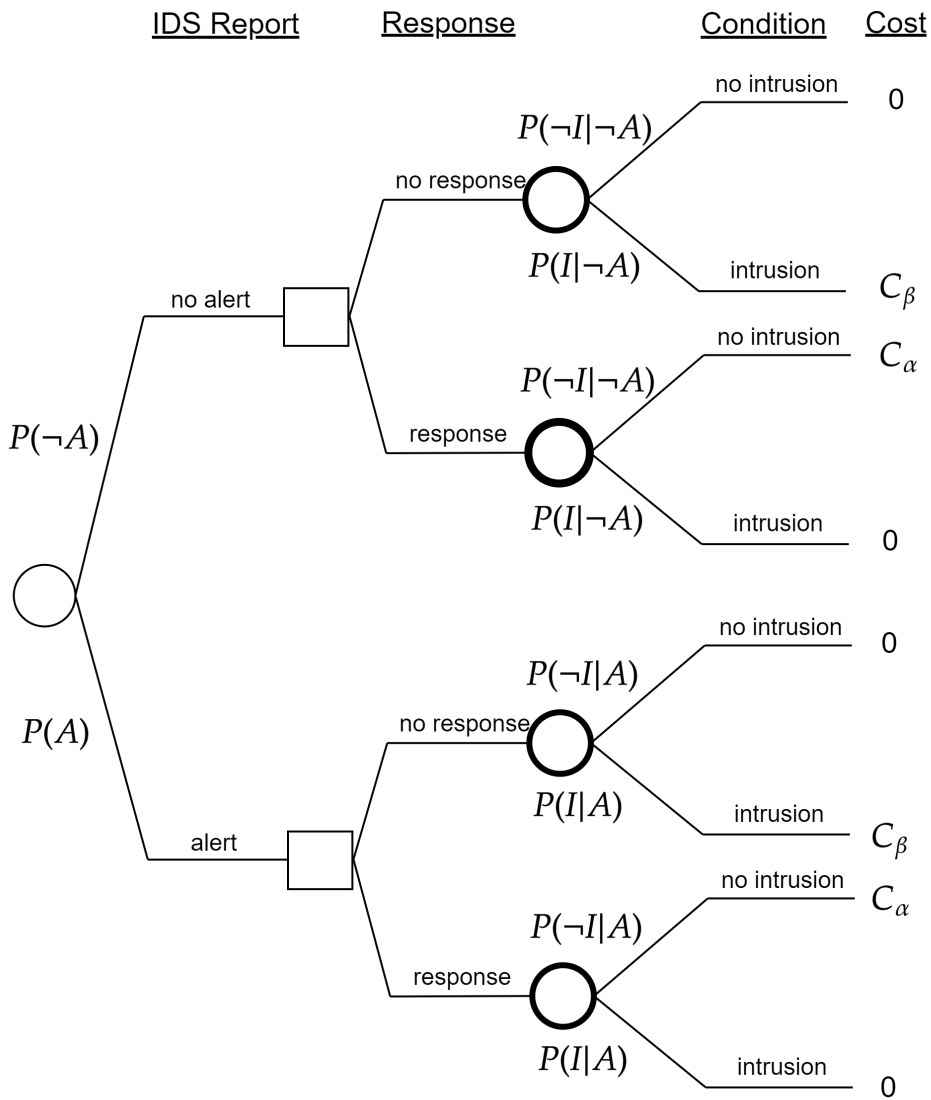
IDS decision tree:



- **Respond to IDS report (decision node)**
- This tree can decide to react without an alert
- This tree can also decide to ignore the alert

IDS Evaluation

IDS decision tree:



- **Intrusion Condition (event node)**
  Note the conditional probabilities on the tree

Top branch:

$$P(\neg I|\neg A) = NPV =$$

$$\frac{(1-B)*(1-FP)}{(1-B)*(1-FP)+B*(1-TP)}$$

$$P(I|\neg A) = 1 - P(\neg I|\neg A)$$

Bottom branch:

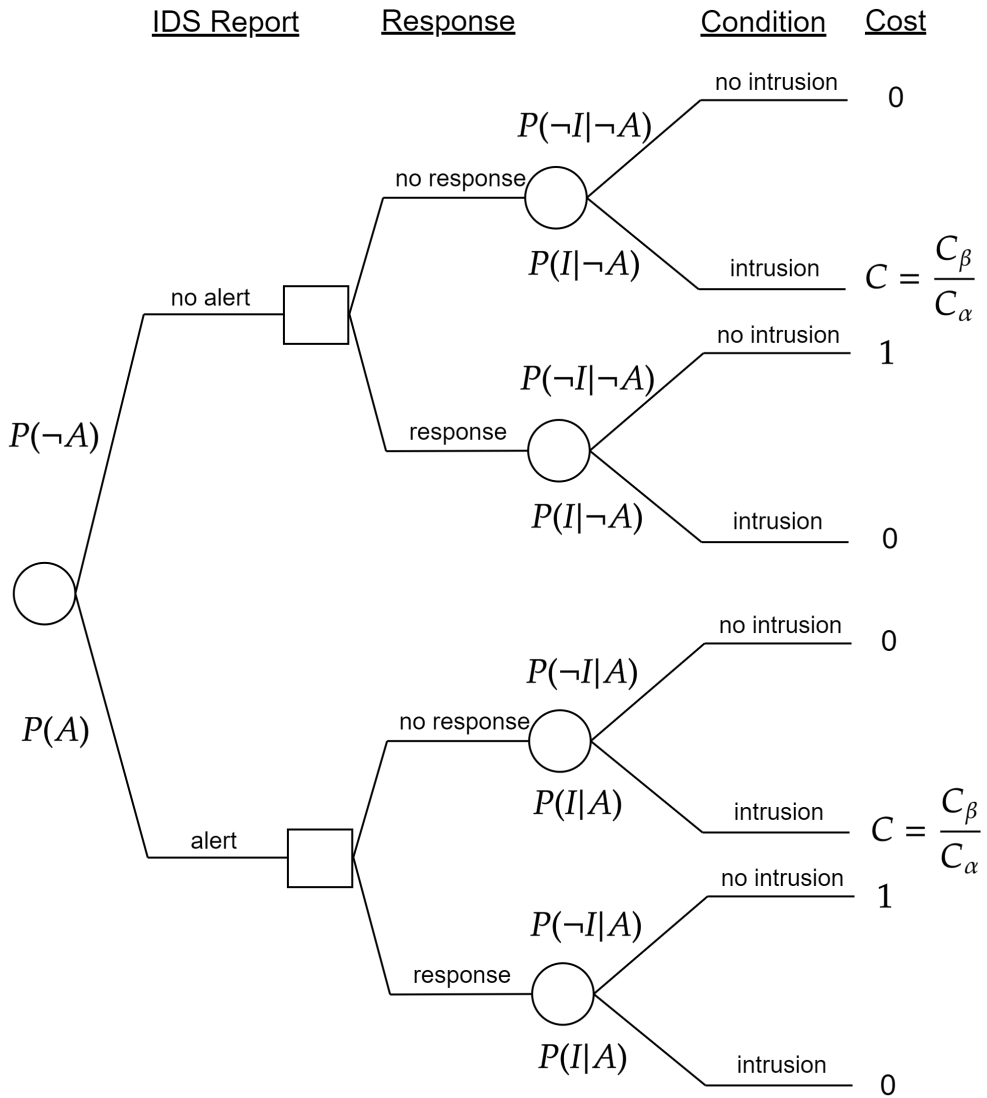
$$P(I|A) = PPV =$$

$$\frac{B*TP}{B*TP+(1-B)*FP}$$

$$P(\neg I|A) = 1 - P(I|A)$$

IDS Evaluation

IDS decision tree:

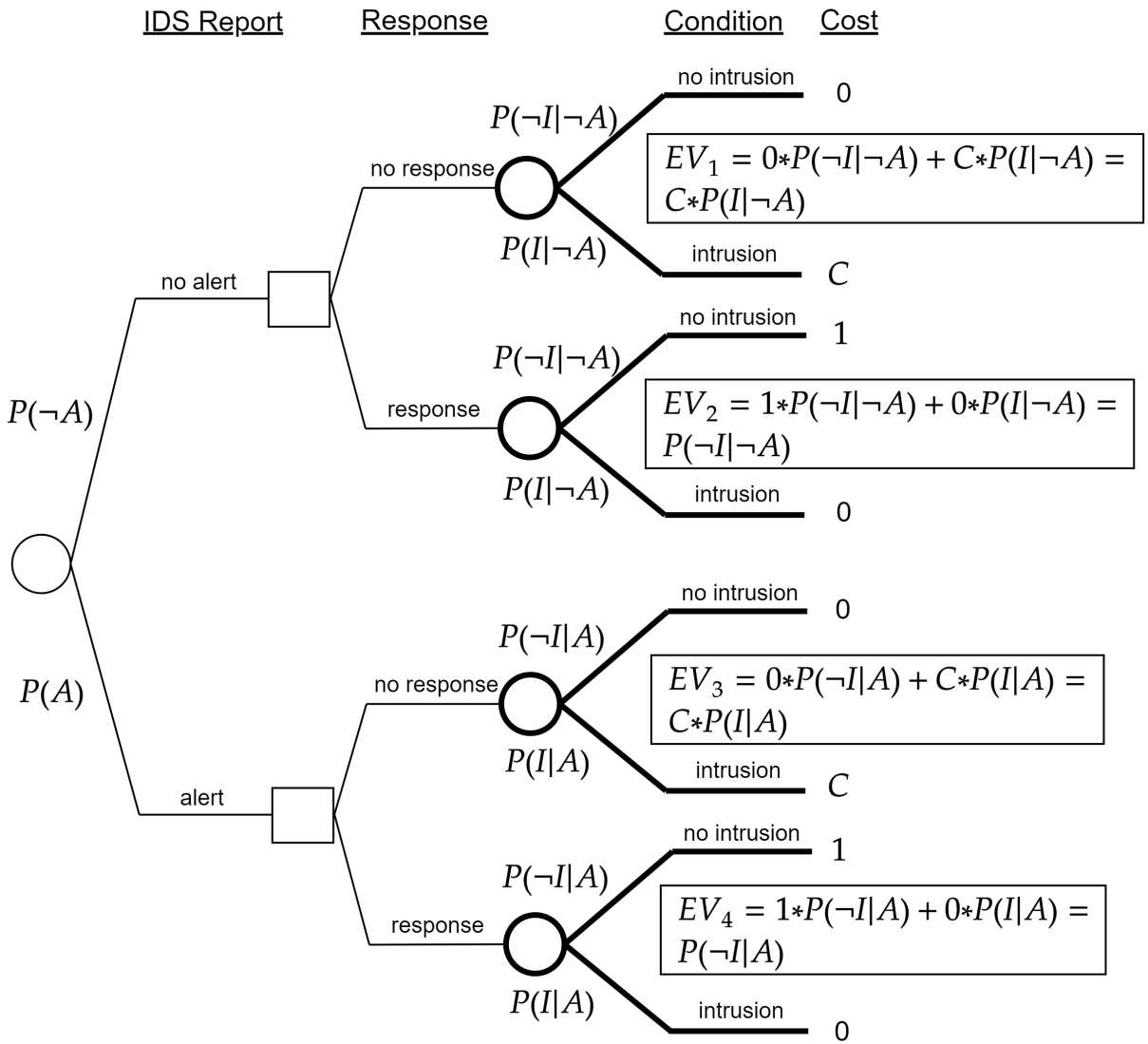


- **Cost of** *FP*, *FN*
- C_α is the cost to respond when there is no intrusion
- C_β is the cost to not respond when there is an intrusion
- $C = \frac{C_\beta}{C_\alpha}$ is the cost ratio
  e.g. if $C = 10^{-6}$ in a network, the cost of getting hacked is 10^6 times more than the cost of network administrator performing checks

IDS Evaluation

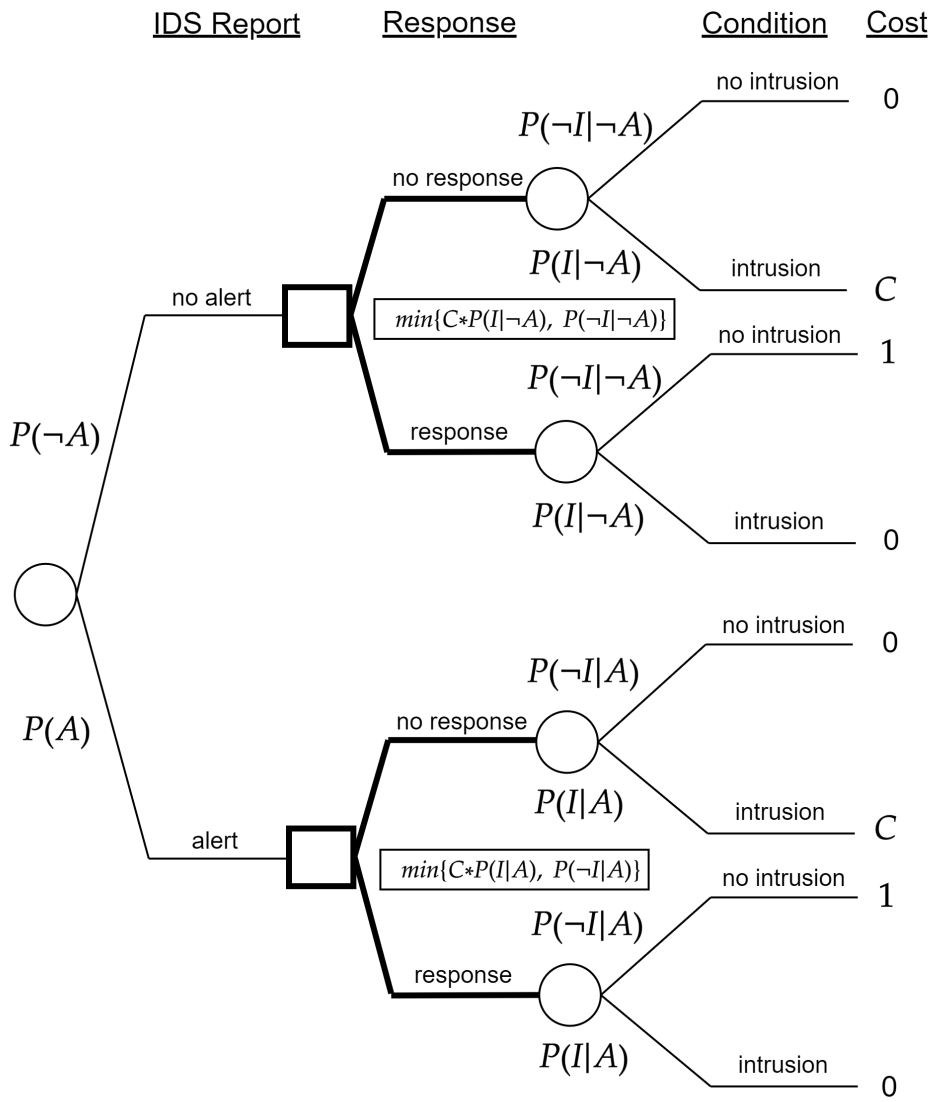
- We can compute the expected cost of the IDS by performing a tree rollback

Rollback Step 1:



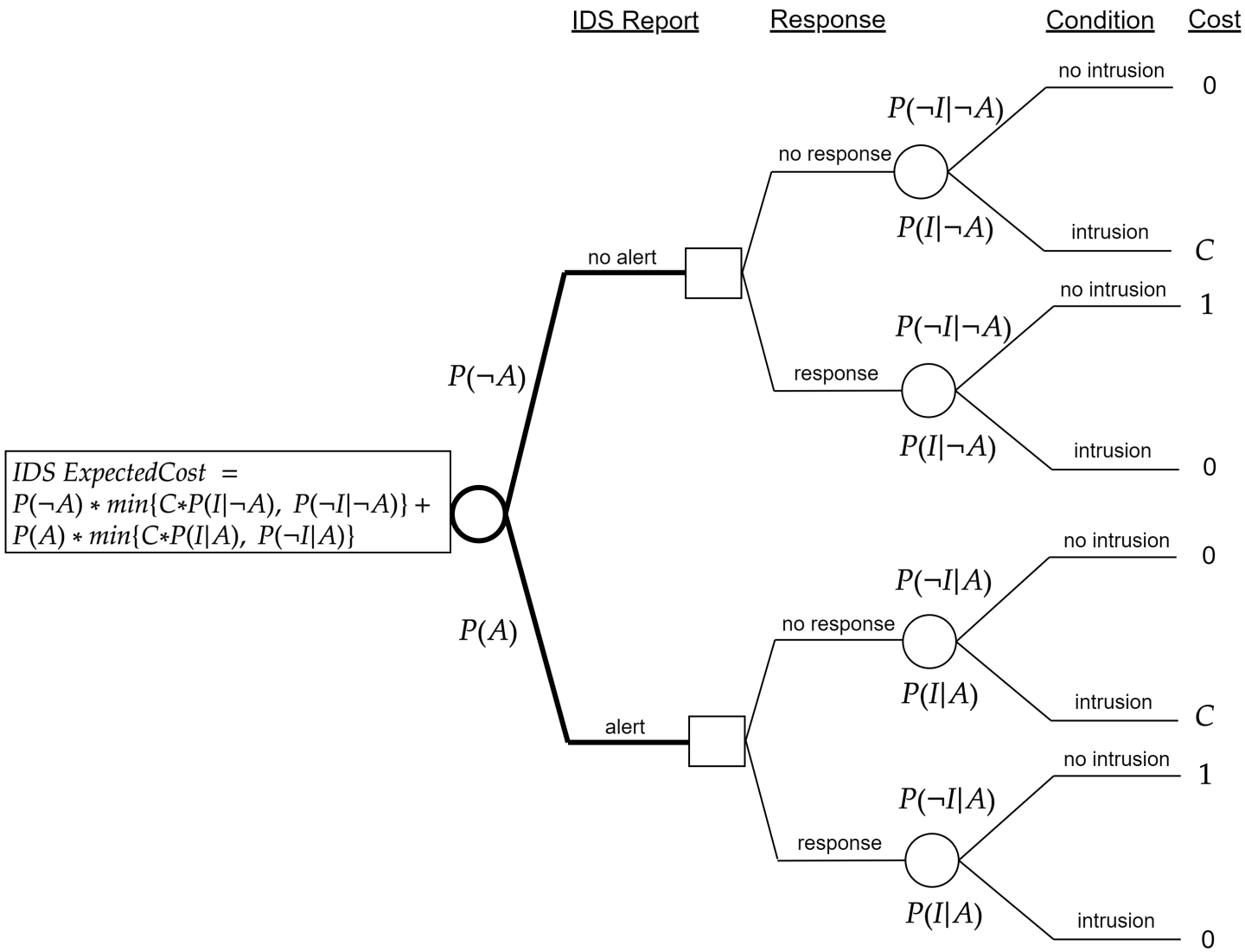
IDS Evaluation

Rollback Step 2:



IDS Evaluation

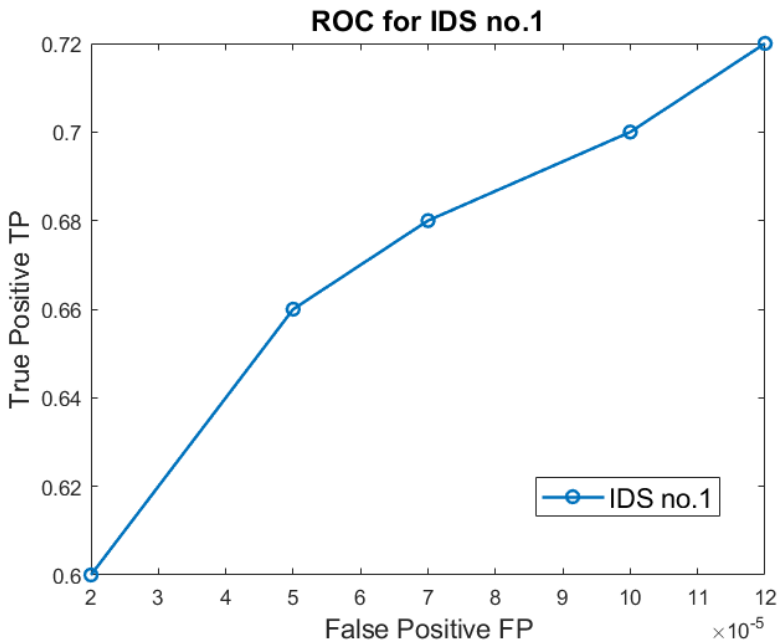
Rollback Step 3:



IDS Evaluation

Example: Optimal operating point

We have developed an IDS and after trying various rules/thresholds, we generated the following ROC curve, consisting of the (TP, FN) pairs described in the table. The base rate $B = 8\%$ and the cost ratio $C = 1000$. Using the IDS decision tree, find the optimal operating point in the curve.



TP	FP $*10^{-5}$
0.72	15
0.7	10
0.68	7
0.66	5
0.6	2

(TP, FP) pairs for various IDS configurations

IDS Evaluation

- We use the IDS expected cost derived from the decision tree:

$$\text{Cost} = P(\neg A) * min\{C * P(I|\neg A), P(\neg I|\neg A)\} + P(A) * min\{C * P(I|A), P(\neg I|A)\}$$

- We can simplify to:

$$\text{Cost} = P(\neg A) * min\Big\{C * \frac{P(\neg A|I)P(I)}{P(\neg A)}, \frac{P(\neg A|\neg I)P(\neg I)}{P(\neg A)}\Big\} +$$

$$P(A) * min\Big\{C * \frac{P(A|I)P(I)}{P(A)}, \frac{P(A|\neg I)P(\neg I)}{P(A)}\Big\} =$$

$$min\{C * FN * B, TN * (1 - B)\} + min\{C * TP * B, FP * (1 - B)\}$$

IDS Evaluation

- For every ROC point (TP, FP), we compute $FN = 1 - TP$ and $TN = 1 - FP$ and then the IDS Cost, producing the following table:

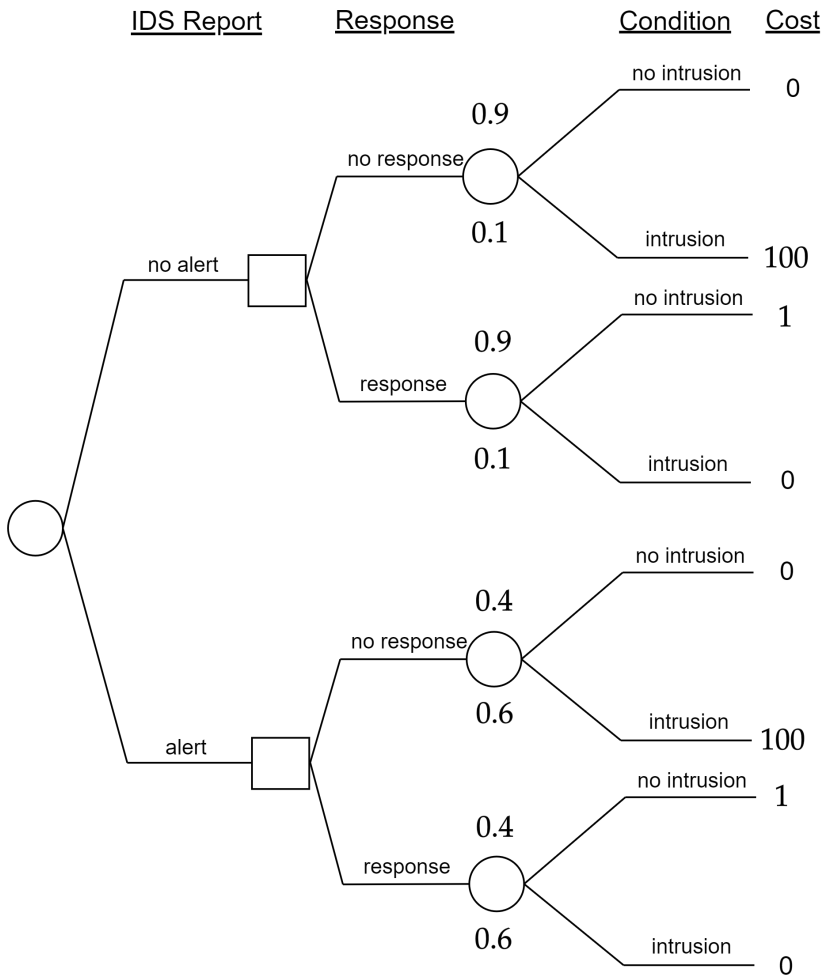
TP	FP $*10^{-5}$	Cost
0.72	15	0.4481
0.7	10	0.4801
0.68	7	0.5121
0.66	5	0.5440
0.6	2	0.6400

- The best operating point for the IDS is the one that minimizes the cost i.e. the point $(TP = 0.72, FP = 15 * 10^{-5})$
- Verify these steps using Matlab code: `optimal_operating_points.m`

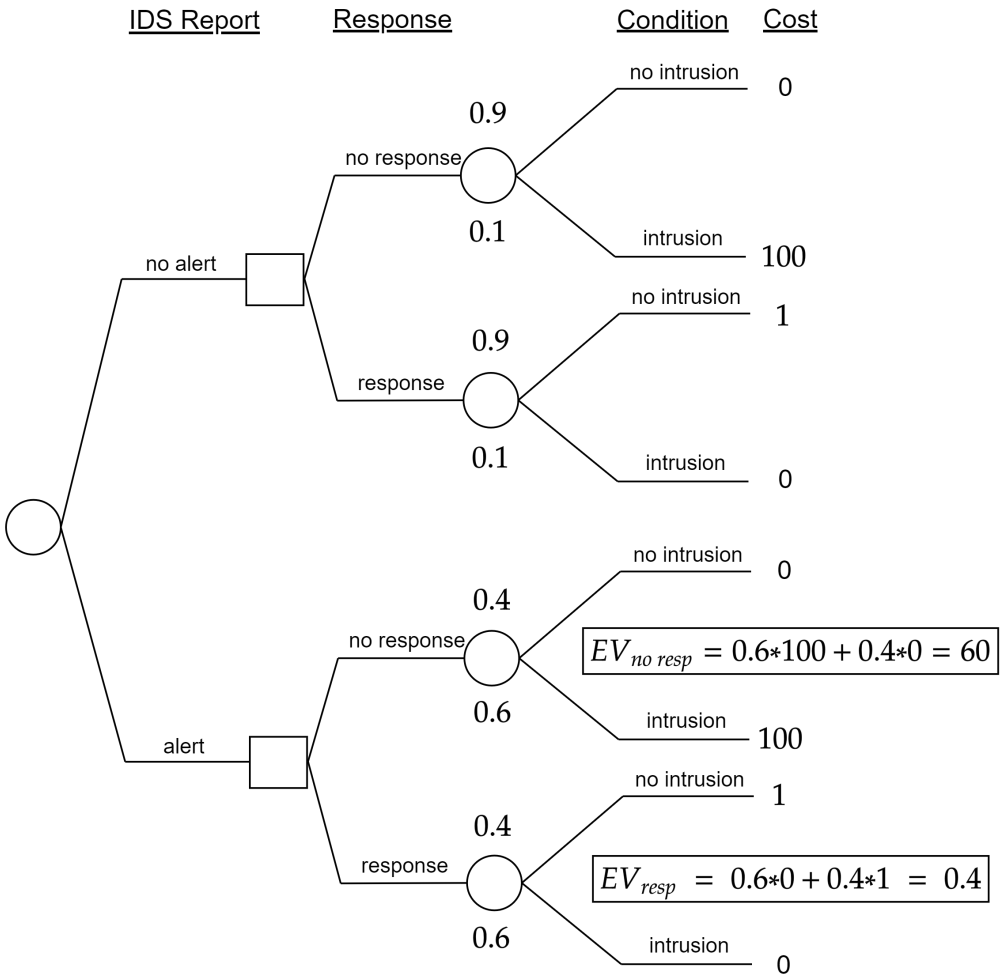
IDS Evaluation

Example: Respond or Ignore?

Given that there is an alert, find the if the the following tree decides to respond to the alert or not.



IDS Evaluation



- Perform a rollback on the bottom branch
- See that $EV_{noresp} > EV_{resp}$
- The cost of not responding is larger than the cost of responding to the alert
- The decision node choose the branch that minimizes the cost
- Thus the IDS will decide to respond to the alert

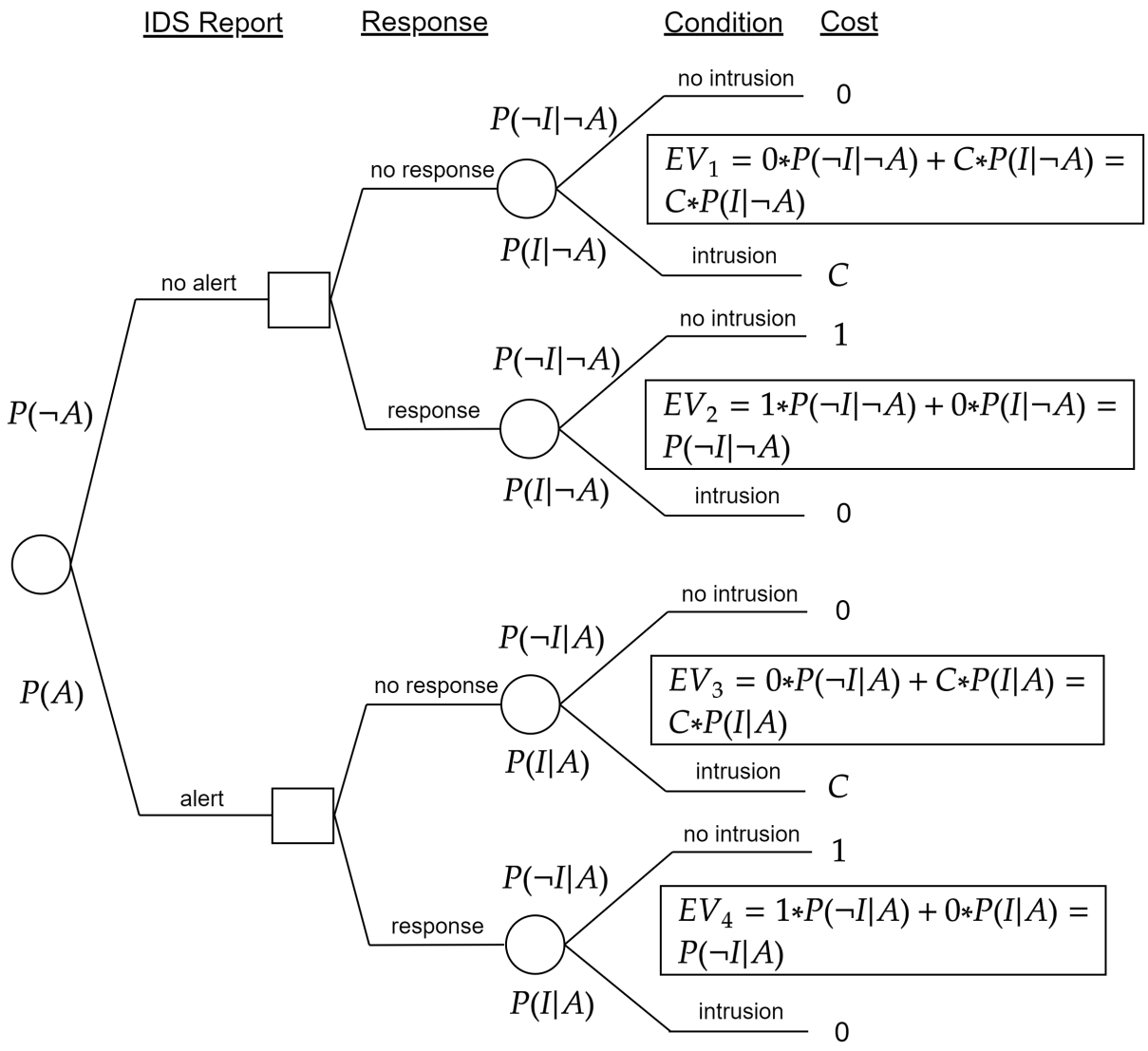
IDS Evaluation

Example: IDS Response

What are the conditions such that the becomes IDS useless?
That is, what is condition such that the IDS chooses to not respond when there is an alert. Likewise, what is the condition such that the IDS chooses to respond when there is no alert.

- We perform the tree rollback
- We state the conditions that determine the behavior of the 2 decision nodes

IDS Evaluation



IDS Evaluation

- Given that there is no alert (top branch), the IDS will still choose to respond when $EV_1 > EV_2$ i.e. when

$$C * P(I|\neg A) > P(\neg I|\neg A)$$

- Given that there is an alert (bottom branch), the IDS will ignore it (no response) when $EV_3 < EV_4$ i.e. when

$$C * P(I|A) < P(\neg I|A)$$

IDS Evaluation

Example: IDS comparison

Let the following two IDSs with the respective ROC curve pairs (TP, FP). Using a cost-based analysis, find which is the best IDS when the base rate $B = 0.05$ and the cost ratio $C = 100$.

IDS no. 1	
TP	FP
0.82	0.0015
0.85	0.0020
0.87	0.0025

IDS no. 2	
TP	FP
0.82	0.002
0.83	0.003
0.95	0.005

IDS Evaluation

- To find the best IDS, we need to first find the best (TP, FP) pair for both IDSs, using the cost formula

$$Cost = min\{C * FN * B, TN * (1 - B)\} + min\{C * TP * B, FP * (1 - B)\}$$

IDS no. 1		
TP	FP	Cost
0.82	0.0015	0.9014
0.85	0.0020	0.7519
0.87	0.0025	0.6524

IDS no. 2		
TP	FP	Cost
0.82	0.002	0.9019
0.83	0.003	0.8529
0.95	0.005	0.2548

- Subsequently, we find which IDS has the lowest cost at its optimal point
- The optimal point for IDS no. 1 is $(0.87, 0.0025)$ and for IDS no.2 is $(0.95, 0.005)$. The respective costs are 0.6524 and 0.2548, thus the we prefer IDS no.2