

# Managing Alert Rates

## Introduction

When *Detect* generates an alert it means that a computed risk measure for this transaction has exceeded a threshold. Choosing this threshold is critical to getting the best performance from the system. It also allows you to manage the alert rate so that it is appropriate for the staffing levels available to process alerts.

As the threshold is lowered the number of alerts will go up and similarly as it raised the number will drop. In the extreme, if we set the threshold to zero then all transactions will be alerted, the alert rate will be very high and there will be a huge number of *false positives*. Conversely, setting the threshold to one will mean nothing gets alerted but there will be no false positives.

With the threshold set to zero we will have detected 100% of the fraud but at the cost of far too many false positives. With the threshold set to one we will have detected no fraud at all but will have had no false positives.

In between these two extremes the amount of fraud detected depends on two things:

- the chosen threshold
- the quality of the system

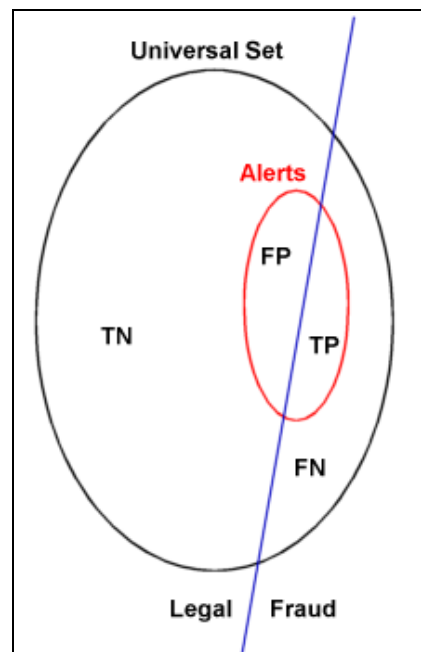
The quality of the system is a measure of the reliability of the alerts. A simple measure is the ratio of correct alerts to the total number of alerts. Unfortunately the quality of the system, measured in this simple way, changes as the threshold changes. To get a true picture of the overall system quality it is necessary use techniques like, for instance, ROC analysis (see appendix A).

## Measuring Performance

The following defines the metrics used to measure performance in the rest of this document.

On the right is a Venn diagram representing all the transactions. The red circle represents those transactions which are alerted.

True Positive (TP)	Fraud	- alerted
False Positive (FP)	Not Fraud	- alerted
True Negative (TN)	Not Fraud	- not alerted
False Negative (FN)	Fraud	- not alerted



## Useful Metrics

The table below list some of the metrics used

<b>True Positive Fraction</b> (fraction of fraud, correctly alerted)	$TPF = \frac{TP}{TP + FN} = \frac{TP}{F}$
<b>False Positive Fraction *</b> (fraction of legal, wrongly alerted)	$FPPF = \frac{FP}{FP + TN} = \frac{FP}{L}$
<b>True Positive Alerts</b> (ratio of correct alerts to total alerts)	$TPA = \frac{TP}{TP + FP} = \frac{TP}{A}$
<b>False Positive Alerts</b> (ratio of wrongly alerted to total alerts)	$FPA = \frac{FP}{TP + FP} = \frac{FP}{A}$
<b>False Positive Ratio *</b> (ratio of wrong alerts to correct alerts)	$FPR = \frac{FP}{TP} \quad FPR = \frac{1 - TPA}{TPA}$

\*

Note: the false positive fraction is often referred to in the literature and by companies as the false positive ratio. It is not always clear which definition is being used. In the following we use the above definitions.

## Choosing a threshold

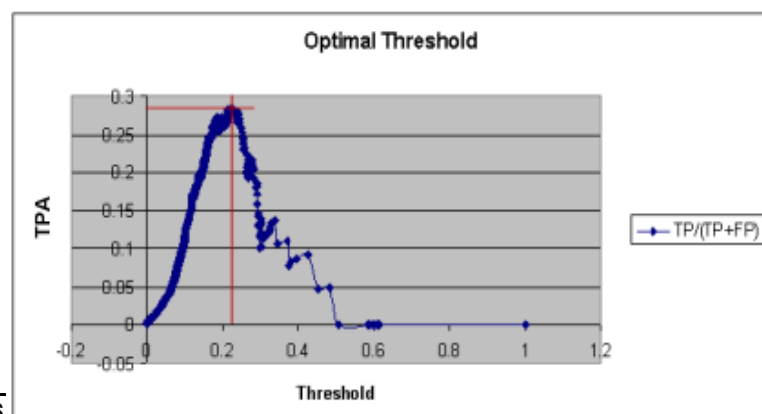
If we plot the True Positive Alerts (TPA) against threshold we will usually get a graph like the one below:

You will see that there is an optimal threshold at which the ratio of correct alerts is maximal. At this point the quality of the system is at its best in terms of the percentage of alerts that are correct. As the threshold is reduced the total number of alerts will increase but so will the number of false positives and so the quality, as we have defined it, reduces. As the threshold is increased a more complicated effect starts to assert itself; the total number of alerts is reduced but the quality of those alerts is also reduced because of the sparseness and quality of the training data.

This latter effect can be dramatically reduced by pre-processing the training data and by the introduction of a technique that we have developed called *predictive tagging*. These techniques also improve the overall performance of the system.

In practice, we often need to select a threshold that gives as few false-positives as possible but with the best detection or true-positives. This is the same as saying the highest *true positive alerts*. However, it should be borne in mind that the detection rate or *true positive fraction* at this threshold may not be as high as required. To detect more fraud we necessarily have to compromise on the quality and expect more false positives. Ultimately, the threshold chosen must also be based on the resources available to process alerts.

Detect provides three alert levels. In general, the



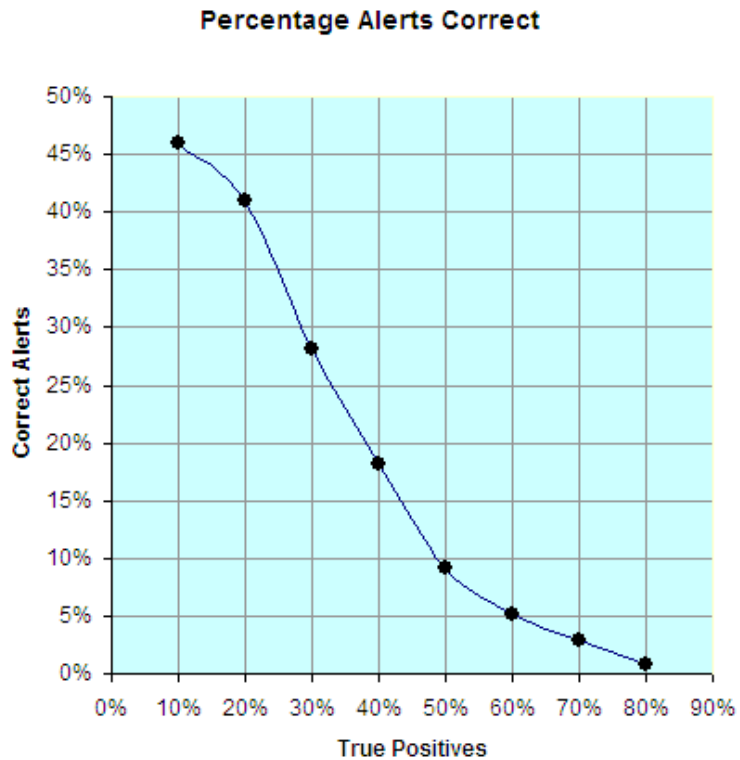
medium level alerts should be set to the optimal threshold discussed above. The high level alerts can then be set at a higher threshold and the lower level alerts at a lower level. In this way operators are able to manage the alerts more effectively.

## Some Typical Figures for Detect’s Risk Engine

The performance of *Detect*, like any machine-learning system, is dependent on the quality of the training data, including its consistency and the amount of information available in each transaction. The following figures are based on issuer data that contained the transaction fields described in appendix B. The figures measure the performance of the Risk Engine alone. Of course, in conjunction with patterns, black lists and white-lists *Detect*’s overall performance is even higher than the figures quoted below.

The graph is a percentage plot of the True Positive Alerts (percentage correct alerts) against the True Positive Fraction (percentage of fraud caught)

The graph shows how the TPA changes with different overall detection rates (TPF). On the left of the graph (where the alert threshold is very high) almost 50% of alerts are correct but very little of the overall fraud is captured. At the other extreme, 80% or more of fraud can be detected at the cost of many false alerts.



At the 70% TPF (fraud caught) level the percentage of correct alerts (TPA) is 3.2%.

The FPR is then just  $(1 - \text{TPA}) / \text{TPA}$  and is equal to 30 :1

## Appendix A - Measuring Overall Performance

This section describes an approach for measuring the overall performance of the system. *Detect* is essentially a classifier of fraud. However, fraud is a rare-event and hence these measures always need to be treated carefully.

### Receiver Operating Characteristic (ROC) Curves

This is a technique for measuring the effectiveness of a classification process. It has its origins in signal processing but is now widely used in statistical medicine to determine the effectiveness of tests and treatments.

Consider a rule that returns a value between 0 and 1 which is a measure of evidence for fraud. Call this measure  $c$ . Then for this rule to act as a classifier we need to decide on a threshold  $t$  for  $c$  above which we decide it is fraud.

If we set  $t$  to be 0 then every transaction is classified as fraud, conversely, if we set it to 1 then no transactions are classified as fraud. So as we move the threshold from 0 to 1 we can observe the behaviour of various metrics like false positive ratio (FPR) and plot graphs of how these metrics change relative to each other.

The ROC curve is just such a graph and is a plot of True Positive Fraction (TPF) against False Positive Fraction (FPF) for different threshold values  $t$ , where

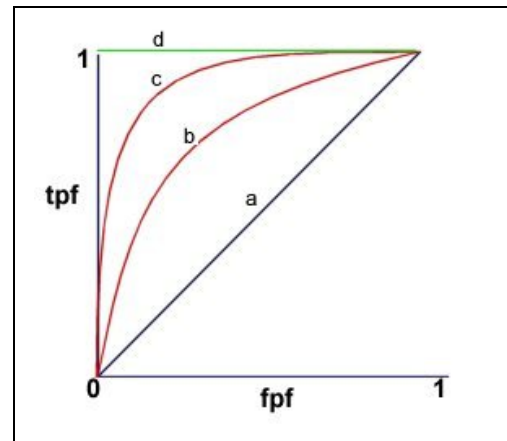
$$TPF = \frac{TP}{TP + FN} = \frac{TP}{F} \text{ and } FPF = \frac{FP}{FP + TN} = \frac{FP}{L}$$

To the right is a typical ROC curve.

Consider first the case of the straight-line (a). This means that as we moved the fraud decision threshold from 0 to 1 that  $TPF = FPF$ :

$$\text{So } \frac{TP}{F} = \frac{FP}{L} \text{ and hence } \frac{TP}{FP} = \frac{F}{L}$$

This means that the probability of the classifier getting it right is equal to the probability of fraud itself. It is essentially a random sampling.



The best classifier would be where:

$$\frac{TP}{F} = 1, \frac{FP}{L} = 0 \text{ and hence } TP = F \text{ and } FP = 0$$

This would be represented by the graph (d).

The red graphs (b) and (c) therefore represent rules of increasing ability to discriminate fraud.

You would not expect a rule to produce results below graph (a) as this would indicate a bias towards getting it wrong, however it is not necessary to assume that a rules behaviour is always positive (see below)

In general we can say that the greater the area under the ROC curve the better the rule is at discriminating fraud.

The ROC curve has many useful properties not least of which is it makes no assumptions about the underlying populations.

## The GINI value

The GINI is defined as twice the area under the ROC curve minus one.

This is a more convenient measure as it has the property that when a classifier is very bad (ie random) then the GINI is 0.

So we have  $G = 2A - 1$  and as  $A \in [0, 1]$  then  $G \in [-1, +1]$

## What is a bad classifier in terms of the ROC curve ?

A bad classifier is one that randomly classifies an event. If the probability of fraud is 0.002 (1 in 500 are fraud) then we would expect the classifier to get it right with a probability of 0.002.

This is represented by the diagonal line on a ROC graph.

If a classifier has a bias towards getting it wrong then it too may be regarded as a good classifier... it is discriminating fraud. It will have a curve that will tend to be below the diagonal.

For a perfectly positive classifier ( $A = 1$ ) we have  $G = 1$ .

For a perfectly negative classifier ( $A = 0$ ) we have  $G = -1$ .

However, if  $G < 0$  ie  $A < 0.5$  then the probability of fraud as determined by the classifier is not  $c$  (say) but  $(1 - c)$ . (Its actually classifying legal).

This can be taken into account when combining the output of rules (see section 4)

Note: Where a curve crosses the diagonal then it is behaving as a +ve discriminator for some ranges of  $t$  and as a -ve discriminator for other ranges of  $t$ . In general therefore it is not a good rule and the above formula is still valid. (A rule that behaves in this way should be inspected as it could be split into two rules. This is going to be due to non-linearity's in the way the rule generates its probability.)

## Appendix B – Transaction Fields

The fields of the transactions used to measure the performance of *Detect*.

Field	Notes
txn_id	unique transaction identifier
org_code	organisation code
cardaccount	credit card number
txn_date	transaction date and time (received at issuer)
mcc	merchant category code
mer_country	merchant country
mer_code	merchant code
value_normal	normalised value (amount)
value_orig	value in original currency
request_code	iso request code
reply_code	iso reply code
pos_cond_code	iso pos condition code
fraud_code	used to tag historic transactions: null or fraud type