Fraud Index Model

Training/Testing Note (revision 3.0)

Cheng Peng

September 10, 2018

Table of Contents

[1. Introduction 2](#_Toc528310462)

[1.1. Motivation 2](#_Toc528310463)

[1.2. Model Taxonomy 3](#_Toc528310464)

[1.3. Origin and History of the Model 3](#_Toc528310465)

[2. Data Formats and Model Structure for the Index Model 4](#_Toc528310466)

[2.1. Data Format and Attributes 4](#_Toc528310467)

[2.2. Model Structure 4](#_Toc528310468)

[3. Data Source 4](#_Toc528310469)

[3. An R Function For Testing and Validation the Index Model 5](#_Toc528310470)

[3.1. R Function - Validation and Testing Index Models 5](#_Toc528310471)

[3.2. Definitions of Indexes 5](#_Toc528310472)

[3.3. Discriminatory Power 5](#_Toc528310473)

[3.4. KPIs 6](#_Toc528310474)

[4. Model Selection 6](#_Toc528310475)

[4.1. Basic Model Structure 6](#_Toc528310476)

[4.2. Separability of Genuine and Fraud Index Distributions 7](#_Toc528310477)

[4.3. The Structure of the Final Candidate Model and Operational KPIs 12](#_Toc528310478)

[5. Performance of Candiated Models 19](#_Toc528310479)

[5.1. Saparability of the Candicate Models 19](#_Toc528310480)

[5.2. Discriminatory Power of the Candicate Models 20](#_Toc528310481)

[5.3. Porformance Curves of the Candicate Models 20](#_Toc528310482)

[5.4. Operation Costs of the Candicate Models 21](#_Toc528310483)

[5.5. Recommended Final Models 25](#_Toc528310484)

[5.6. SP Implementation 26](#_Toc528310485)

[6. Retospective Evaluation of the Final Models 26](#_Toc528310486)

[6.1. Leakage Analysis 26](#_Toc528310487)

[6.2. Lift Analysis 28](#_Toc528310488)

[6.3. Summary of Retrospective Analysis of the Final Models 28](#_Toc528310489)

[7. Some SP Implementation Notes 29](#_Toc528310490)

[7.1. Scenario 1 - Two Time Frames 29](#_Toc528310491)

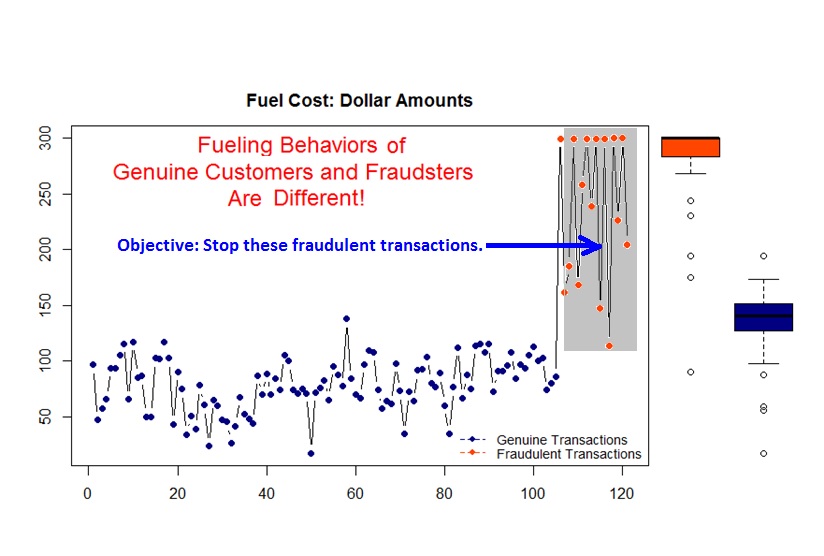
[7.1. Scenario 2 - Single Time Frame 29](#_Toc528310492)

[7.3. The Ddtermination of Decision Boundary 30](#_Toc528310493)

[8. Postface 30](#_Toc528310494)

## 1. Introduction

The objective of this model is to identify the first fraudulent transaction and terminate the compromised card to avoid further fraud loss.



Data structure for the lean Index Model

### 1.1. Motivation

The motivation for developing this index model is to reduce the number of large fraud cases (with more than $3,000 loss) and the average loss per case. We have been looking for patterns in reported large cases since earlier this year. Some obvious patterns such as many fraudulent transactions having a dollar amount near card limits/pump limits were observed. However, many genuine transactions also have similar amounts.

The standard classification models/ algorithms cannot use this pattern to catch fraud. We borrowed the idea of a control chart in monitoring defects in the failure of quality control and developed a purely data-driven control-chart-like algorithm to detect fraudulent transactions. The model is simple and highly scalable.

### 1.2. Model Taxonomy

The model (algorithm) is an unsupervised classification model. There are several hyperparameters involved in the model. The optimization step for identifying the best hyperparameter is based on error metrics such as false positives, precision recalls, and operation cost.

### 1.3. Origin and History of the Model

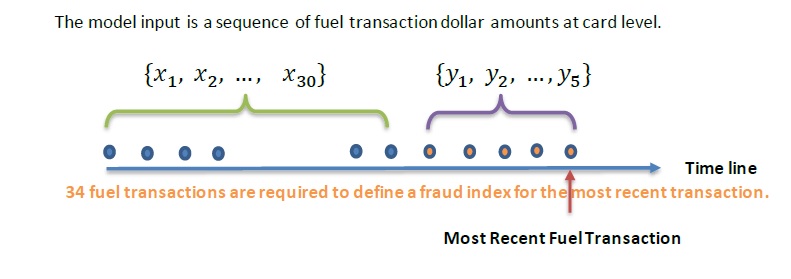
This is a novel model whose idea and formulation are not available in existing literature and software libraries. The model formulation and development went through several stages.

* **Conceptual Model (March 2018)** We consider each fuel card as a machine that produces transactions and a fraudulent transaction to be considered a defective product. A control chart in quality engineering is used to detect defective products whose quality characteristic has a shift from the process mean and/or variance. Since fraudsters don’t know legitimate customer’s spending behavior, the fraudulent transaction will change the pattern of genuine spending. Therefore, we can borrow the logic in the control chart to detect fraud.
* **Proof-of-Concept (April 2018)** The initially proposed version of the index model combines the idea of process capability index (PCI) and control chart. Examples based on real transaction data indicate good performance of the model. Due to technical difficulties in implementation, the physical model has on basic arithmetic operations - we call it crude index model.
* **Initial Crude Model (May-June 2018)** A crude version index model that only involves the four basic operations: addition, subtraction, multiplication, and division and max() and min(). The model was fully validated and tested with good performance in various metrics. A technical challenge in implementing the model is that values from a function cannot be used for another function in SP. The crude version index model contained a square term that is considered a function.
* **Simplified Crude Index Model (July 2018)** A simplified crude index model with four basic operations was built and validated. The modified model has a simple structure like the coefficient of variation. The performance of the modest model is resonantly good for detection.
* **The Final Index Model (August 2018)** Two of the key derived attributes used in the definition of the rolling index require a fixed number of historical transactions to work. However, SP only counts past transactions within a specified time frame. This time frame will cause sequential transactions either disconnected or overlapped. The final model will take two disconnected sub-sequences based on two disjoint time windows.

## 2. Data Formats and Model Structure for the Index Model

The model is defined based on sequence data (consecutive historical transactions) using the idea of a control chart. The structure of the model contains the same major components in PCI and the data structure required for the model is a short serial data.

### 2.1. Data Format and Attributes



Data structure for the lean Index Model

### 2.2. Model Structure

Three hyperparameters are involved in this model: the number of transactions in the definition of a rolling index, the number of standard deviations used in estimating the upper specification limit. The third hyperparameter is the number of transactions used in estimating the USL (we chose 30 for the model in the training phase).

## 3. Data Source

Data used in this validation and testing analysis consists of two subsets. One subset is based on all compromised cards in the last year that have more than 34 legitimate transactions immediately before the fraudulent transaction (add the most recent/first fraudulent transaction to get 35 transactions in the sequence). The other subset consists of 35 most recent legitimate transactions from randomly selected genuine cards from SP. The combined subsets were used for the whole analysis.

The following analytical data sets were prepared using SAS.

* idx.data.wpc contains information based on confirmed WP transactions
* idx.data.gdc contains information based on randomly sampled genuine transactions from the cards sample from SP;
* idx.data.loss contains information from the same source in which idx.data.wpc set was defined. Some additional aggregated information on fraud loss was calculated and included in this data. Three major derived variables: FRAUD\_1ST, Origination, IDX\_SAVE will be used in leakage analysis for the final model(s) at the very end of this note.

For ease of defining rolling indexes, we store transactions from each card in a column in a matrix,

## 3. An R Function for Testing and Validation of the Index Model

We define six different primary indexes and various model KPIs in the R function **idx.model.validation()**.

### 3.1. R Function - Validation and Testing Index Models

Six index models with various tuning hyper-parameters are included in the function. Arguments (some of them are hyper-parameters) to be passed into the function are

* range.s = splitting the half range into this number of intervals
* pc = vector of penalized coefficients (weight) used in the proposed IDX models
* num.roll = number of observations used in rolling idx
* std.num = number of standard deviation used in USL
* wp.cnt = daily average number of WP fraudulent transactions (estimate)
* tot.trx = daily total number of transactions (estimate)
* num.cut = vector of cut-off indexes used in assessing model KPI Table
* low.cut, up.cut = end points of the index interval for calculating the KPI curves
* filename = part of an output file name based on various combinations of hyperparameters.

### 3.2. Definitions of Indexes

We define a Cpk-like index only using several aggregate functions: max, mean, and the four basic arithmetic operations. We also include the penalizing shift of the mean of 30 historical transaction dollar amounts and the that of most recent 5 transactions with different weights. The upper specification limits (USL) were estimated based on the 30 historical transactions using mean + pc\*(max-mean)/2 (where pc is a vector of 5 positive weights tuned by validation).

### 3.3. Discriminatory Power

We use KS curves to evaluate the discriminatory power of the proposed index. Histograms of both indexes associated with genuine and fraudulent transactions were plotted to visualize the discrepancy between the two distributions.

**A side note:** A better measure for assessing the performance of a classification model is a separability metric which tells how good a model can separate the bad outcomes from the good ones.

### 3.4. KPIs

Several types of KPIs were calculated and reported in the R function for model validation and testing: Model development KPIs and model implementation KPIs.

* **Model Development KPIs** These KPIs were based on sensitivity and specificity (using the outcome data): ROC curve, FPR, FNR, TNR(specificity), and TPR (sensitivity or recall).
* **Model implementation KPIs** Since the validation sample is NOT a population, it is typical case-control-like data: under-sampling of good (valid) cards and ALL valid compromised cards. The precision of the model is not estimable! We need to know the population-level fraud rate (about 50 out of 1000000). The Bayes rule needs to be used to calculate the precision. It is expected that KPI precision is very low due to the rarity of fraud. However, it is significantly higher than the average number of fraud (i.e., 50/1000000) if the model has predictive power.
* **Operation KPI Metric - False Positives** The data used in the validation and testing of the proposed index models is not population data. The detection rate is not equal to the predictive positives. False positives can be estimated based on the outcome data that is dependent on the cut-off of the index values.
* **Overall KPI Table** The overall KPI tables that include all aforementioned KPIs were also generated by the R function so that they can be used for selecting final models.

## 4. Model Selection

In this section, we tune parameters using various model performance metrics with the R function introduced in the previous section.

### 4.1. Basic Model Structure

The general structure of the proposed index model is given below

where is the specification limit based on the 30 historical transactions before the most recent transactions with

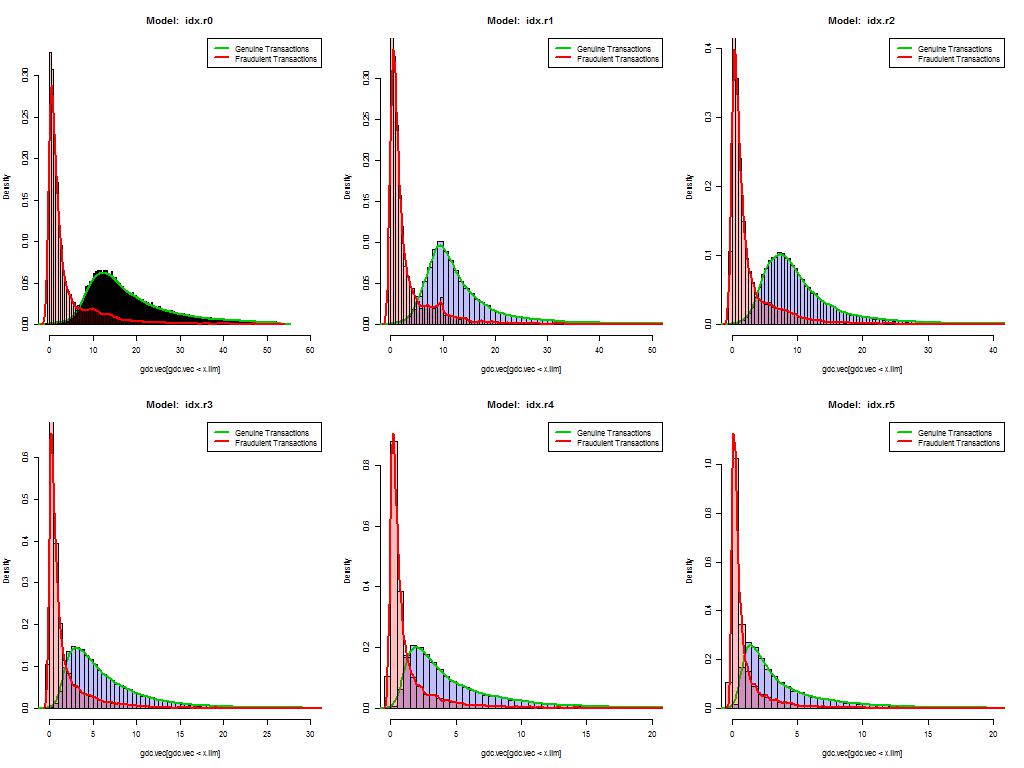
We can see from the above formula that the index is dependent on three hyperparameters:

* number of transactions used in the calculation of the rolling index
* number of estimated standard deviations used in USL
* weight used to penalize the shift of the means from the 30 historical transactions and the most recent 5 transactions.

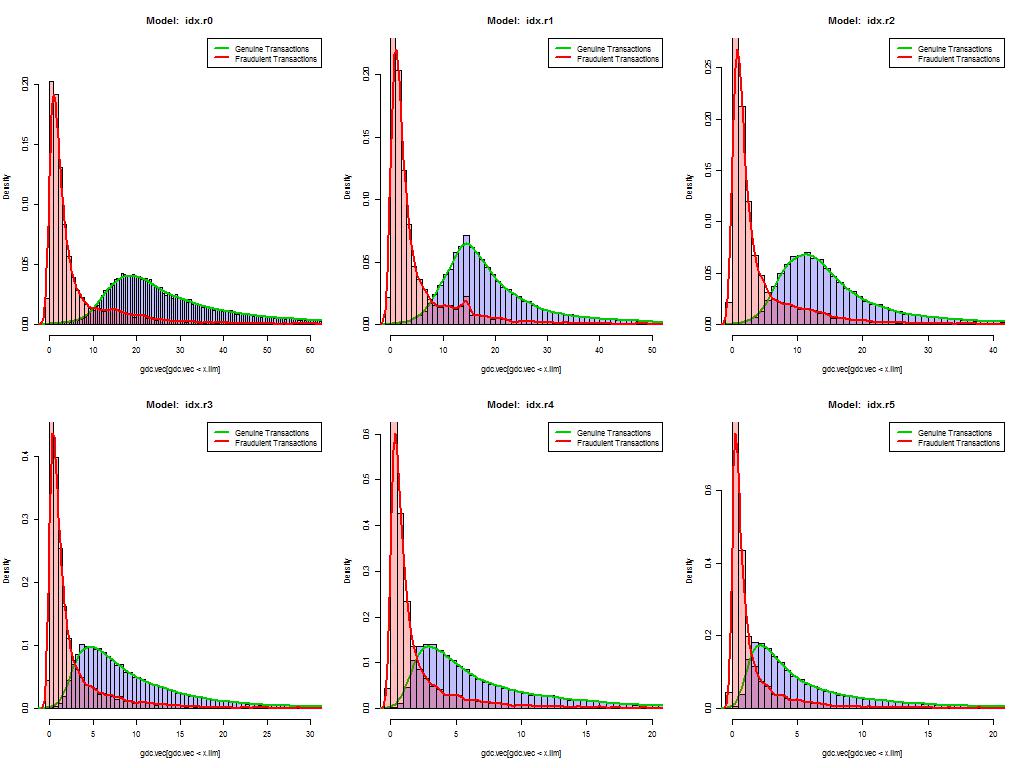
### 4.2. Separability of Genuine and Fraud Index Distributions

Since there is no separability measure of classification models in theory and practice, we present a few plots to visualize the capability of separating fraud from genuine distributions for the proposed models.

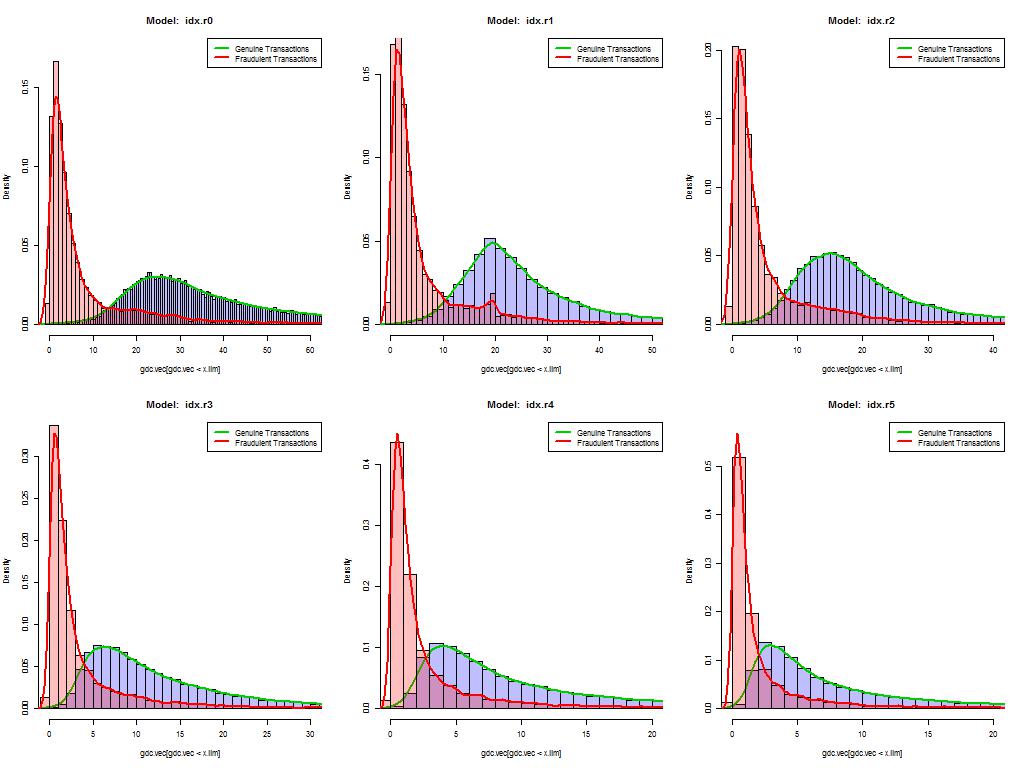
* **Effect of Penalizing Coefficients** We first look at how the penalizing coefficient () impacts the separability of the distributions of genuine and fraud indexes for the given number of standard deviations ().



Case 4.2.1. Histograms of index distributions - rolling index based on 5 observations, USL based on 10 standard deviations



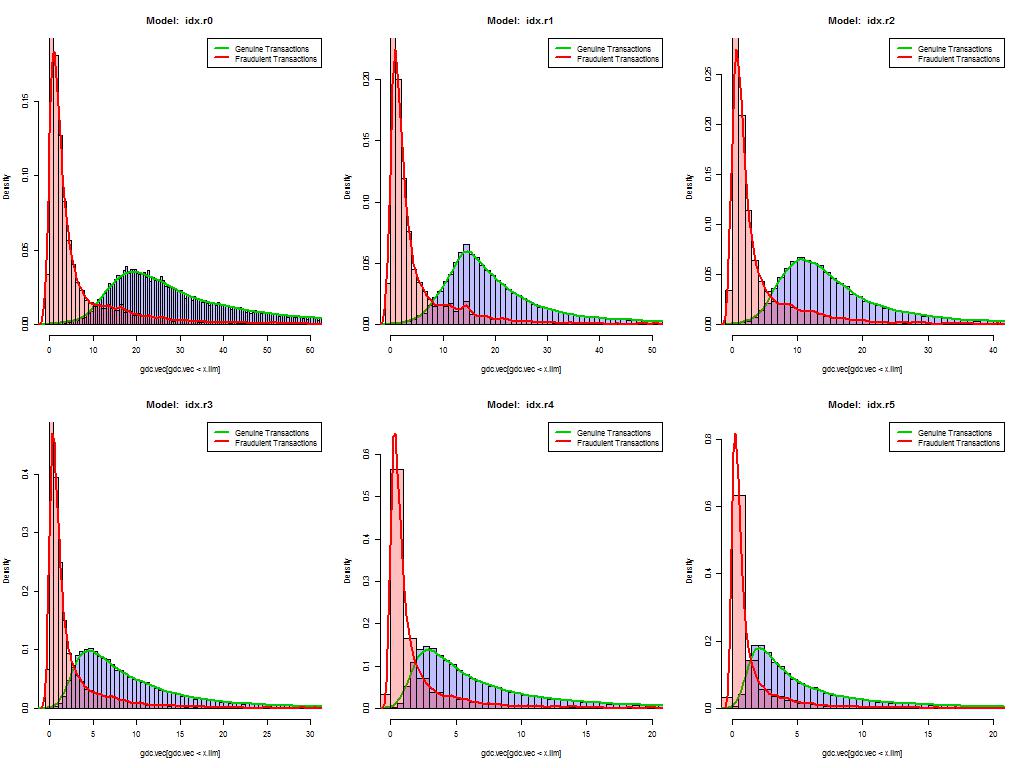
Case 4.2.2. Histograms of index distributions - rolling index based on 5 observations, USL based on 15 standard deviations



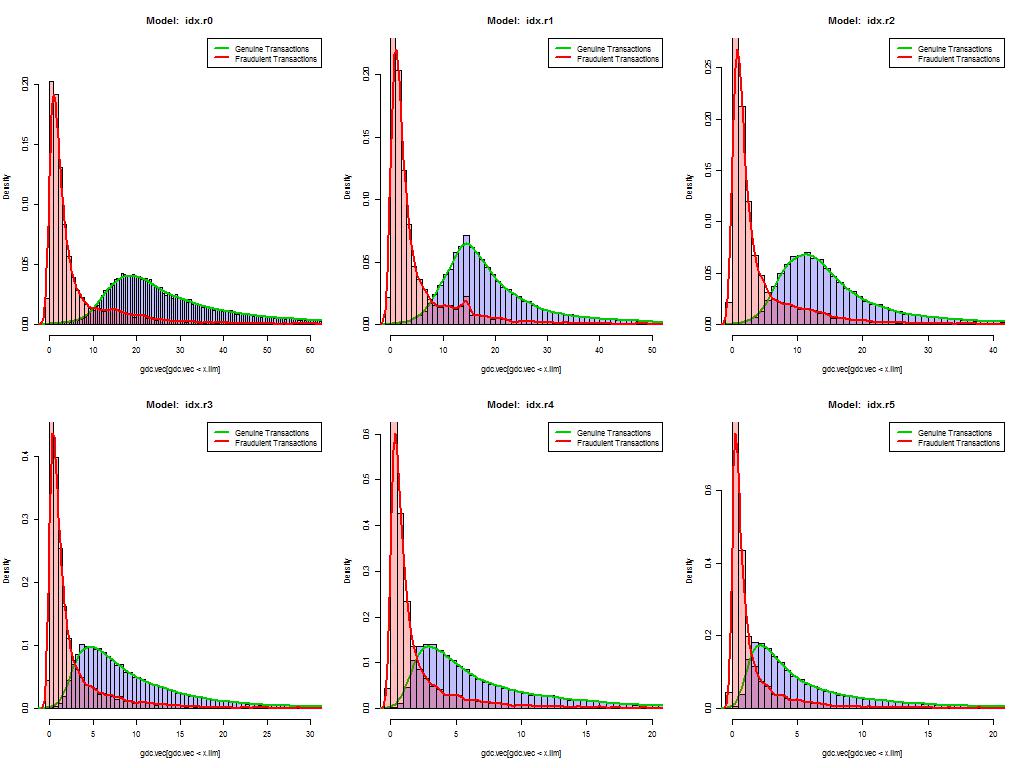
Case 4.2.3. Histograms of index distributions - rolling index based on 5 observations, USL based on 20 standard deviations

We can see that as increases, the two distributions become more difficult to separate. It turns out that the index with no penalized term is easier to separate from each other.

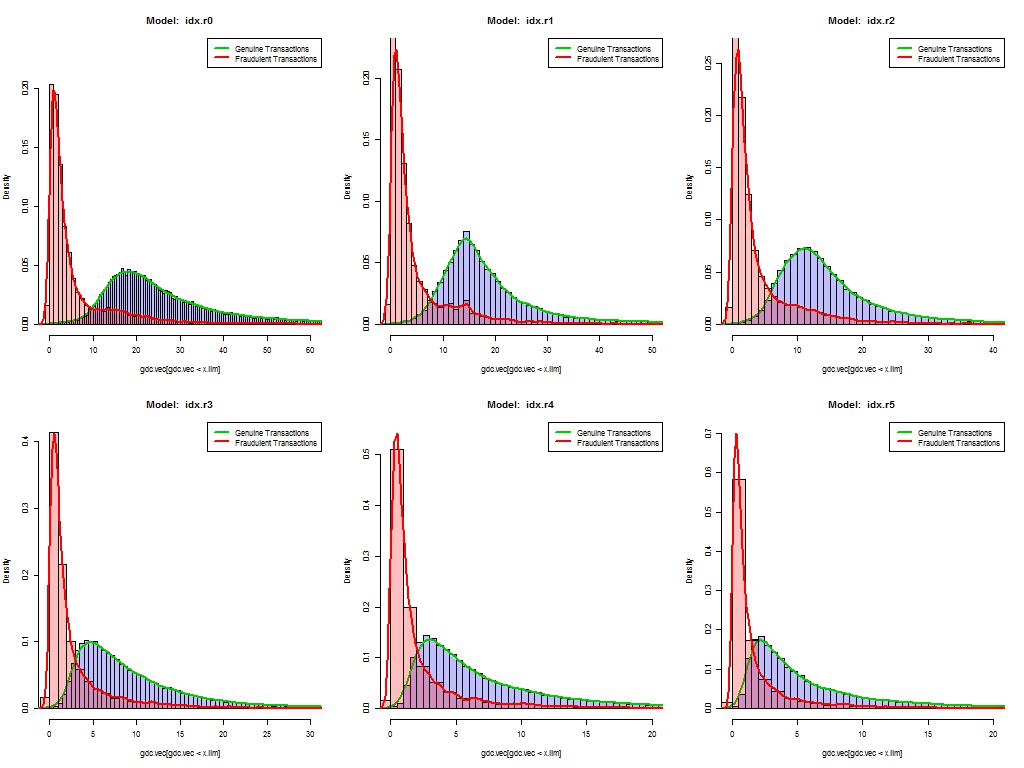
* **Effect of the Number of Standard Deviations ()** We next look at the number of standard deviations used in USL () for the given number of observations used in the rolling index ().



Case 4.2.1. Histograms of index distributions - rolling index based on 4 observations, USL based on 15 standard deviations



Case 4.2.2. Histograms of index distributions - rolling index based on 5 observations, USL based on 15 standard deviations



Case 4.2.3. Histograms of index distributions - rolling index based on 7 observations, USL based on 15 standard deviations

It seems that the number of transactions in the rolling index does not have a significant impact on the separability of the distributions of fraud and genuine indexes. However, it may impact the sensitivity of the index. We will investigate this effect later.

Since the KS curve measures the discrepancy between two distributions but not the separability (although it is related to the separability), We will not present the KS statistics and the KS curves at this point. However, KS statistics (curves) will be reported in the analysis of the final model(s).

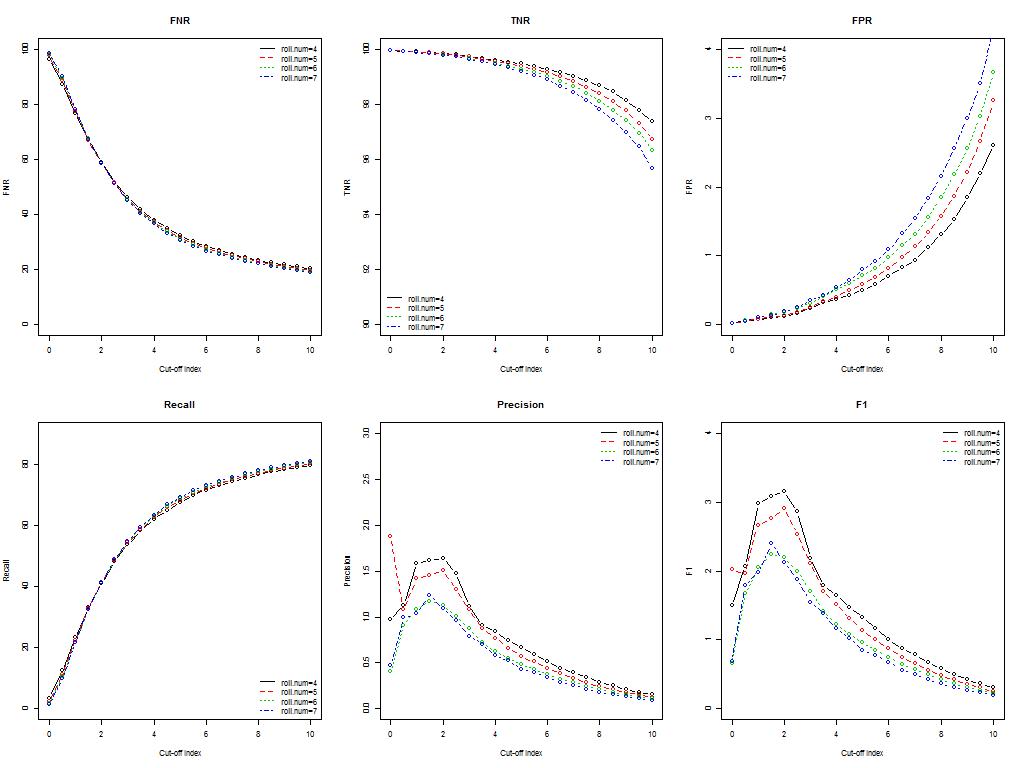
### 4.3. The Structure of the Final Candidate Model and Operational KPIs

We will now focus on the model with the following general form (with no penalizing term)

Unlike the previously proposed index model, this model could generate negative index values. We still use the linear decision boundary for this model. The smaller the index, the more likely to be fraudulent.

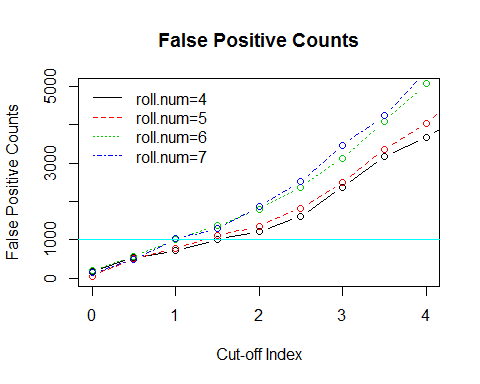
#### 4.3.1. Effect of the Number of Transactions Used Index on KPIs

KPI curves based on the proposed model with various but fixed .



Case 4.3.1.1. KPIs for candidate models - impact on the KPI with various number of observations used in the definition of rolling index

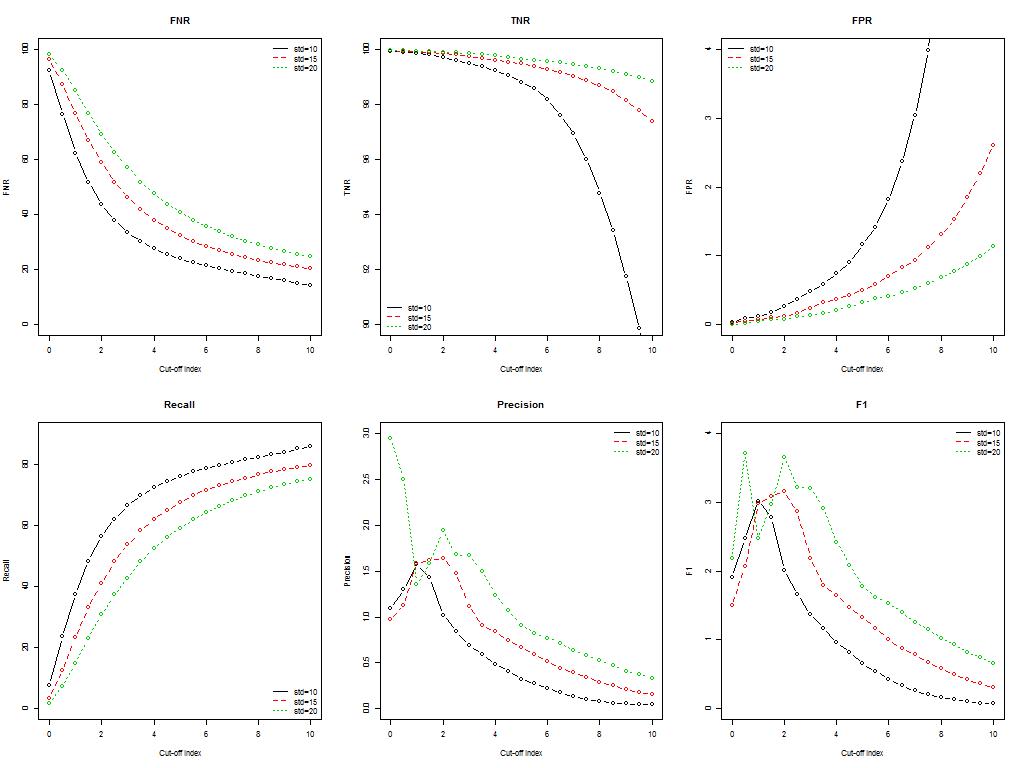
For the fixed , it seems that rolling index models with 4 and 5 observations are better than the ones using more than 5 observations. Next, we look at the false positive counts generated from proposed models with different cut-off index values.



The operational PKI - false positive counts - also support the above claim. In the next subsection, we examine whether the number of standard deviations used in the index model will affect the KPIS.

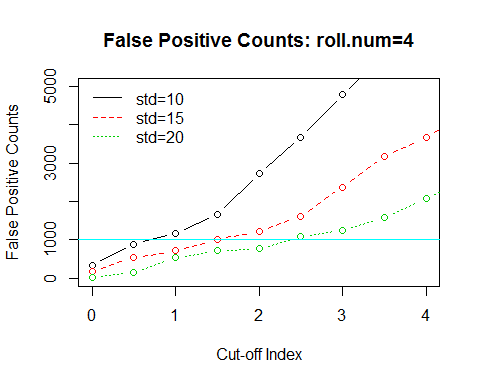
#### 4.3.2. Effect of the Number of Standard Deviations Used Index on KPIs

KPI curves based on the proposed model with various but fixed .

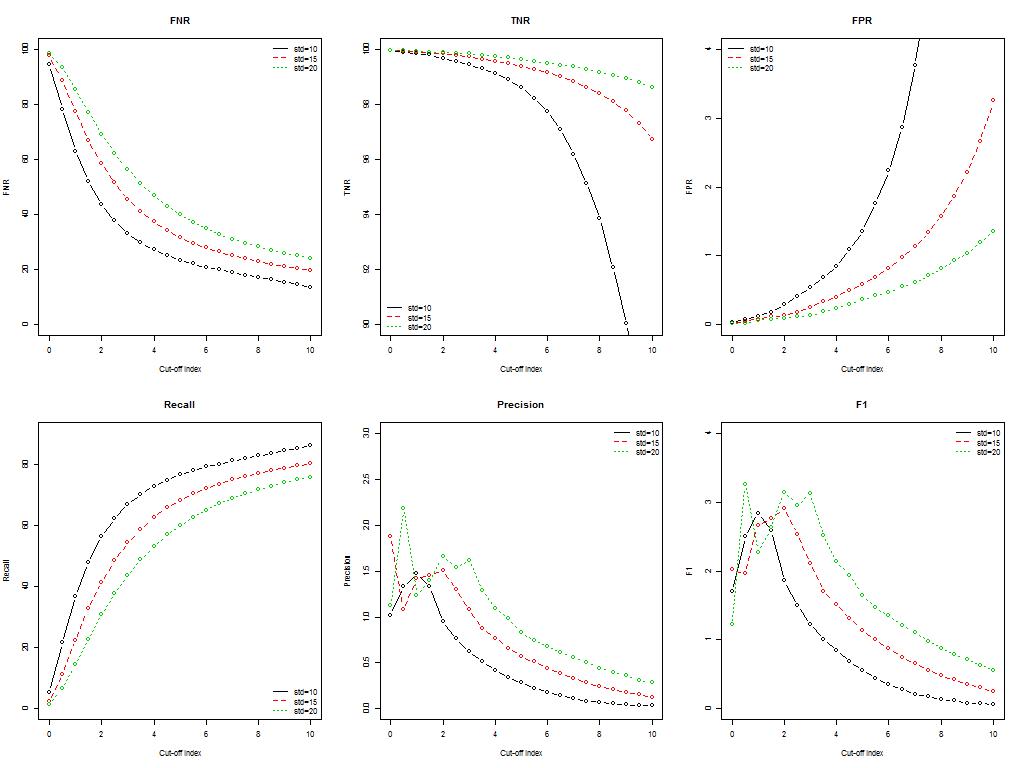


Case 4.3.2.1. KPIs of the candidate model - impact on the index with various numbers of standard deviation used in the definition in the index (number of obs in the rolling index = 4)

The curve of false positive counts vs. cut-off index based on 4-obs rolling index.

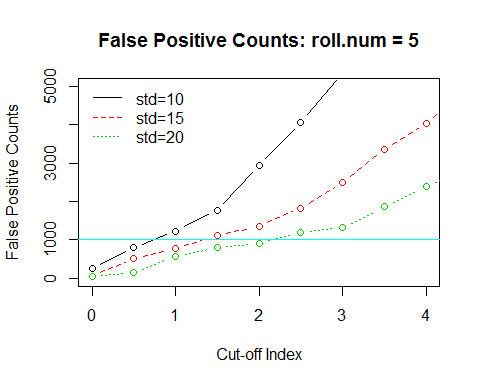


KPI curves based on the proposed model with various but fixed .



Case 4.3.2.1. KPIs of the candidate model - impact on the index with various numbers of standard deviation used in the definition in the index (number of obs in the rolling index = 5)

The curve of false positive counts vs. cut-off index based on 5-obs rolling index.



#### 4.3.3. Candidate Models

Based on the results in the previous subsections, we choose the following four index models for further evaluation. Recall the general form of the final candidate family of models

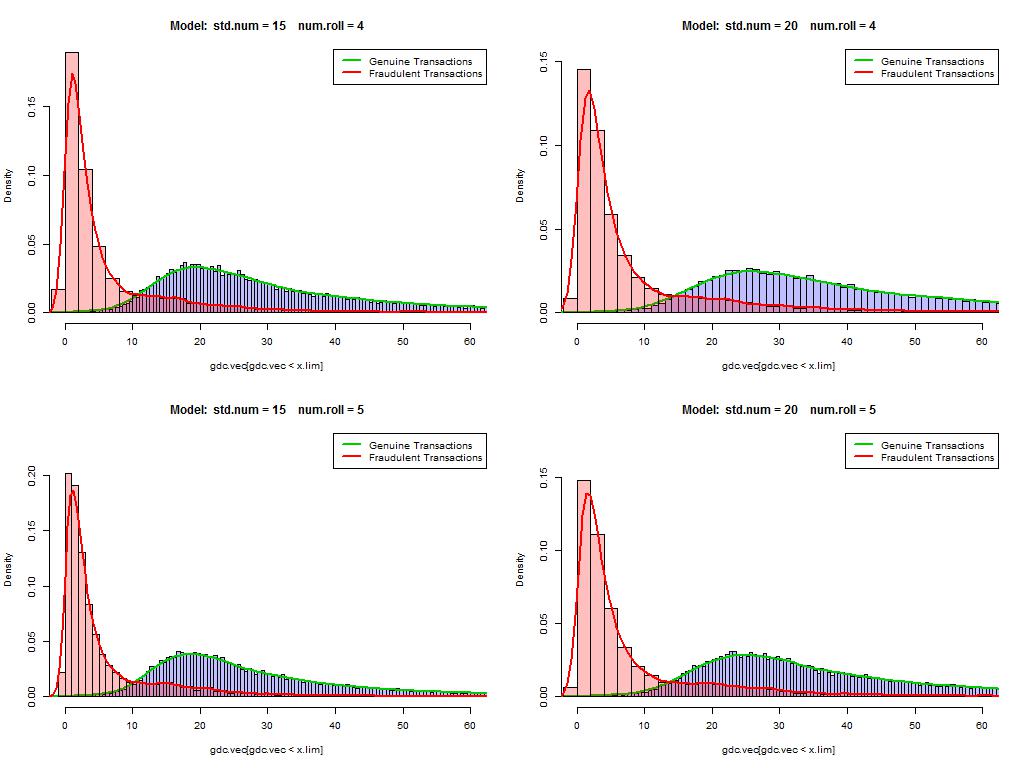
* Model 1. and
* Model 2. and
* Model 3. and
* Model 4. and :

## 5. Performance of Candidate Models

We reduced our initially proposed testing models from 60 to the above 4 candidate models. We name these models by including the values of the hyperparameters: idx4.15, idx4.20, idx5.15, and idx5.20, respectively.

### 5.1. Separability of the Candidate Models

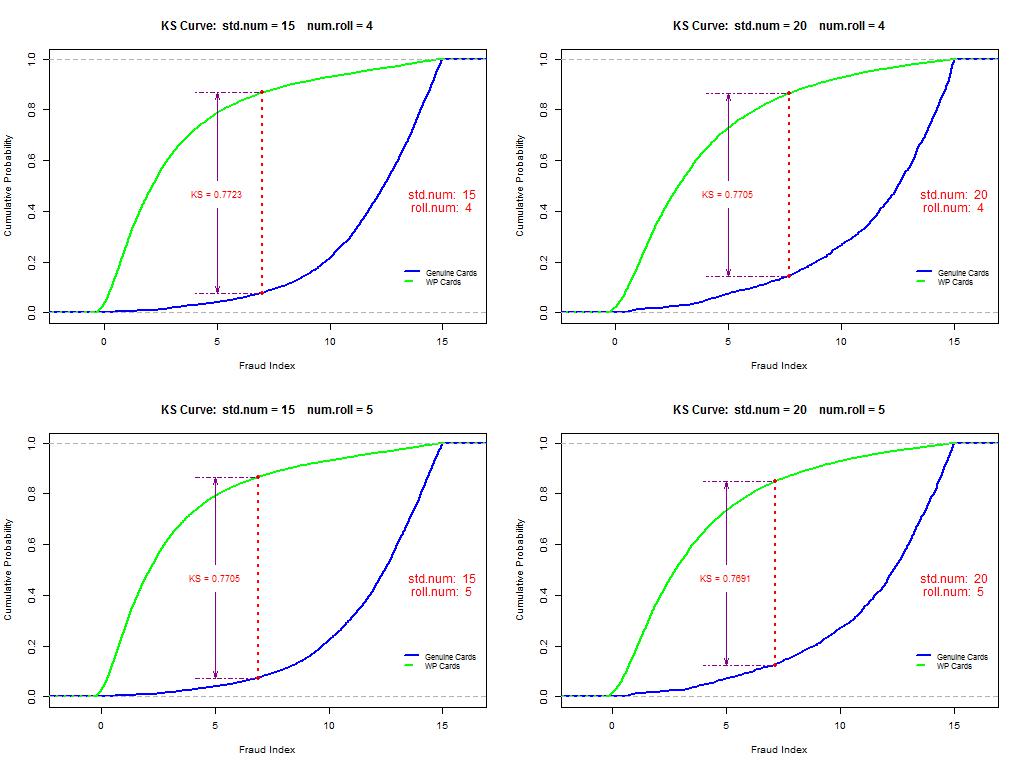
Visualization of the separability of the candidate models



Case 5.1.1. Histograms of index distributions - Separability of candidate models

All four candidate models have similar separability. Model 3 seems to be slightly better than others.

### 5.2. Discriminatory Power of the Candidate Models

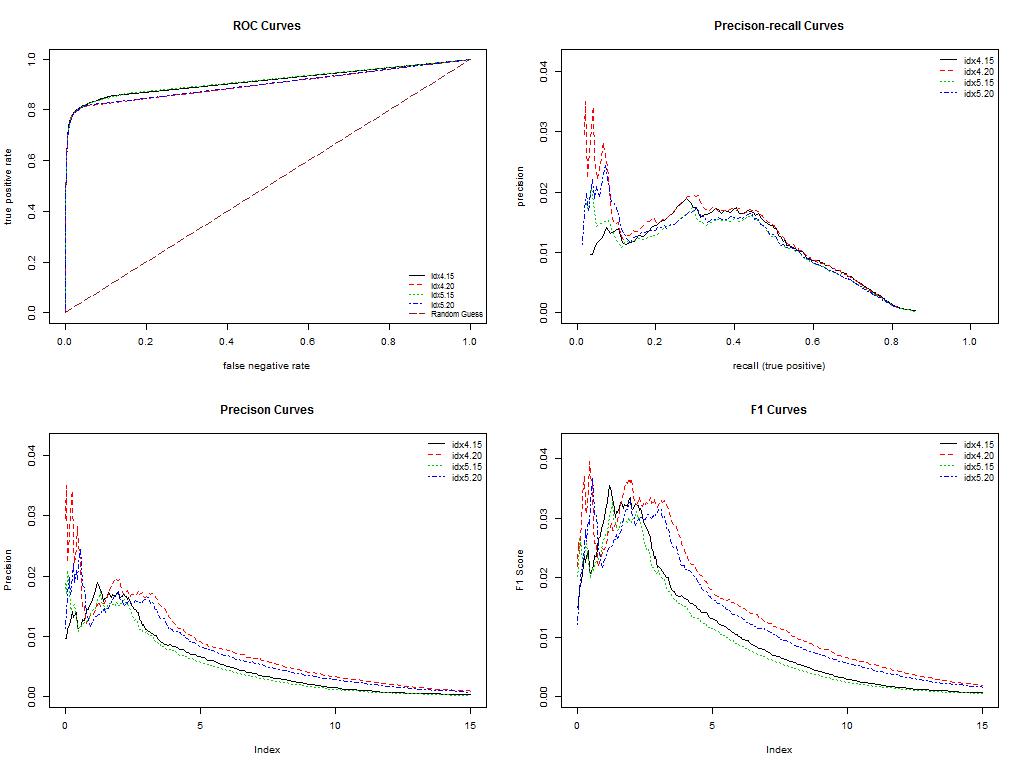


Case 5.2.1. K-S Curves - Discriminatory power of candidate models

There is no significant difference among the four candidate models concerning KS.

### 5.3. Performance Curves of the Candidate Models

We modified the following performance function a bit to avoid null row/column in the confusion matrix (hence avoid potential computing errors due to dimension collapsing in the calculation)



Case 5.3.1. K-S Curves - PKIs of candidate models

It seems that there is a big variation near the practical decision boundary. We will look at the operation cost (false positives) in the following

### 5.4. Operation Costs of the Candidate Models

Next, we present several inline performance tables:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| cut.off | perf.r1.FNR | perf.r1.TNR | perf.r1.FPR | perf.r1.Recall | perf.r1.Precision | perf.r1.F1 | perf.r1.False.Positives |
| 0.0 | 96.62 | 99.98 | 0.02 | 3.36 | 0.97 | 1.50 | 171.7869 |
| 0.5 | 96.07 | 99.98 | 0.02 | 3.92 | 0.97 | 1.55 | 543.9918 |
| 1.0 | 95.44 | 99.98 | 0.02 | 4.55 | 1.12 | 1.80 | 715.7786 |
| 1.5 | 94.54 | 99.97 | 0.02 | 5.45 | 1.18 | 1.93 | 1002.0901 |
| 2.0 | 93.53 | 99.97 | 0.03 | 6.45 | 1.24 | 2.08 | 1231.1392 |
| 2.5 | 92.57 | 99.97 | 0.03 | 7.42 | 1.42 | 2.38 | 1603.3441 |
| 3.0 | 91.65 | 99.97 | 0.03 | 8.34 | 1.31 | 2.26 | 2376.3850 |
| 3.5 | 90.63 | 99.96 | 0.03 | 9.35 | 1.34 | 2.35 | 3178.0571 |
| 4.0 | 89.51 | 99.96 | 0.04 | 10.47 | 1.39 | 2.45 | 3664.7866 |
| 4.5 | 88.51 | 99.95 | 0.05 | 11.48 | 1.17 | 2.12 | 4323.3029 |
| 5.0 | 87.54 | 99.94 | 0.05 | 12.45 | 1.13 | 2.07 | 5039.0815 |
| 5.5 | 86.52 | 99.94 | 0.06 | 13.46 | 1.16 | 2.14 | 5869.3847 |
| 6.0 | 85.49 | 99.94 | 0.06 | 14.49 | 1.19 | 2.20 | 7014.6305 |
| 6.5 | 84.37 | 99.94 | 0.06 | 15.61 | 1.28 | 2.37 | 8303.0320 |
| 7.0 | 83.15 | 99.93 | 0.07 | 16.84 | 1.26 | 2.35 | 9448.2778 |
| 7.5 | 81.96 | 99.93 | 0.07 | 18.02 | 1.35 | 2.51 | 11223.4088 |
| 8.0 | 80.92 | 99.93 | 0.07 | 19.07 | 1.43 | 2.66 | 13055.8021 |
| 8.5 | 79.79 | 99.93 | 0.07 | 20.20 | 1.45 | 2.70 | 15317.6626 |
| 9.0 | 78.74 | 99.93 | 0.07 | 21.24 | 1.52 | 2.84 | 18467.0885 |
| 9.5 | 77.78 | 99.93 | 0.07 | 22.21 | 1.53 | 2.86 | 22017.3505 |
| 10.0 | 76.69 | 99.93 | 0.07 | 23.29 | 1.60 | 3.00 | 26140.2353 |
| 10.5 | 75.65 | 99.93 | 0.07 | 24.34 | 1.67 | 3.13 | 31923.7266 |
| 11.0 | 74.41 | 99.93 | 0.07 | 25.57 | 1.76 | 3.28 | 37535.4310 |
| 11.5 | 73.37 | 99.93 | 0.07 | 26.62 | 1.83 | 3.42 | 45466.2582 |
| 12.0 | 72.28 | 99.93 | 0.07 | 27.71 | 1.90 | 3.55 | 53712.0279 |
| 12.5 | 71.22 | 99.92 | 0.08 | 28.77 | 1.83 | 3.44 | 62673.5763 |
| 13.0 | 70.38 | 99.91 | 0.08 | 29.61 | 1.75 | 3.31 | 72436.7967 |
| 13.5 | 69.63 | 99.91 | 0.09 | 30.35 | 1.74 | 3.28 | 83402.5253 |
| 14.0 | 68.71 | 99.90 | 0.10 | 31.28 | 1.58 | 3.01 | 96429.6962 |
| 14.5 | 67.68 | 99.90 | 0.10 | 32.31 | 1.63 | 3.11 | 108798.3508 |
| 15.0 | 66.85 | 99.90 | 0.10 | 33.14 | 1.63 | 3.10 | 121739.6284 |

Performance table of model 2

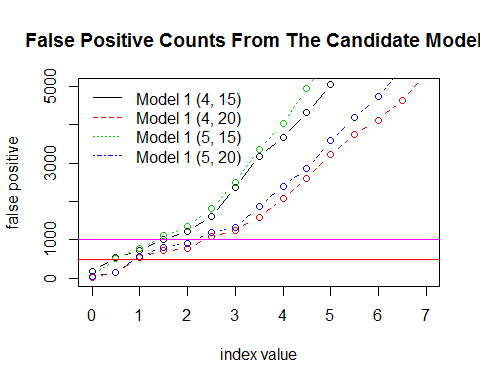
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| cut.off | perf.r2.FNR | perf.r2.TNR | perf.r2.FPR | perf.r2.Recall | perf.r2.Precision | perf.r2.F1 | perf.r2.False.Positives |
| 0.0 | 98.25 | 99.99 | 0.00 | 1.74 | 2.95 | 2.19 | 28.63115 |
| 0.5 | 97.91 | 99.99 | 0.00 | 2.07 | 3.50 | 2.60 | 143.15572 |
| 1.0 | 97.37 | 99.99 | 0.01 | 2.62 | 2.23 | 2.41 | 543.99175 |
| 1.5 | 96.87 | 99.99 | 0.01 | 3.12 | 2.65 | 2.87 | 715.77862 |
| 2.0 | 96.34 | 99.99 | 0.01 | 3.65 | 3.09 | 3.34 | 773.04091 |
| 2.5 | 95.94 | 99.99 | 0.01 | 4.05 | 3.41 | 3.70 | 1087.98351 |
| 3.0 | 95.52 | 99.99 | 0.01 | 4.47 | 2.54 | 3.24 | 1259.77038 |
| 3.5 | 94.85 | 99.99 | 0.01 | 5.14 | 2.20 | 3.08 | 1574.71297 |
| 4.0 | 94.16 | 99.99 | 0.01 | 5.82 | 2.48 | 3.48 | 2090.07358 |
| 4.5 | 93.35 | 99.99 | 0.01 | 6.63 | 2.82 | 3.95 | 2605.43419 |
| 5.0 | 92.63 | 99.98 | 0.01 | 7.36 | 2.51 | 3.74 | 3235.31938 |
| 5.5 | 92.05 | 99.98 | 0.02 | 7.94 | 2.26 | 3.52 | 3750.67999 |
| 6.0 | 91.17 | 99.97 | 0.03 | 8.81 | 1.68 | 2.82 | 4122.88487 |
| 6.5 | 90.51 | 99.97 | 0.03 | 9.48 | 1.48 | 2.57 | 4638.24548 |
| 7.0 | 89.60 | 99.96 | 0.03 | 10.38 | 1.49 | 2.60 | 5325.39296 |
| 7.5 | 88.83 | 99.95 | 0.04 | 11.16 | 1.28 | 2.30 | 5955.27815 |
| 8.0 | 88.11 | 99.95 | 0.05 | 11.88 | 1.21 | 2.19 | 6842.84365 |
| 8.5 | 87.41 | 99.95 | 0.05 | 12.57 | 1.28 | 2.32 | 7701.77799 |
| 9.0 | 86.67 | 99.95 | 0.05 | 13.32 | 1.28 | 2.33 | 8847.02379 |
| 9.5 | 85.75 | 99.95 | 0.05 | 14.24 | 1.36 | 2.49 | 9935.00730 |
| 10.0 | 85.09 | 99.94 | 0.05 | 14.89 | 1.35 | 2.48 | 11395.19569 |
| 10.5 | 84.15 | 99.94 | 0.06 | 15.83 | 1.36 | 2.51 | 12683.59722 |
| 11.0 | 83.33 | 99.94 | 0.06 | 16.66 | 1.43 | 2.64 | 14172.41675 |
| 11.5 | 82.43 | 99.94 | 0.06 | 17.56 | 1.51 | 2.78 | 16205.22805 |
| 12.0 | 81.53 | 99.94 | 0.06 | 18.46 | 1.51 | 2.80 | 18638.87537 |
| 12.5 | 80.73 | 99.94 | 0.06 | 19.26 | 1.58 | 2.91 | 21444.72757 |
| 13.0 | 79.95 | 99.93 | 0.07 | 20.03 | 1.50 | 2.79 | 24450.99780 |
| 13.5 | 79.12 | 99.93 | 0.07 | 20.87 | 1.50 | 2.79 | 28459.35809 |
| 14.0 | 78.37 | 99.93 | 0.07 | 21.62 | 1.55 | 2.89 | 32410.45609 |
| 14.5 | 77.63 | 99.93 | 0.07 | 22.35 | 1.54 | 2.88 | 36905.54585 |
| 15.0 | 76.78 | 99.93 | 0.07 | 23.20 | 1.60 | 2.98 | 42717.66828 |

Performance table for model 3.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| cut.off | perf.r3.FNR | perf.r3.TNR | perf.r3.FPR | perf.r3.Recall | perf.r3.Precision | perf.r3.F1 | perf.r3.False.Positives |
| 0.0 | 97.80 | 99.99 | 0.01 | 2.19 | 1.88 | 2.02 | 57.26229 |
| 0.5 | 97.04 | 99.99 | 0.01 | 2.95 | 1.69 | 2.15 | 515.36061 |
| 1.0 | 96.30 | 99.99 | 0.01 | 3.68 | 2.10 | 2.68 | 773.04091 |
| 1.5 | 95.75 | 99.99 | 0.01 | 4.24 | 1.82 | 2.54 | 1116.61465 |
| 2.0 | 95.01 | 99.98 | 0.02 | 4.97 | 1.43 | 2.22 | 1345.66381 |
| 2.5 | 93.98 | 99.98 | 0.02 | 6.00 | 1.48 | 2.37 | 1832.39328 |
| 3.0 | 92.93 | 99.97 | 0.02 | 7.06 | 1.52 | 2.50 | 2490.90961 |
| 3.5 | 91.99 | 99.97 | 0.03 | 8.00 | 1.53 | 2.57 | 3349.84396 |
| 4.0 | 90.89 | 99.96 | 0.03 | 9.10 | 1.31 | 2.28 | 4036.99144 |
| 4.5 | 89.69 | 99.95 | 0.04 | 10.29 | 1.18 | 2.12 | 4953.18808 |
| 5.0 | 88.68 | 99.95 | 0.05 | 11.31 | 1.09 | 1.98 | 5926.64701 |
| 5.5 | 87.71 | 99.95 | 0.05 | 12.28 | 1.18 | 2.15 | 6928.73708 |
| 6.0 | 86.59 | 99.94 | 0.06 | 13.40 | 1.16 | 2.13 | 8159.87631 |
| 6.5 | 85.43 | 99.94 | 0.06 | 14.56 | 1.20 | 2.21 | 9677.32700 |
| 7.0 | 84.32 | 99.93 | 0.06 | 15.67 | 1.23 | 2.28 | 11337.93340 |
| 7.5 | 83.10 | 99.93 | 0.07 | 16.89 | 1.21 | 2.27 | 13428.00699 |
| 8.0 | 81.94 | 99.93 | 0.07 | 18.05 | 1.25 | 2.33 | 15833.02316 |
| 8.5 | 80.68 | 99.92 | 0.07 | 19.31 | 1.28 | 2.40 | 18667.50651 |
| 9.0 | 79.58 | 99.92 | 0.08 | 20.41 | 1.30 | 2.45 | 22160.50620 |
| 9.5 | 78.48 | 99.92 | 0.08 | 21.50 | 1.37 | 2.58 | 26741.48939 |
| 10.0 | 77.62 | 99.92 | 0.08 | 22.37 | 1.43 | 2.68 | 32582.24296 |
| 10.5 | 76.51 | 99.92 | 0.08 | 23.47 | 1.44 | 2.72 | 38222.57852 |
| 11.0 | 75.33 | 99.91 | 0.08 | 24.66 | 1.46 | 2.76 | 45323.10247 |
| 11.5 | 74.04 | 99.91 | 0.08 | 25.95 | 1.54 | 2.90 | 52767.20016 |
| 12.0 | 72.83 | 99.91 | 0.08 | 27.16 | 1.61 | 3.04 | 62501.78945 |
| 12.5 | 71.76 | 99.91 | 0.08 | 28.23 | 1.67 | 3.16 | 73954.24743 |
| 13.0 | 70.61 | 99.91 | 0.08 | 29.37 | 1.74 | 3.28 | 87468.14785 |
| 13.5 | 69.78 | 99.90 | 0.09 | 30.21 | 1.57 | 2.99 | 100781.63026 |
| 14.0 | 68.96 | 99.90 | 0.10 | 31.02 | 1.52 | 2.91 | 114610.47327 |
| 14.5 | 68.05 | 99.89 | 0.11 | 31.94 | 1.49 | 2.84 | 130729.80789 |
| 15.0 | 67.03 | 99.89 | 0.11 | 32.96 | 1.45 | 2.79 | 146133.36387 |

Performance table for model 4.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| cut.off | perf.r4.FNR | perf.r4.TNR | perf.r4.FPR | perf.r4.Recall | perf.r4.Precision | perf.r4.F1 | perf.r4.False.Positives |
| 0.0 | 98.67 | 99.99 | 0.01 | 1.31 | 1.13 | 1.22 | 57.26229 |
| 0.5 | 98.36 | 99.99 | 0.01 | 1.62 | 1.40 | 1.50 | 143.15572 |
| 1.0 | 98.03 | 99.99 | 0.01 | 1.96 | 1.68 | 1.81 | 572.62290 |
| 1.5 | 97.67 | 99.99 | 0.01 | 2.32 | 1.98 | 2.14 | 801.67206 |
| 2.0 | 97.05 | 99.99 | 0.01 | 2.94 | 1.68 | 2.14 | 916.19664 |
| 2.5 | 96.53 | 99.99 | 0.01 | 3.45 | 1.97 | 2.51 | 1202.50809 |
| 3.0 | 96.08 | 99.99 | 0.01 | 3.90 | 2.22 | 2.83 | 1317.03267 |
| 3.5 | 95.63 | 99.99 | 0.01 | 4.35 | 1.87 | 2.61 | 1861.02442 |
| 4.0 | 95.08 | 99.99 | 0.01 | 4.91 | 2.10 | 2.94 | 2405.01618 |
| 4.5 | 94.36 | 99.98 | 0.01 | 5.63 | 1.93 | 2.87 | 2863.11450 |
| 5.0 | 93.51 | 99.98 | 0.01 | 6.48 | 2.21 | 3.30 | 3578.89312 |
| 5.5 | 92.76 | 99.98 | 0.01 | 7.23 | 2.46 | 3.67 | 4180.14716 |
| 6.0 | 92.18 | 99.98 | 0.02 | 7.81 | 2.22 | 3.46 | 4724.13892 |
| 6.5 | 91.36 | 99.97 | 0.02 | 8.63 | 1.85 | 3.05 | 5497.17983 |
| 7.0 | 90.49 | 99.97 | 0.03 | 9.49 | 1.81 | 3.04 | 6098.43388 |
| 7.5 | 89.49 | 99.97 | 0.03 | 10.50 | 1.64 | 2.84 | 7071.89281 |
| 8.0 | 88.78 | 99.96 | 0.04 | 11.21 | 1.38 | 2.46 | 8102.61402 |
| 8.5 | 87.95 | 99.95 | 0.05 | 12.03 | 1.30 | 2.34 | 9305.12211 |
| 9.0 | 87.27 | 99.95 | 0.05 | 12.72 | 1.22 | 2.22 | 10364.47448 |
| 9.5 | 86.40 | 99.94 | 0.06 | 13.59 | 1.17 | 2.16 | 11910.55630 |
| 10.0 | 85.57 | 99.94 | 0.06 | 14.42 | 1.24 | 2.29 | 13513.90042 |
| 10.5 | 84.67 | 99.94 | 0.06 | 15.32 | 1.26 | 2.32 | 14974.08881 |
| 11.0 | 83.75 | 99.94 | 0.06 | 16.23 | 1.33 | 2.46 | 17178.68698 |
| 11.5 | 82.97 | 99.93 | 0.06 | 17.02 | 1.33 | 2.47 | 19812.75231 |
| 12.0 | 82.05 | 99.93 | 0.07 | 17.93 | 1.34 | 2.50 | 22303.66192 |
| 12.5 | 81.10 | 99.93 | 0.07 | 18.89 | 1.36 | 2.53 | 25882.55504 |
| 13.0 | 80.17 | 99.93 | 0.07 | 19.81 | 1.37 | 2.55 | 30005.43992 |
| 13.5 | 79.54 | 99.93 | 0.07 | 20.45 | 1.41 | 2.63 | 34471.89853 |
| 14.0 | 78.65 | 99.92 | 0.07 | 21.33 | 1.41 | 2.65 | 39138.77516 |
| 14.5 | 77.92 | 99.92 | 0.07 | 22.07 | 1.46 | 2.74 | 44149.22553 |
| 15.0 | 77.22 | 99.92 | 0.08 | 22.76 | 1.40 | 2.64 | 49703.66765 |



### 5.5. Recommended Final Models

The following two models are recommended for SP implementation

* Model 1. and
* Model 2. and :

### 5.6. SP Implementation

#### 5.6.1. Definitions

Model Formula to Built in the Safer Payment

(1). idx.max.30 = maximum of $\max\{x\_1, x\_2, \dots, x\_{30}\}$.  
(2). idx.avg.30 = average of $\max\{x\_1, x\_2, \dots, x\_{30}\}$.  
(3). idx.max.05 = maximum of $\max\{y\_1, y\_2, \dots, y\_{5}\}$.  
(4). idx.avg.05 = average of $\max\{y\_1, y\_2, \dots, y\_{5}\}$.  
(5). idx.ctr = transaction counter starting from $x\_1$.

The SP commands used in the definition of the above two models are given by:

`{r label, what=ever} idx01 =div(sub(sub(mult(15;idx.max.30);mult(13;idx.avg.30));mult(2;idx.avg.05));sub(idx.max.05;idx.avg.05)) \\ idx02 = div(sub(sub(mult(20;idx.max.30);mult(18;idx.avg.30));mult(2;idx.avg.05));sub(idx.max.05;idx.avg.05)) `

#### 5.6.2. SP Rule

The suggested decision boundary for both models is a number between 0.5 and 1.0. The actual decision bound will eventually be determined by SP through a testing procedure (trial-and-error). The initial recommended boundary is 0.75.

SP Rule:

**IF** IDX < 0.75 **AND** idx.ctr > 34 **THEN** Alert

## 6. Retrospective Evaluation of the Final Models

Two types of analyses will be presented in this section: leakage analysis and lift analysis.

### 6.1. Leakage Analysis

Two pieces of information will be extracted from the models: False negatives (i.e., the origination point is not the fraud department) and late catch.

We calculate the index value for each WP transaction under the two final recommended models and then select a sequence of cut-off index values (decision boundaries) to assess leakage and the lift from the baseline (random guess) in different derived risk intervals.

The following function calculates the leakage information of the two classification models.

Tabulating the leakage information

The leakage table for model idx.5.15. The unit of the last column is the dollar amount that could be saved if this model were in place last year.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | tot.catch | tot.fraud.dept | tot.other.dept | late.fraud.dept | late.other.dept | ontime.fraud.dept | pred.idx.savings |
| 0 | 170 | 106 | 64 | 67 | 53 | 39 | 394201.8 |
| 0.5 | 873 | 647 | 226 | 438 | 199 | 209 | 1407909.2 |
| 1 | 1731 | 1371 | 360 | 964 | 319 | 407 | 2294720.9 |
| 1.5 | 2547 | 2072 | 475 | 1478 | 417 | 594 | 2949726.5 |
| 2 | 3207 | 2636 | 571 | 1878 | 505 | 758 | 3520709.6 |
| 2.5 | 3760 | 3117 | 643 | 2206 | 570 | 911 | 3937905.9 |
| 3 | 4215 | 3501 | 714 | 2501 | 632 | 1000 | 4303412.9 |
| 3.5 | 4563 | 3807 | 756 | 2735 | 672 | 1072 | 4535421.5 |
| 4 | 4861 | 4069 | 792 | 2935 | 704 | 1134 | 4740449.4 |
| 4.5 | 5104 | 4284 | 820 | 3102 | 729 | 1182 | 4867575.2 |
| 5 | 5297 | 4449 | 848 | 3230 | 754 | 1219 | 5022875.8 |

The leakage table for model idx.5.20.

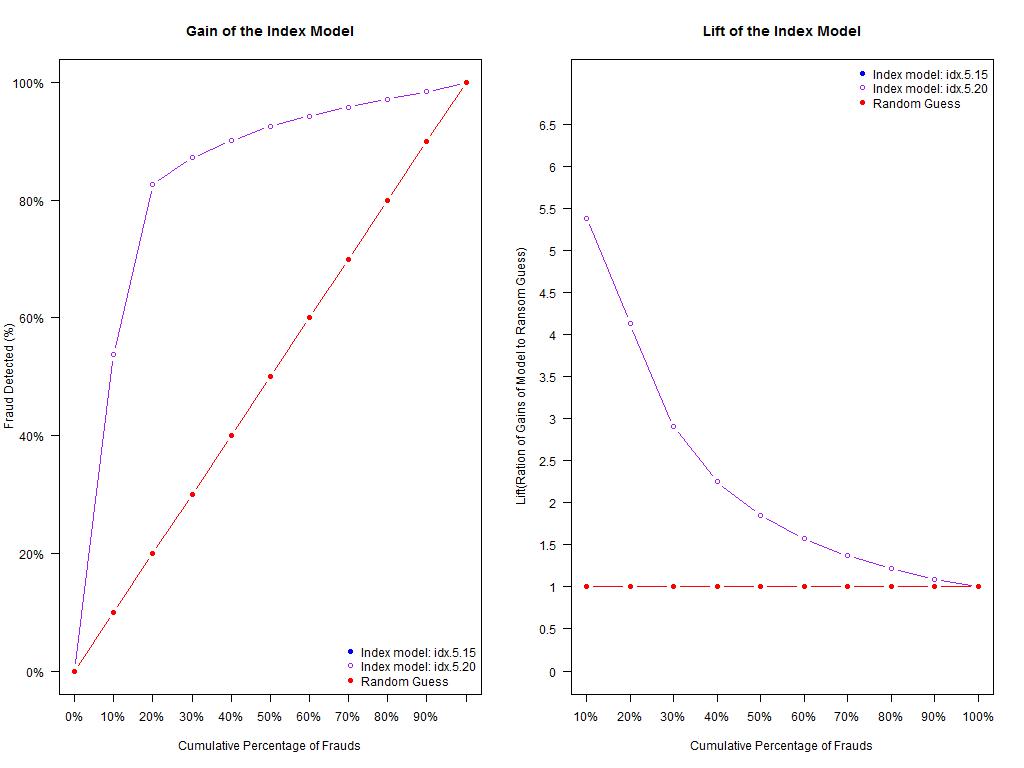
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | tot.catch | tot.fraud.dept | tot.other.dept | late.fraud.dept | late.other.dept | ontime.fraud.dept | pred.idx.savings |
| 0 | 102 | 58 | 44 | 34 | 36 | 24 | 209578.0 |
| 0.5 | 498 | 344 | 154 | 232 | 137 | 112 | 914467.1 |
| 1 | 1115 | 848 | 267 | 581 | 235 | 267 | 1656875.2 |
| 1.5 | 1763 | 1399 | 364 | 982 | 323 | 417 | 2329090.0 |
| 2 | 2397 | 1937 | 460 | 1377 | 406 | 560 | 2868239.0 |
| 2.5 | 2924 | 2384 | 540 | 1699 | 476 | 685 | 3315488.4 |
| 3 | 3373 | 2779 | 594 | 1969 | 528 | 810 | 3694431.1 |
| 3.5 | 3787 | 3138 | 649 | 2224 | 575 | 914 | 3947678.5 |
| 4 | 4115 | 3418 | 697 | 2435 | 619 | 983 | 4263039.2 |
| 4.5 | 4420 | 3684 | 736 | 2635 | 654 | 1049 | 4426773.2 |
| 5 | 4654 | 3889 | 765 | 2796 | 679 | 1093 | 4596553.0 |

### 6.2. Lift Analysis

This subsection evaluates the actual predictive effect of the proposed model compared with the baseline model (random guessing or average fraud rate).

We first calculate the lift scores using the following R function.

#### 6.2.1. Lift Analysis



Case 6.2.1. Lift-gain of candidate models

### 6.3. Summary of Retrospective Analysis of the Final Models

The two models performed almost the same in terms of lift and gain. The leakage analysis also yields similar results. Note that the cut-offs of idx.5.20 are supposed to be slightly bigger than idx.5.15.

## 7. Some SP Implementation Notes

Here are some notes on the implementation of the model in the Safer Payments. Due to SP technical difficulties, the structure of sequence data has to be modified in order to make the model work in the Safer Payments. The initial proposal of the models is based on 35 consecutive transactions in the current one. The most recent 5 transactions will be used for idx-5trx type attributes and the next 30 consecutive transactions will be used to estimate the upper specification limit (USL).

SP has to use a time frame and collect data points within the time frame. However, the number of transactions falling into the given time frame is random. This will result in two “disconnected” sequences of data or two overlapped sub-sequences of data.

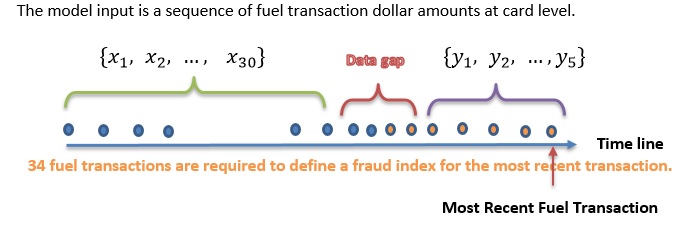
### 7.1. Scenario 1 - Two Time Frames

We still use two different time frames to collect data to define the two different types of attributes.

* **Time Frame: 0 - 10 Days** In this 10-day window, most cards will have at least 5 transactions including the current one. We use only the most recent 5 transactions (including the current transaction) to define the rolling index.

Remark: If the number of transactions is less than 5 (counter < 5), the corresponding index will not be used for detection. If the number of transactions is greater than 5, only the most recent 5 will be used in the definition of the rolling index.

* **Time Frame: 10 - 70 Days** Using a similar logic, 30 most recent transactions will be chosen from this time window to define idx-30trx type attributes and use these attributes to estimate the upper specification limit (USL) to be used in the definition of the rolling index (with most recent 5 transactions).

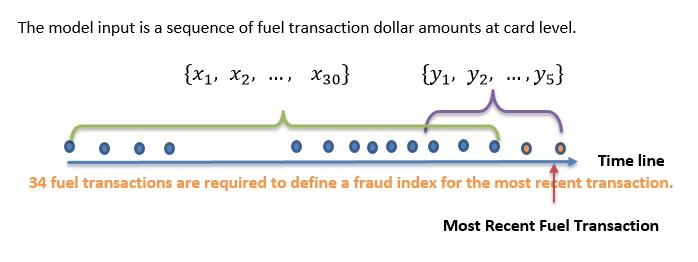


Data structure for the Index Model - two-time-frame

*Remark*: The data gap between the two sequence data (5 and 30 transactions are taken from the two disjoint time windows) may negatively impact the performance of the model.

### 7.1. Scenario 2 - Single Time Frame

The single time frame expands from zero to 90 days. The most recent 5 transactions (including the current one) are used to calculate the rolling index and the recent 34 transactions (excluding the current one) are used to estimate the upper specification limit (USL).



Data structure for the Index Model - Single-time-frame

**The attributes of this scenario are**

* C\_Idx\_35trx\_Max (does NOT include current)
* C\_Idx\_35trx\_Avg (does NOT include current)
* C\_Idx\_35trx\_Freq (includes current)
* C\_Idx\_5trx\_AvgB (includes current)
* C\_Idx\_5trx\_MaxB (includes current)

### 7.3. The Determination of Decision Boundary

Since the changes in the structure used in the index models, we need to re-tune the decision bound. We would expect to lose some degree of accuracy due to the modification of the structure of the sequence data.

## 8. Postface

Some analytical comments on the models.

* **Model Structure**

The design idea of the index models comes from the control charting methodology in statistical quality control. The “quality characteristic” used in the index model is transformed from the original spending using the process capability index (PCI) - this is where the name of this model comes from. Several special features of this model are

1. The model (I prefer algorithm) was defined in a way that the unit of the original attributes was canceled out in the analytical expression - that is, it is a unitless index value.
2. The definition of the index requires an upper specification limit (USL) of the original attribute. We use historical to use historical information on the attributes (not including the current value) to estimate the value based on a data-driven approach.
3. We also define a (time) window and move that window to collect values inside to calculate the rolling index.
4. The control chart idea was used to find the decision bound (lower specification limit - LSL) to make a decision - the detection rule. LSL is dependent on the various performance metrics and operation constraints as well - we consider LSL as a hyper-parameter.

* **What the model CAN NOT do**

The practical argument for the index model to work is that the spending patterns of fraudsters and genuine customers are different and the fraudsters don’t know the genuine customers’ spending patterns. If there is a *significant* change in spending patterns, the transaction that caused the change is considered a fraud. With this in mind, the index model **cannot** detect the following fraudulent transactions:

1. The fraudulent transactions fall into genuine customers’ usual spending range.
2. If a fraudster tests the WP card with a few dollar purchase, the index model will not be able to detect it.

* **Scientific Justification**

The fact that the index algorithm works is not by accident, it has a rich theory behind it.

1. The PCI is a metric that measures the potential shift of both process mean and variance. A fraudulent activity shifts both the mean and variance of genuine spending.
2. Since the definition of index is based on a moving window, the resulting index values are autocorrelated. If the rolling index involves k values, the resulting index constitutes an AR(k) process.
3. The inferential theory can be developed based on the AR(k) process - since the spending is skewed, we can develop a non-normal based theory for the AR(k) process that includes confidence band and testing procedures as well.

* **Model Strengths**

Index models have several strengths over other machine learning models.

1. The model we defined without using the labels (fraud flags). The performance will not be impacted by the inaccuracy of fraud labeling.
2. The model is independent of the fraud types - in other words, the change in fraud types will not affect the model performance.
3. The model is a parameter-free algorithm (no structural model), and no numerical optimization is involved. Mathematical operations used in the model are the four basic operations. There is no model training stage. The time used for testing a model is also negligible.
4. The model is scalable.