13th - 17th May 2019 – Kuala Lumpur

Fraud Model Development and Deployment in SAS FM

Session 6: Performance Evaluations

Performance Evaluations

Topics

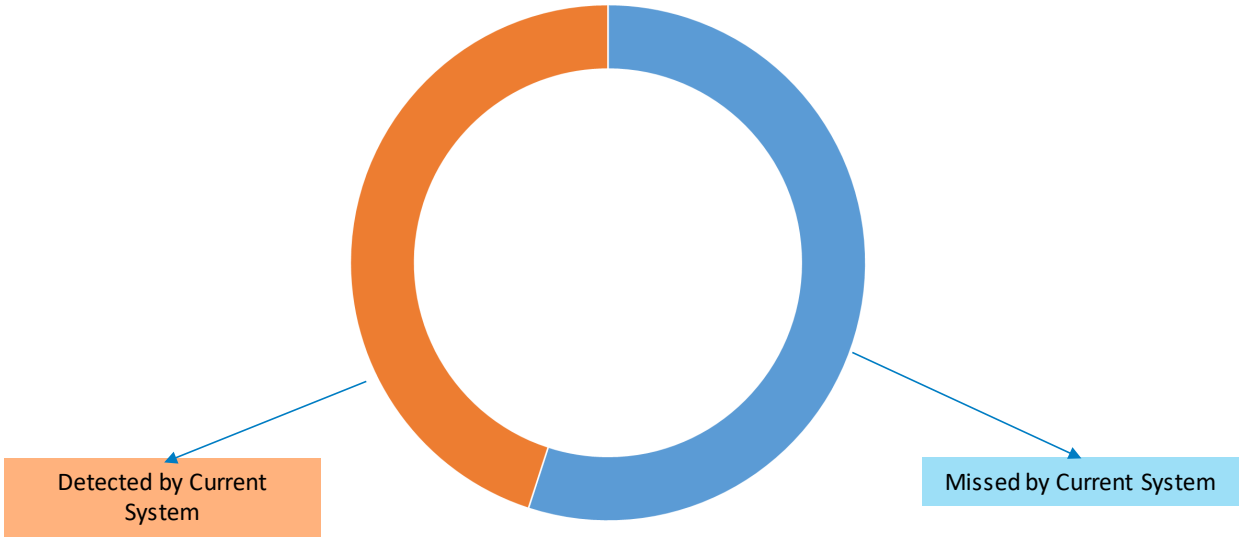
- Anatomy of model performance
- Case level metrics
- Transaction level metrics
- Transaction vs case level performance

Performance Evaluations

Anatomy of Model Performance

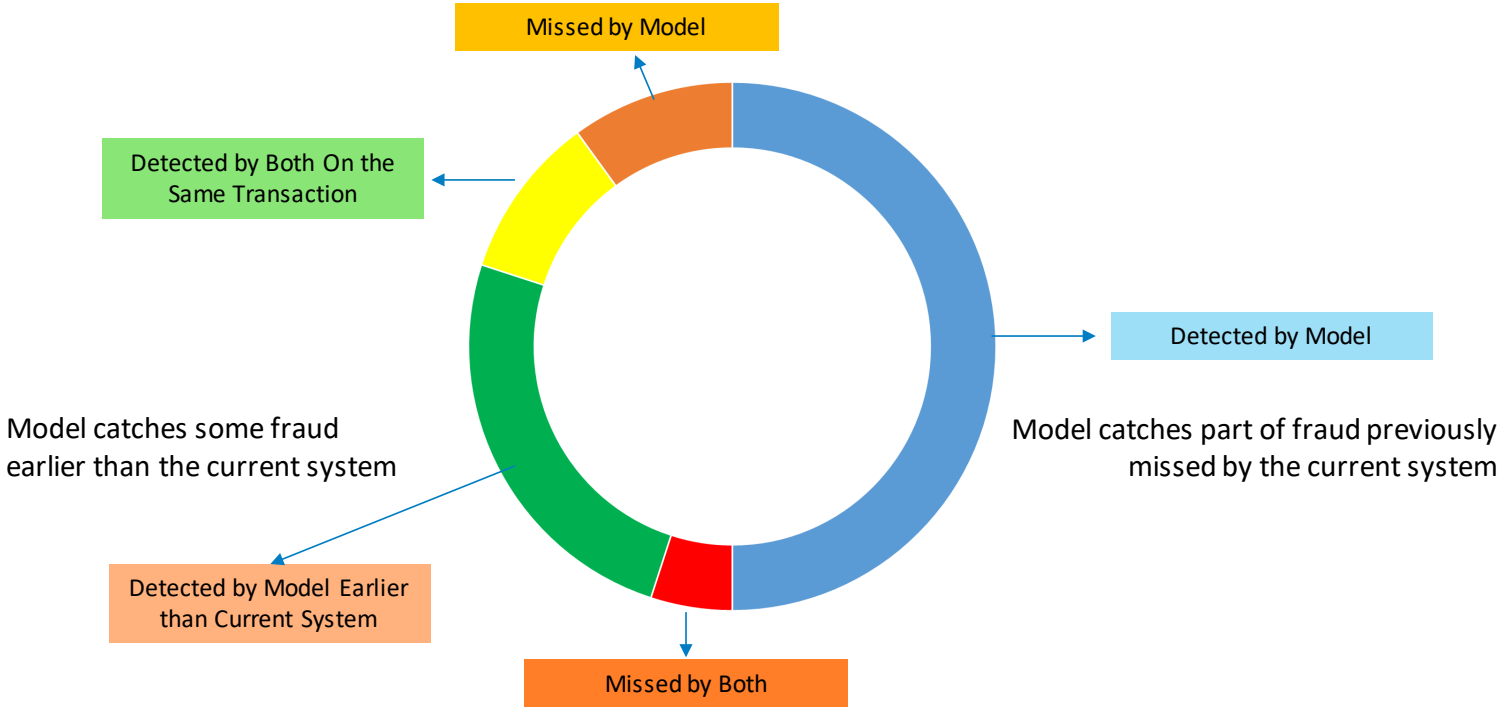
Performance Evaluations

Anatomy of Model Performance



Performance Evaluations

Anatomy of Model Performance



Performance Evaluations

Notes on Evaluation

- Model performance evaluations are always performed as a function of the model score
  - Starting from highest score (999) going to lowest score (1)
- Metrics defined by considering transactions that score at or above a score threshold. i.e. *metric(s)* denotes the value evaluated on transactions that score at or above threshold *s*
  - Score of 1000 is considered to have 0 true positives and 0 false positives since not a single transaction will score at that level
  - Score of 1 will have 100% true positives and 100% false positives since every transaction will have scored at or above 1.
- Some rate/ratio based metrics are already normalized whereas some metrics such as volume may require normalization (for e.g. per day)

Performance Evaluations

Approved and Declined Transactions

- There are 3 points at which an approve / decline (A/D) decision can be made in the flow :
  - Upstream prior to entering SAS FM
    - Typical policy declines such as invalid authentication, insufficient balance, expired cards etc.
  - By the rules within SAS FM
    - Based on model score based rules and other fraud rules
  - Downstream after exiting SAS FM
    - Policy declines, additional fraud processing
- Depending on where SAS FM is placed in the flow, there may be (or not) upstream or downstream decisions
  - Typically we try to place SAS FM as the last decision point so that all prior decisions are available to us

Performance Evaluations

Approved and Declined Transactions

- Upstream decisions are typically mapped to a field within the incoming message
  - E.g. tca_auth_sys_dec / tck_tran_status / tbt_tran_status etc.
- The final decision code after SAS processing is returned via rrr_action_code
- If a downstream process further changes the decision, typically a duplicate message is sent back to SAS FM indicating that that the message is a duplicate, but containing the final decision code
  - Need to be considered when preparing modeling data from consortium feeds

Performance Evaluations

Approved and Declined Transactions

- For first time models, the final decision code available in the historical data will be the only decision code to be considered
- However when using consortium data from consortium for existing models, which decision code to use for model development and evaluations is a critical consideration
- For defining the fraud window for model development, the final decision codes should be used
  - Since it represents the true active fraud state
  - Fraud window is extended to include additional potentially fraud transactions fraud had not been blocked.

Performance Evaluations

Incremental vs. Replacement Evaluations

- An **incremental** evaluation measures the additional model benefit when the model is placed on top of the existing fraud model [+] (including any incumbent SAS FM model)
  - i.e. only the frauds missed by the existing model form the universe of fraud under consideration
- However the model should be evaluated as if it will replace the existing fraud model
  - Model should still get credit for catching fraud that was caught by the previous model and vice versa
- **Replacement** evaluation assumes that the initial post-block declines are real fraud activities had the current model not blocked the fraud episode
  - Some of these declines are treated as approved fraud transactions during evaluations (refer to section on replacement windows in session 4)

+ "Model" also includes rules and other mechanisms when evaluating a first time model

Performance Evaluations

Incremental vs. Replacement Evaluations

- This means that for evaluation purposes, we should use only the upstream decision if evaluating against an existing model
  - The final decision code will manifest the effect of the incumbent model score and will result in a partial degeneration to an incremental evaluation
  - May need to redefine the fraud window based on upstream decision codes
- We are still performing an incremental evaluation relative to the upstream checks, but that is reasonable
  - A pure model evaluation should not consider any approve / decision code
  - But we will always deal with systems where there is an implicit or explicit fraud protection mechanism (policy declines for e.g. are implicit fraud checks)
- Unfortunately when replacing a non SAS FM model, we may not have separate upstream and post model decision codes
  - Hence the model may get evaluated against the entire incumbent system

Performance Evaluations

Transaction Level Metrics

Performance Evaluations

Transaction Level Metrics

- Detection rate / hit rate
- Impact rate
- False positive rate

Performance Evaluations

Impact Rate

- $Impact\ rate\ (s) = \frac{Number\ of\ transactions\ with\ score \geq s}{Total\ number\ of\ transactions}$
- Essentially indicates what % of transactions will be impacted by the following hypothetical rule:

```
if score >= s then decline;
```

- Essentially combines both false positives and true positives
- Very critical metric since it indicates how much operational impact the bank will have on a given day
  - Multiplying this rate by average daily transaction volumes gives the average number of transactions impacted at the given score threshold
- Transactions declined upstream may be excluded from the population since the bank may not act on those declines regardless of the score

Performance Evaluations

Detection Rate

- $Detection\ rate\ (s) = \frac{Number\ of\ fraud\ transactions\ with\ score \geq s}{Total\ number\ of\ fraud\ transactions}$
- Essentially indicates what % of fraud transactions will be detected by the following hypothetical rule:

```
if score >= s then decline;
```

- Variations of this measure can be introduced by applying different filters on the population of transactions being considered. E.g.
  - $CNP\ Detection\ rate\ (s) = \frac{Number\ of\ CNP\ fraud\ transactions\ with\ score \geq s}{Total\ number\ of\ CNP\ fraud\ transactions}$
  - $True\ Detection\ rate\ (s) = \frac{Number\ of\ approved\ fraud\ transactions\ with\ score \geq s}{Total\ number\ of\ approved\ fraud\ transactions}$

Performance Evaluations

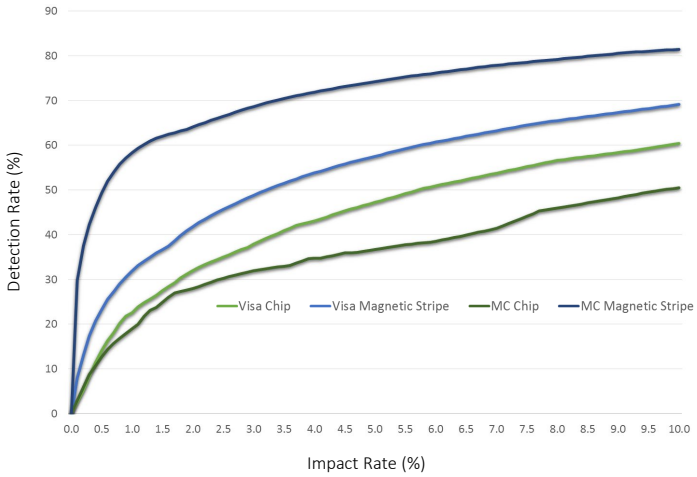
False Positive Rate

- $$False\ positive\ rate\ (s) = \frac{Number\ of\ non\ fraud\ transactions\ with\ score \geq s}{Total\ number\ of\ non\ fraud\ transactions}$$

Performance Evaluations

Transaction Level ROCs

- RoCs are mostly plotted as detection rate vs. impact rate
- In fraud we are almost always interested in the very low impact rate region (< 1 – 2%)
  - Even the smallest of banks will not be able operate above that region



Performance Evaluations

Case Level Metrics

Performance Evaluations

Case Level Evaluation Metrics

- Case / alert level detection rate
- Value detection rate
- No recontact period
- Case / alert false positive ratio
- Outsort volume
- Outsort rate

Performance Evaluations

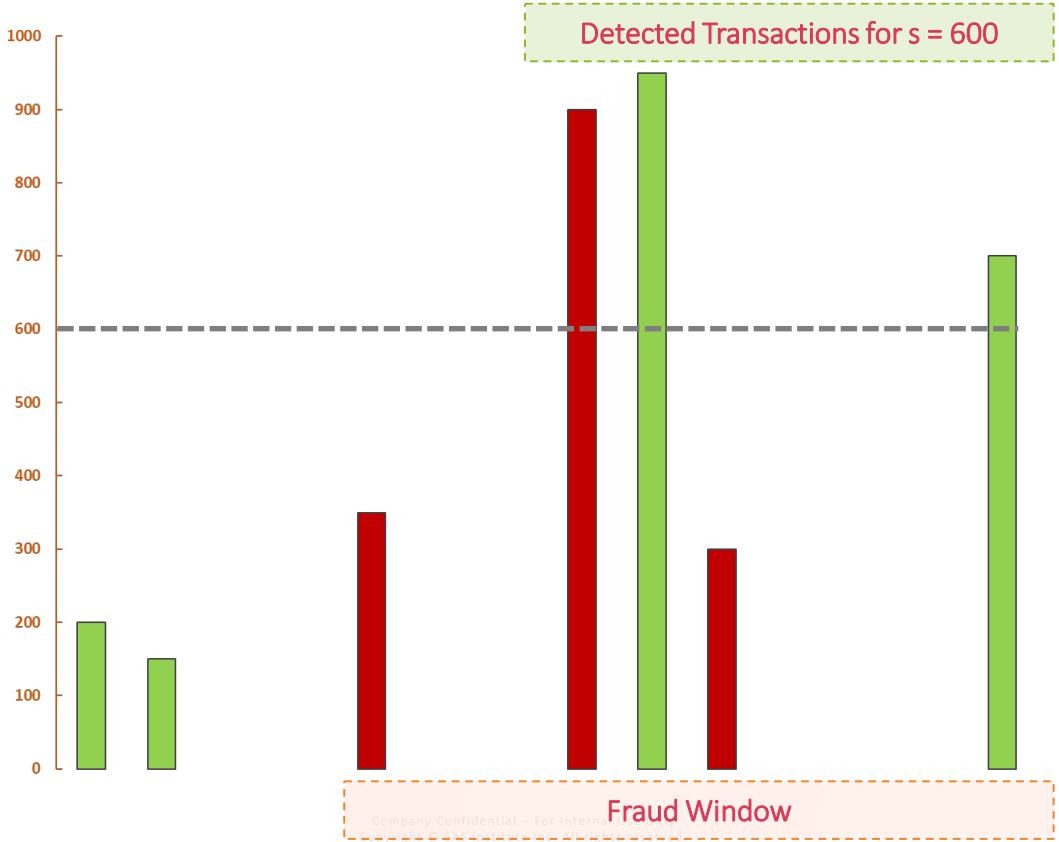
Case Detection Rate (CDR)

- CDR measures if a fraud case (episode) was detected or not
  - Not if individual transactions within the case was detected
- A fraud case is considered detected if there is at least 1 transaction that scores at or above the evaluation threshold
  - All subsequent fraud transactions are all considered detected

- $$CDR\ (s) = \frac{No.\ of\ fraud\ cases\ where\ at\ least\ 1\ transaction\ in\ fraud\ window\ scores \geq s}{Total\ number\ of\ fraud\ cases}$$

Performance Evaluations

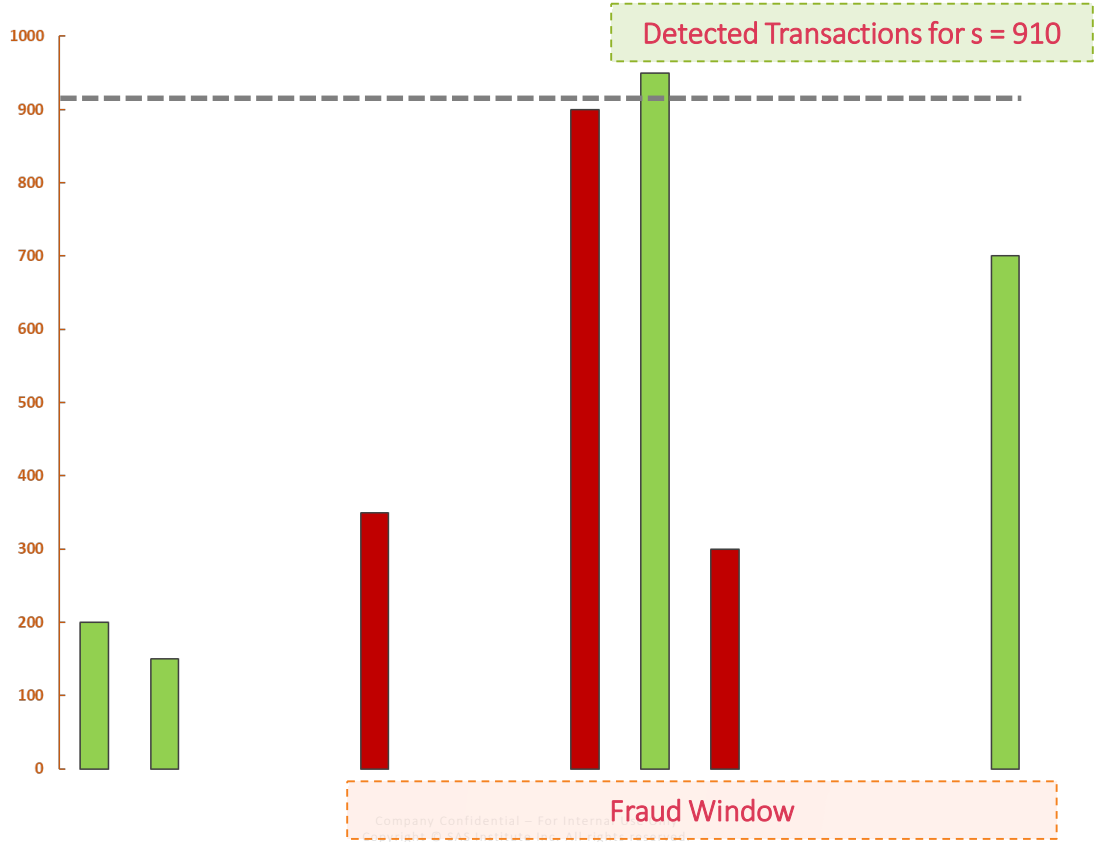
Case Detection Rate - Illustration



Performance Evaluations

Case Detection Rate - Illustration

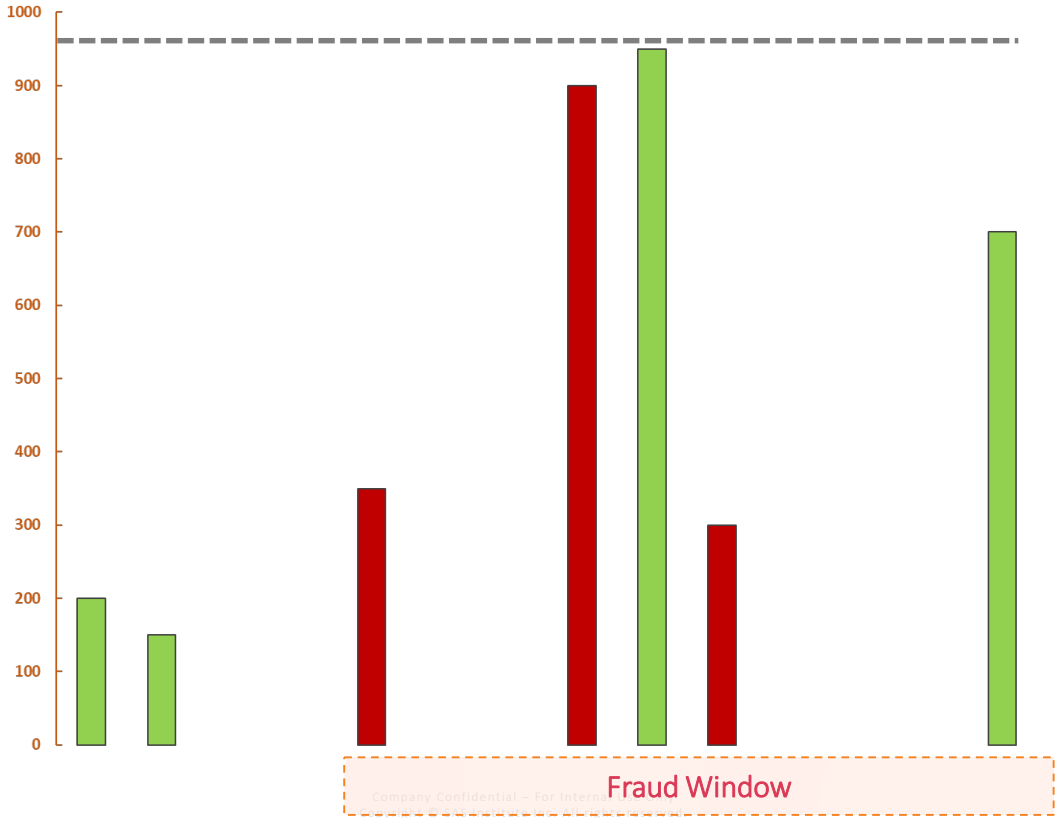
First transaction that triggers the alert need not necessarily be a fraud transaction



Performance Evaluations

Case Detection Rate - Illustration

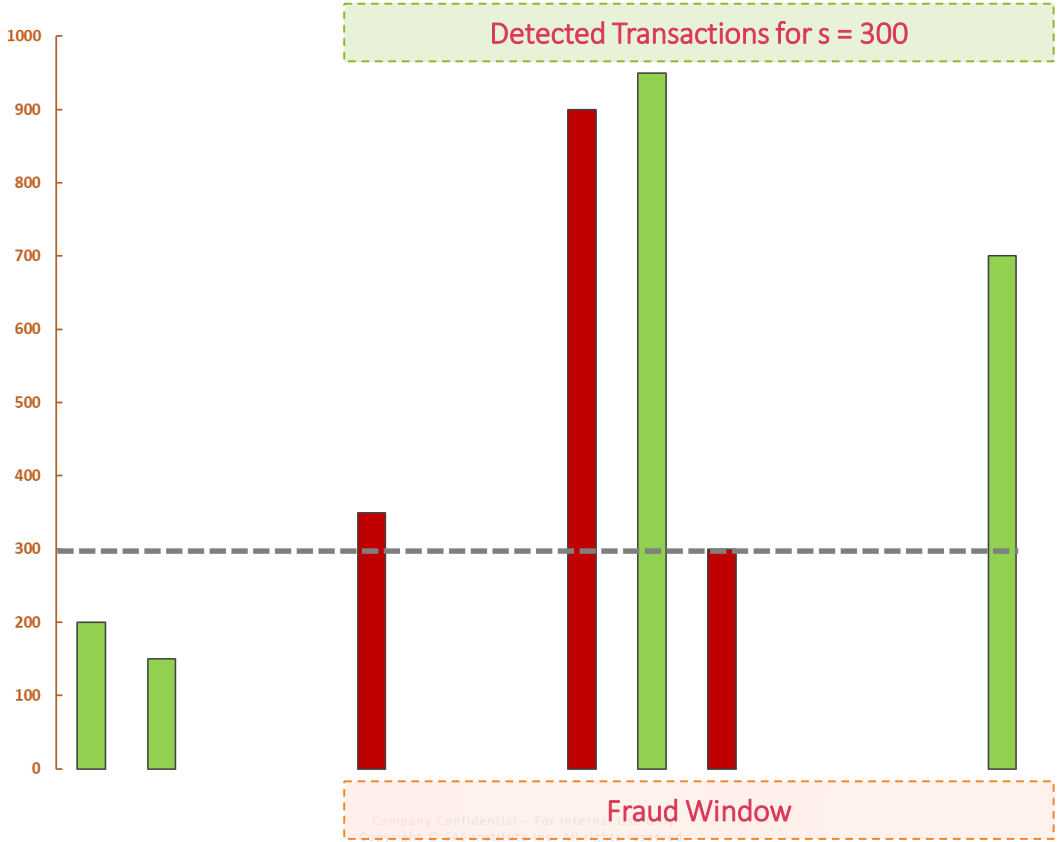
Missed detection for s = 950



Performance Evaluations

Case Detection Rate - Illustration

Case detected on first fraud transaction



Performance Evaluations

Value Detection Rate (VDR)

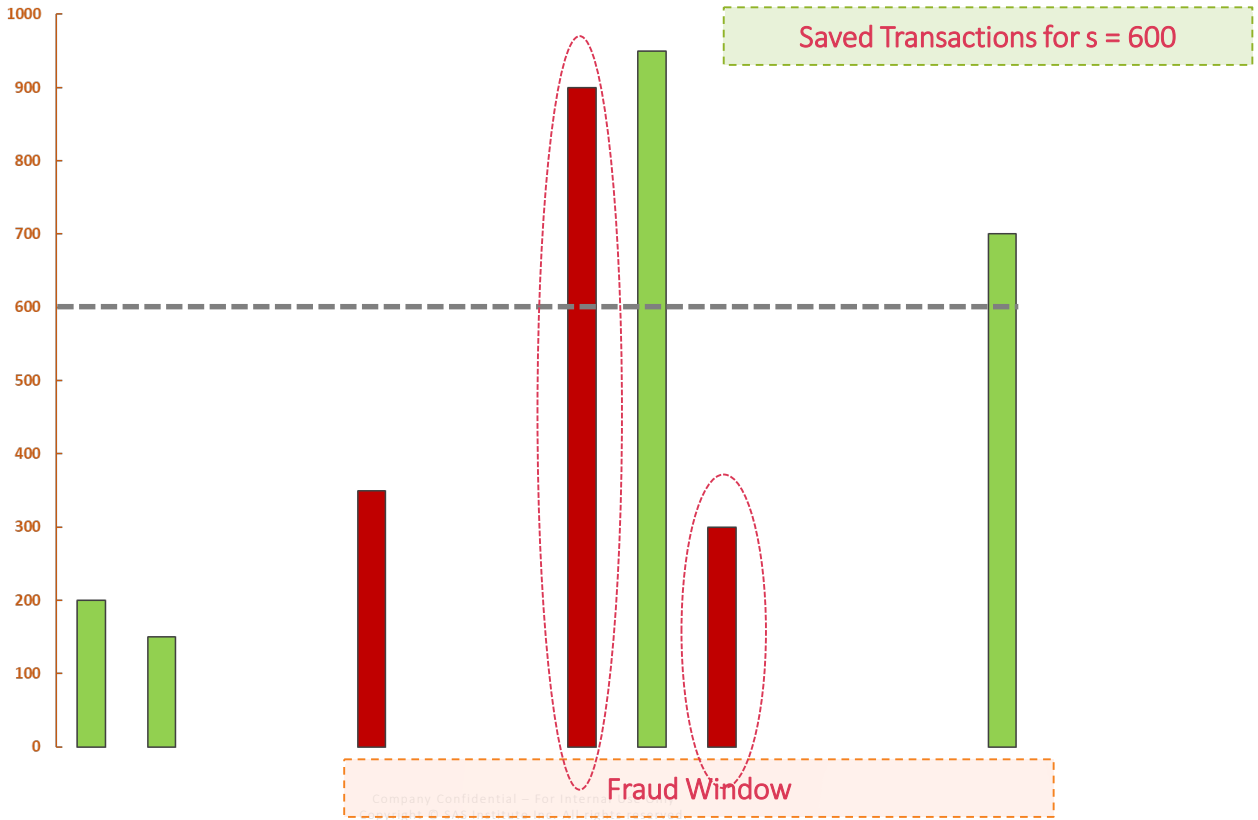
- VDR measures the actual monetary amount of saved fraud losses
- Only the amounts associated with fraud transactions that occur after the case is detected are considered saved.

- $$VDR\ (s) = \frac{Sum\ of\ detected\ fraud\ transactions\ when\ case\ is\ detected\ at\ score \geq s}{Total\ potential\ fraud\ loss}$$

- Fraud transactions that were declined prior to being scored by the model are typically excluded from these calculations

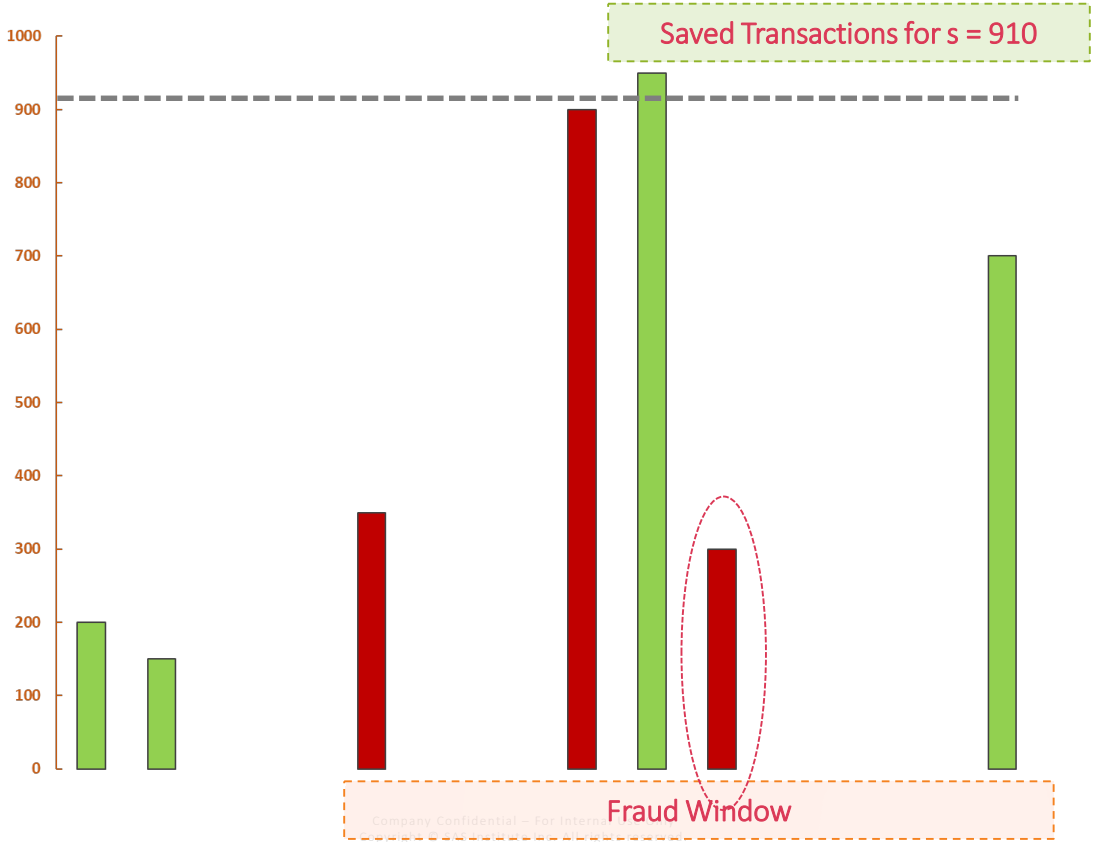
Performance Evaluations

Value Detection Rate - Illustration



Performance Evaluations

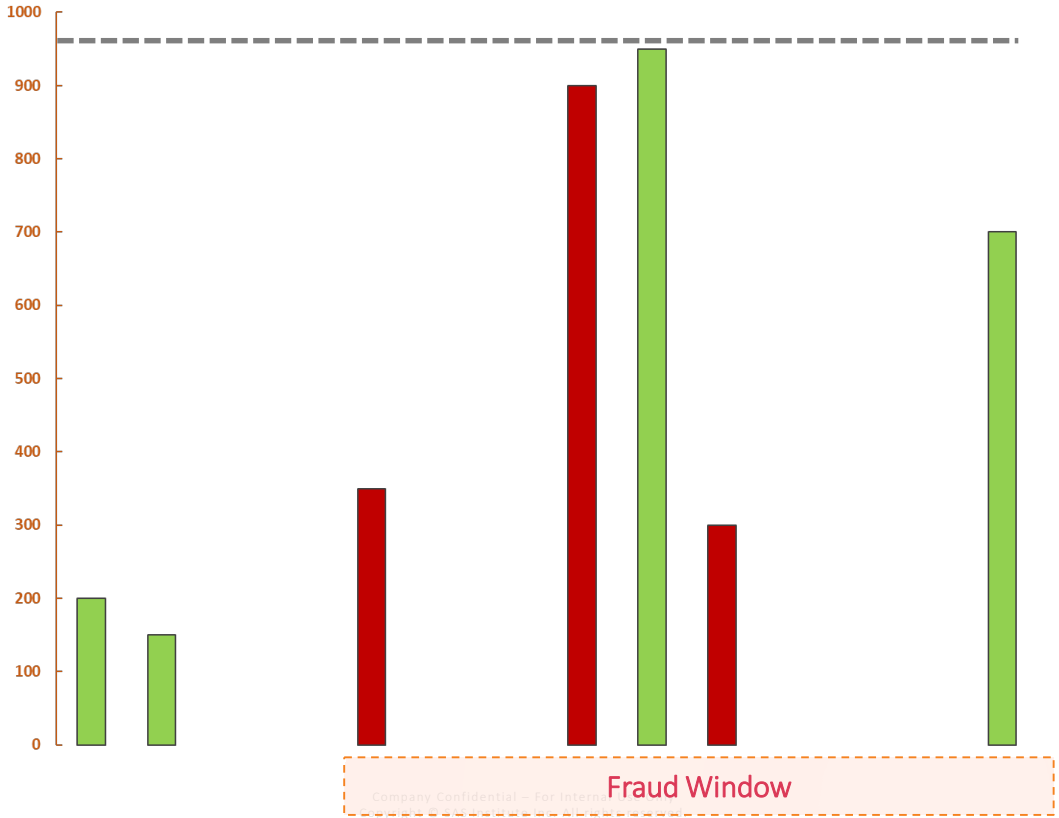
Value Detection Rate - Illustration



Performance Evaluations

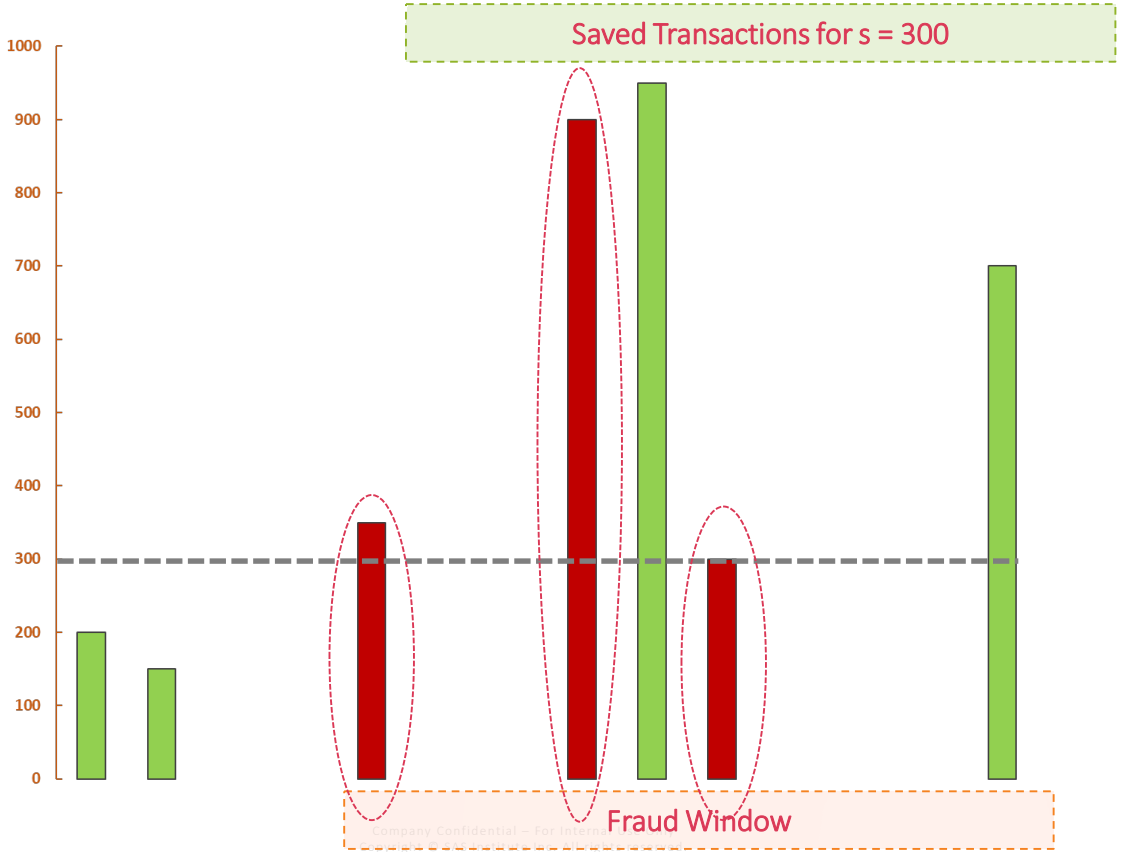
Value Detection Rate - Illustration

No saved transactions at s = 950



Performance Evaluations

Value Detection Rate - Illustration



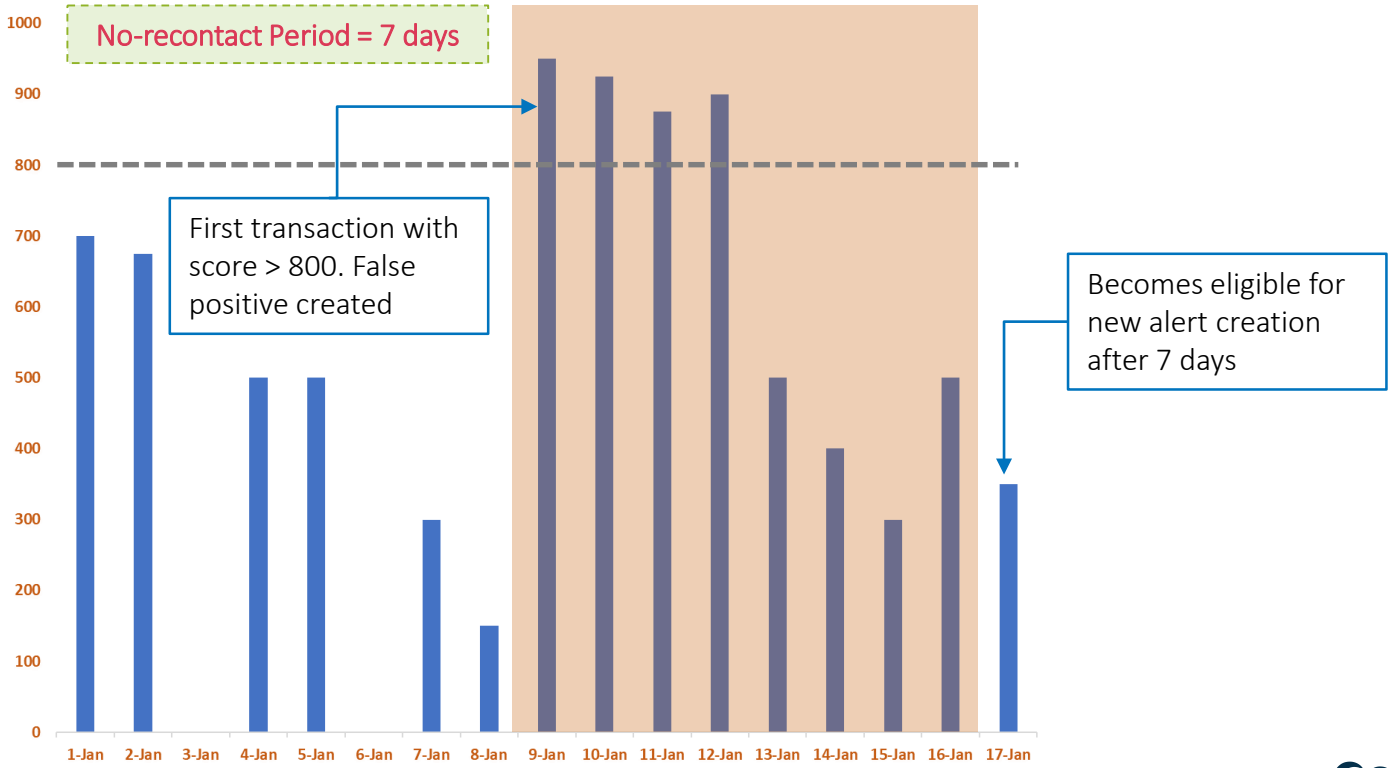
Performance Evaluations

False Positives and No – Recontact Period

- A false positive at score s is triggered when a non-fraudulent transaction score greater than s.
- Typically a cluster of non-fraudulent transactions may score above the evaluation threshold
- In practice however these are all grouped into a single alert
- Also for operational and relationship reasons, a customer may not be contacted for a certain period of time when they have experienced a recent false positive
- The ‘quiet’ period after which a customer is not contacted after a false positive is termed the no – recontact period
- This should be factored in when counting the number of false positives during evaluations.

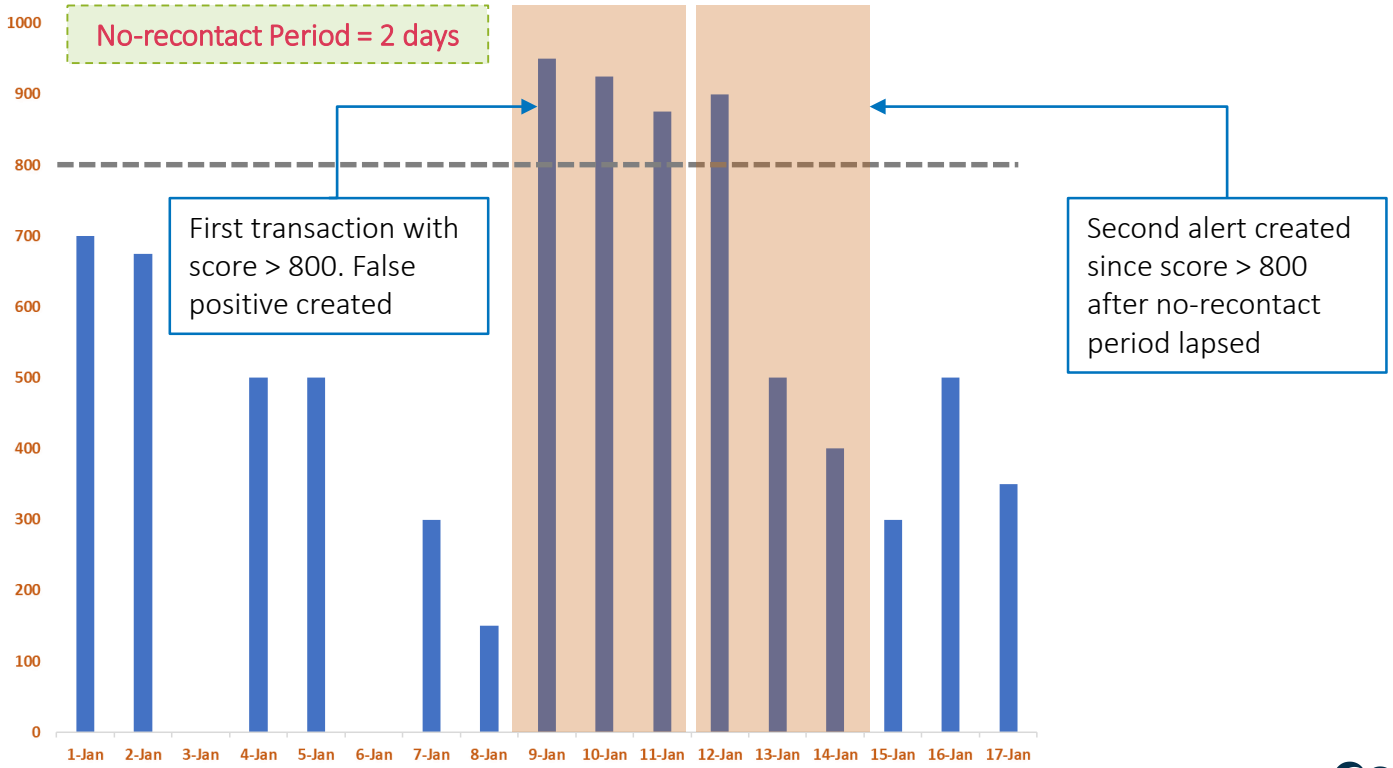
Performance Evaluations

No – Recontact Period Illustration



Performance Evaluations

No – Recontact Period Illustration



Performance Evaluations

Case False Positive Ratio (CFPR)

- A common metric to quantify false positives is the measure of how many false positives the bank needs to create for each detected fraud case

- $$CFPR\ (s) = \frac{No.\ of\ false\ positive\ cases\ created\ at\ score \geq s}{No.of\ fraud\ cases\ detected\ at\ score \geq s}$$

- Range of interest for high fraud rate problems would be between 3:1 to 20:1 (e.g. cards)
- Range of interest for low fraud rate problems could be anywhere between 50:1 to 200:1 (e.g. payments, online banking, deposits)

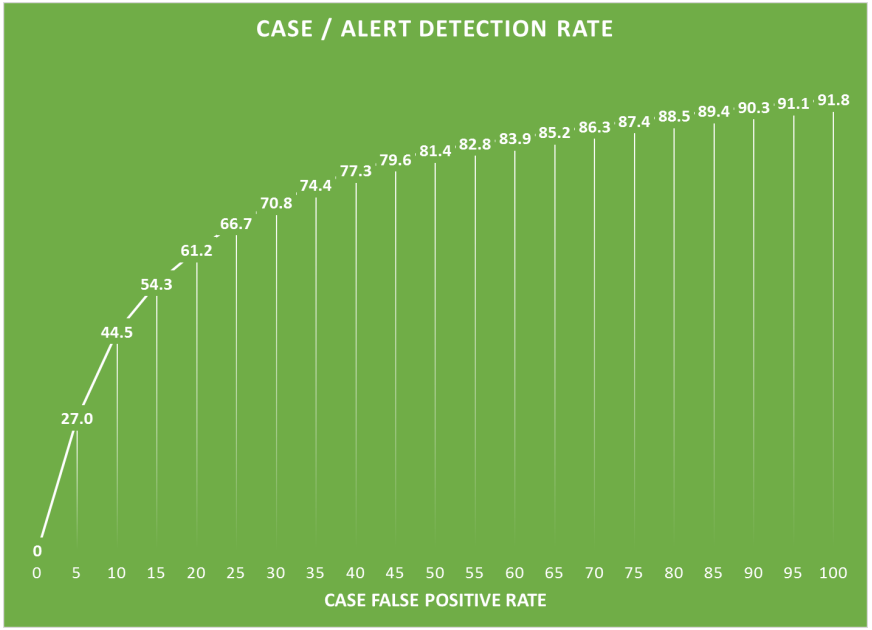
Performance Evaluations

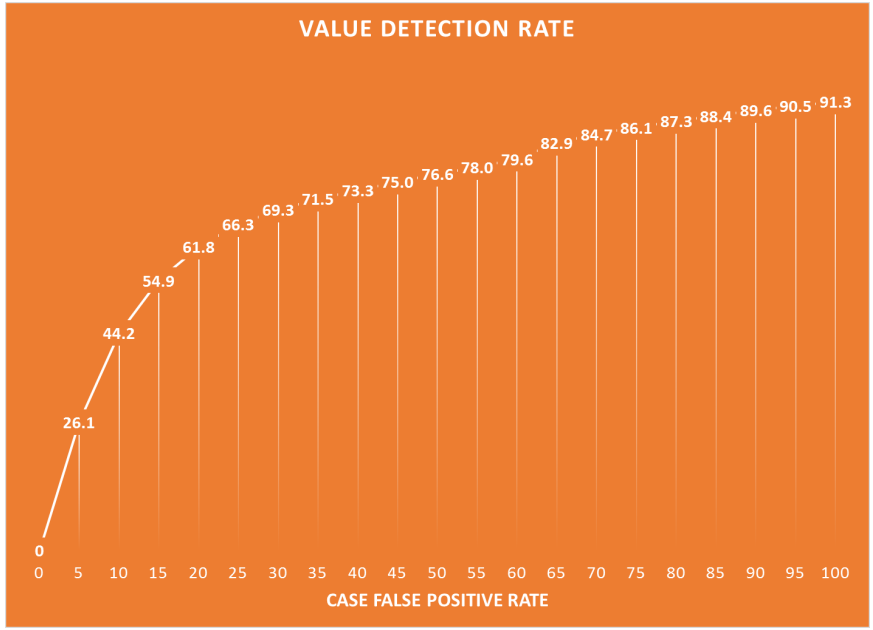
Outsort Volume and Outsort Rate

- Outsort volume for a given period is the number of alerts to be worked during that period
- $$Daily\ Outsort\ (s) = \frac{FPs + Detected\ Cases\ at\ score \geq s}{Days\ in\ the\ evaluation\ period}$$
- $$Daily\ Outsort\ Rate\ (s) = \frac{Daily\ Outsort\ (s)}{Daily\ Outsort\ (1)}$$
- Outsort can be defined for different periods (week, month, quarter etc.)
- Outsort is a better measure in terms of quantifying operational impact since it directly relates the number of alerts to be worked during a given period

Performance Evaluations

Case Level RoCs

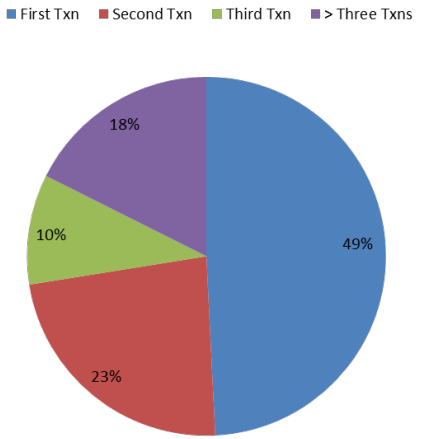




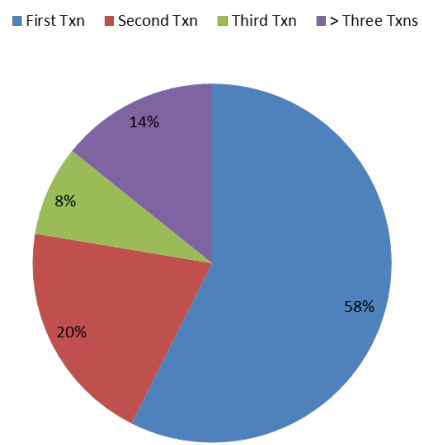
Performance Evaluations

Speed to Detection

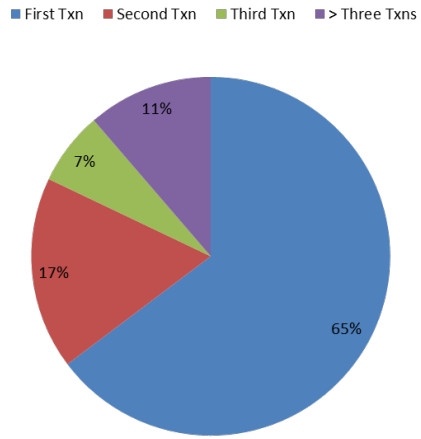
- Measure of how fast cases are detected when they are detected
  - Will have an impact on the value of VDR



CFPR = 5:1
CDR = 49.2%



CFPR = 10:1
CDR = 57.6%



CFPR = 20:1
CDR = 65.4%

Performance Evaluations

Transaction vs Case Level Evaluations

Performance Evaluations

Transaction vs Case Level Evaluations

- The fraud problem typically dictates the evaluation methodology
- Cards are by far the ideal candidates for case level evaluations
  - Cards are typically closed when there is fraud; a clean fraud window can be defined which is ideal for case level evaluations
  - However it can be applied where the notion of 'fraud state' is applicable. For e.g. online user IDs, payment fraud where the account is compromised etc.
- Transaction level evaluations are better suited when the notion of 'fraud state' is not applicable
  - E.g. in check fraud, it is individual checks that are fraudulent; the underlying account does not go into a state of fraud
  - Many payment fraud problems related to social engineering

Performance Evaluations

Transaction vs Case Level Evaluations

- If an entity is put back in use after a compromise (e.g. online user ID), it can be re-compromised again
  - Creates some complexities in demarcating fraud boundaries between episodes
  - Even if there was only a single episode, there needs to be a clear way of identifying where the post-block period ends and where legitimate behavior begins again
- This is also a consideration in fraud tagging for model development