Beer-Talk @ Compass Security

Defeating Man-in-the-Browser Attacks with Clickstream

Fraud prevention using behavioral user profiling in Online Banking

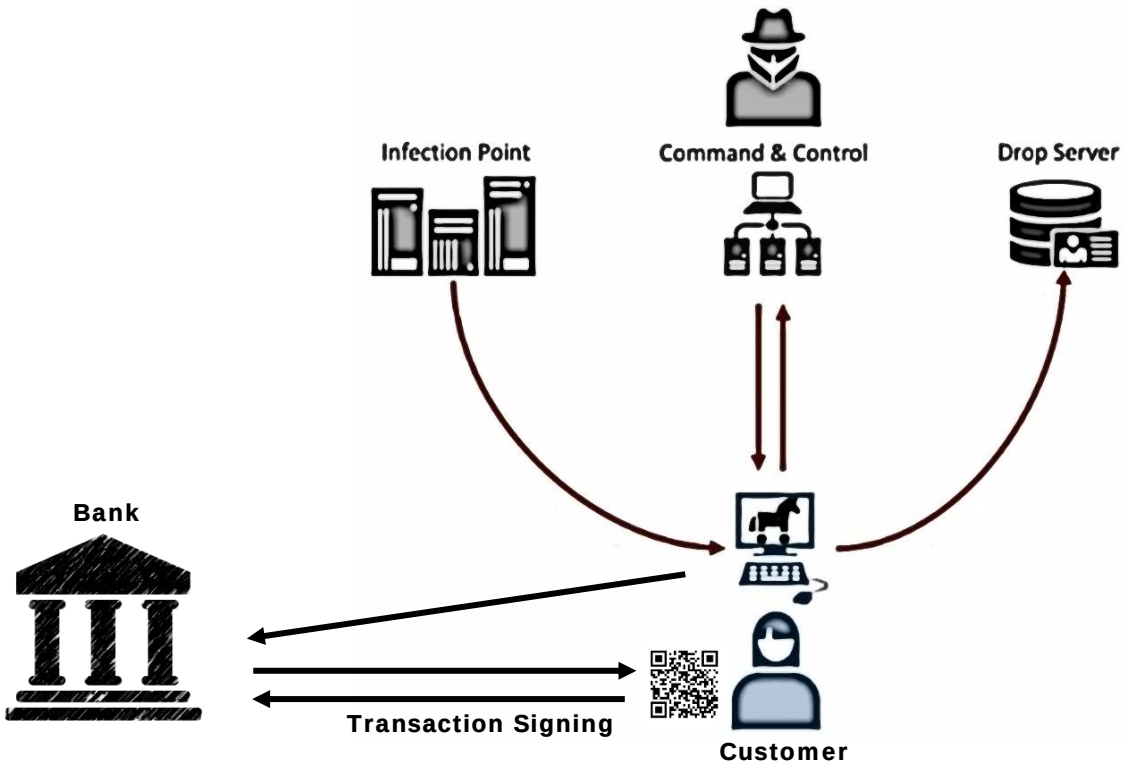
Jona: March 16th, 2017
Bern: March 23rd, 2017

David Stark, Lead Software Security Engineer
Master of Advanced Studies in Information Security and Big Data Analytics
from Lucerne University of Applied Sciences and Arts

david [dot] stark [at] protonmail [dot] com

Security Problem

The Man in the Browser



One Solution Among Others

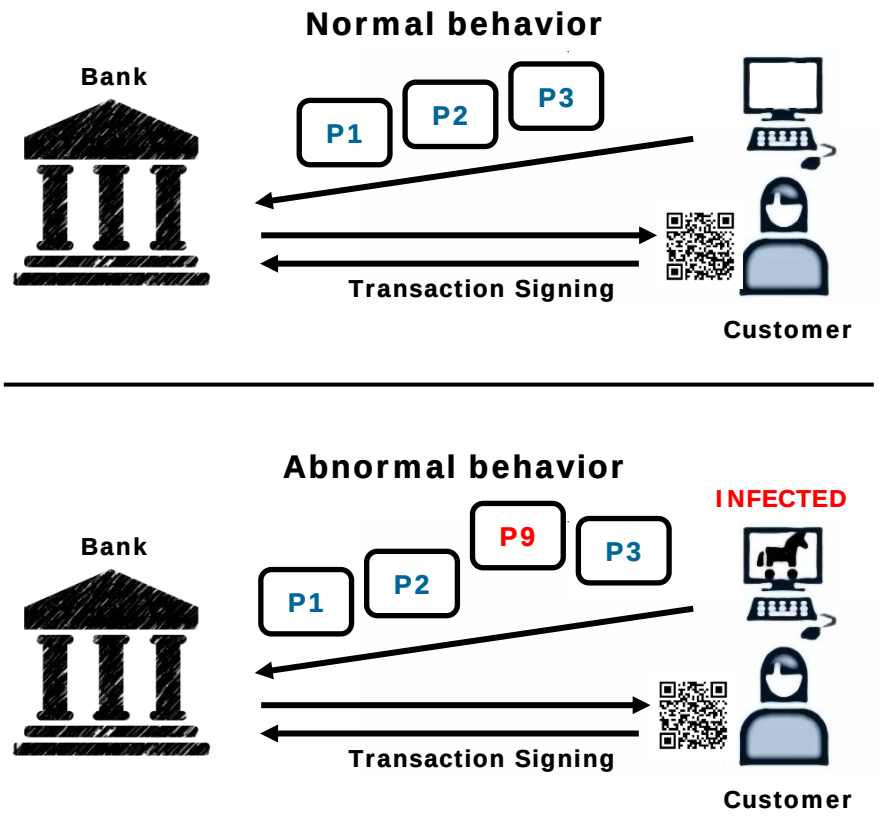
Behavioral Analysis and Anomaly Detection Techniques

using Clickstream Data

Clickstream - Definition

Clickstream is a record of user movements through time at a web site or another software application

Anatomy of Clickstream



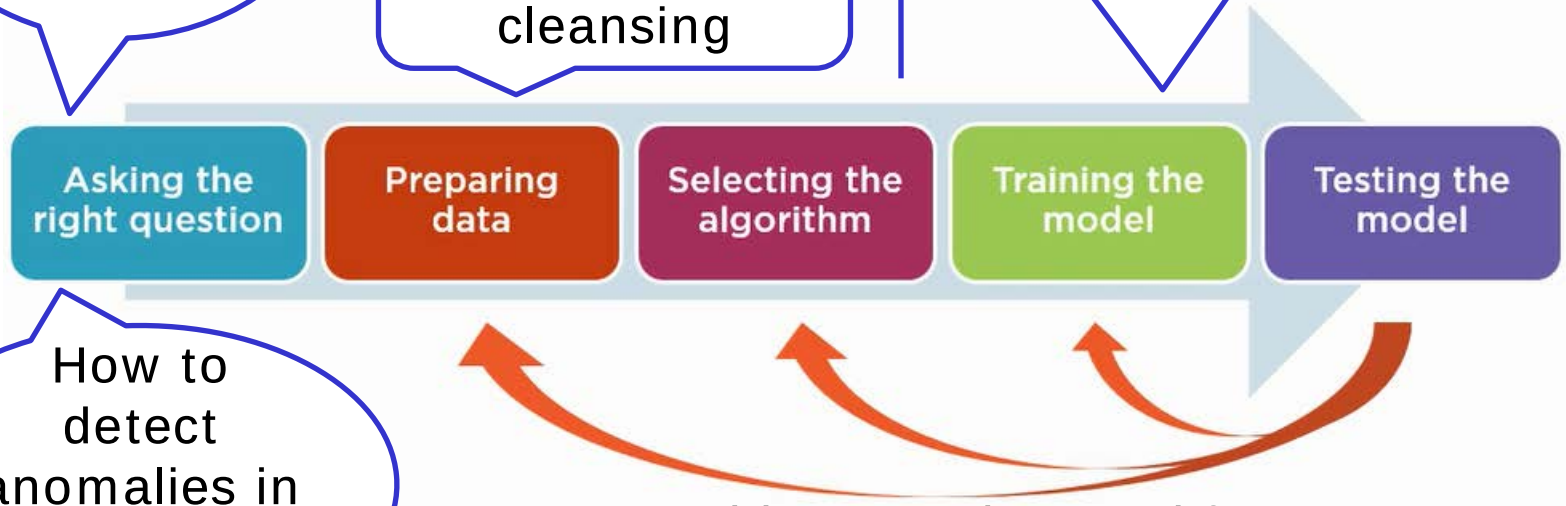
Methodology

How to profile user behavior based on clickstream data?

Markov chains + anomaly detection algorithms

R for data pre-processing and cleansing

Transition model
Frequency model
Inter-arrival times
Extreme Value Analysis
Time series' seasonality

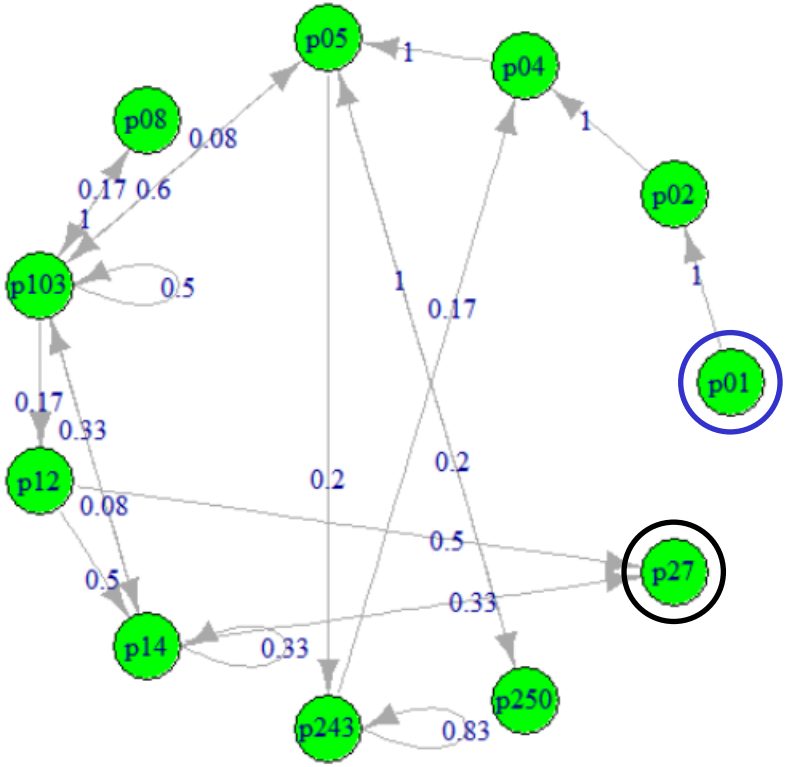


Machine learning workflow

How to detect anomalies in clickstream data?

Clickstream Data - Sample

Session ID	Clickstream
Session1	**p01**,p02,p04,p05,p103,p05,p250,p05,p103,p08,p103,p103,p103,p103,p12,p14,p14,p103,p14,**p27**
Session2	**p01**,p02,p04,p05,p243,p243,p243,p243,p243,p243,p04,p05,p103,p08,p103,p103,p103,p103,p12,**p27**



Clickstream Graph

Clickstream Graph created using R package «clickstream» (https://cran.r-project.org/web/packages/clickstream/index.html)

Profiling using Transition and Frequency Models

How to model user behavior?

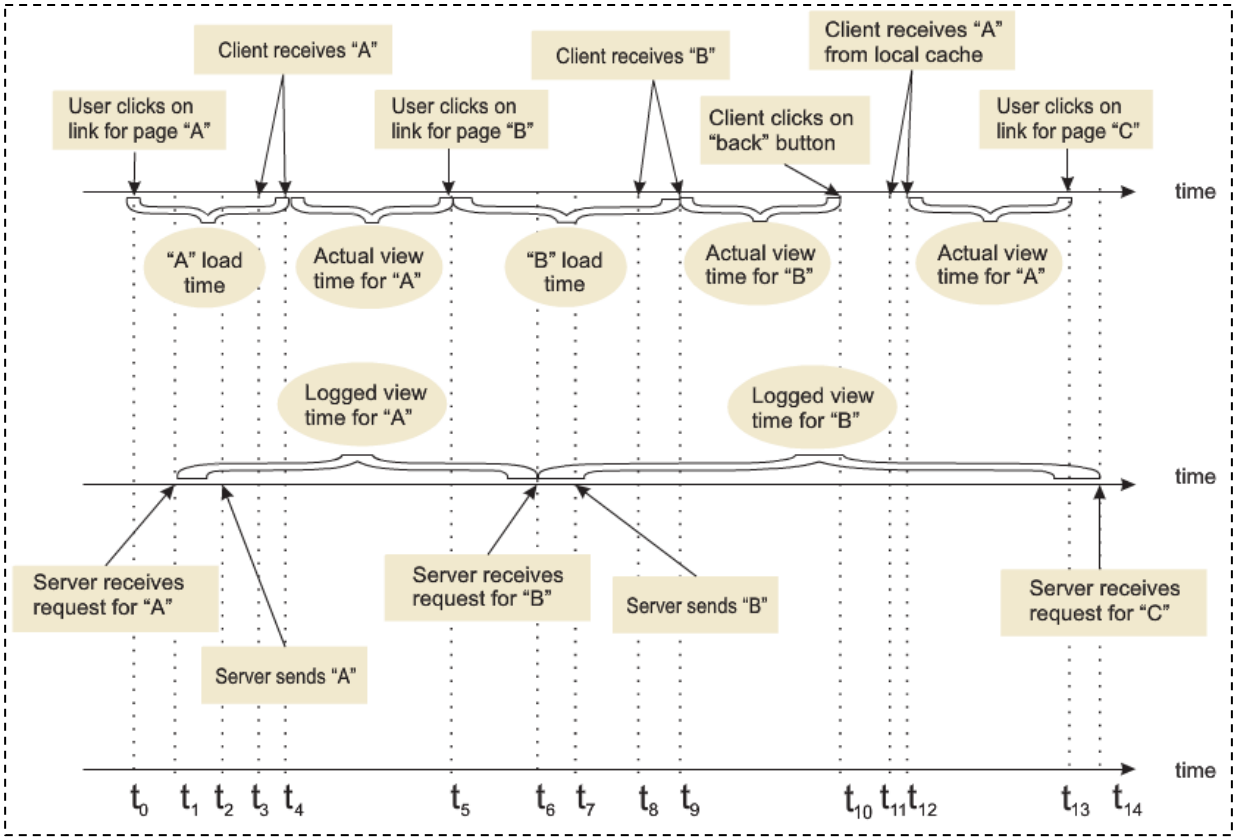
	p01	p02	p04	p05	p08	p103	p12	p14	p243	p250
p01										
p02	1									
p04		1							0.17	
p05			1			0.08				1
p08						0.17				
p103				0.6	1	0.50		0.3		
p12						0.17				
p14						0.08	0.5	0.3		
p243				0.2					0.83	
p250				0.2						
p27							0.5	0.3		

Transition Model

	p01	p02	p04	p05	p103	p250	p08	p12	p14	p27	p243
Session 1	1	1	1	3	7	1	1	1	3	1	0
Session 2	1	1	2	2	5	0	1	1	0	1	6

Frequency Model

Page Time Computation (inter-arrival time)



Inter-arrival time: t5 – t0, t10 – t5, etc.

Source: Gündüz-Ögüdücü S. (2010). Web Page Recommendation Models. Morgan & Claypool Publishers

Inter-Arrival Time Computation

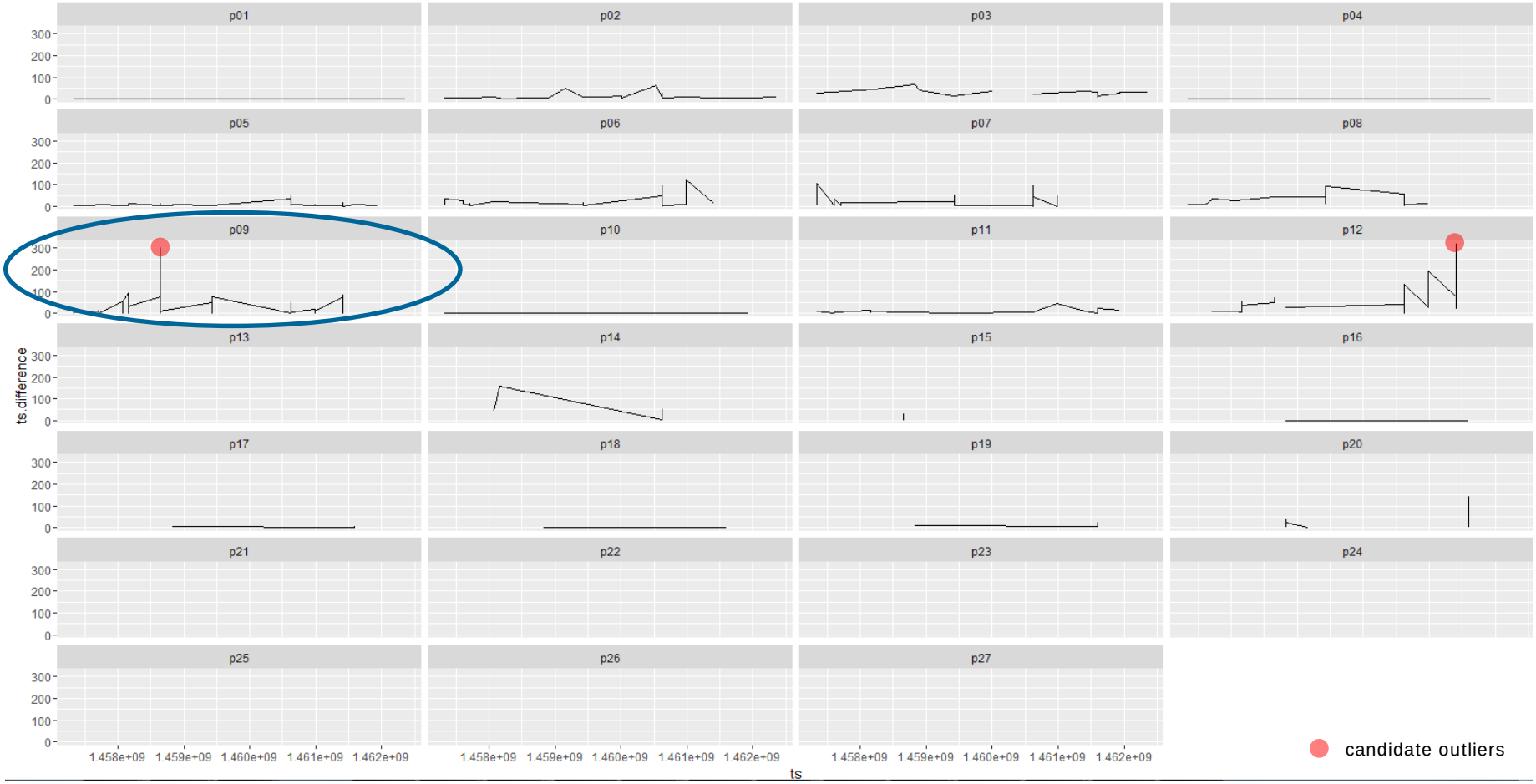
```
# convert date format, example: from 2016-03-07 08:41:41.492 CET to 1457336501
dataframe$ts <- as.POSIXlt(dataframe$ts)
dataframe$ts <- as.numeric(dataframe$ts)
# descending order for inter-arrival time calculation
dataframe <- dataframe [order(-dataframe[,"ts"]), ]
# compute inter-arrival time for each session
dataframe <- dataframe %>% group_by(sesid) %>% mutate(ts.interarrivaltime = lag(ts)-ts)
# revert sorting
dataframe <- dataframe [rev(order(-dataframe[,"ts"])), ]

# plot inter-arrival time vs timestamp for all pages
ggplot(data = dataframe, aes(x = factor(id), y = ts.interarrivaltime, color = id))
+ geom_line(aes(group = id)) + geom_point()

# plot inter-arrival time vs timestamp for each page
ggplot(dataframe, aes(x = ts, y = ts. interarrivaltime))
+ geom_line() + facet_wrap(~ id, ncol = 4)
```

Inter-Arrival Time Computation

```
ggplot(dataframe, aes(x = ts, y = ts.interarrivaltime)) + geom_line() + facet_wrap(~id, ncol = 4)
```



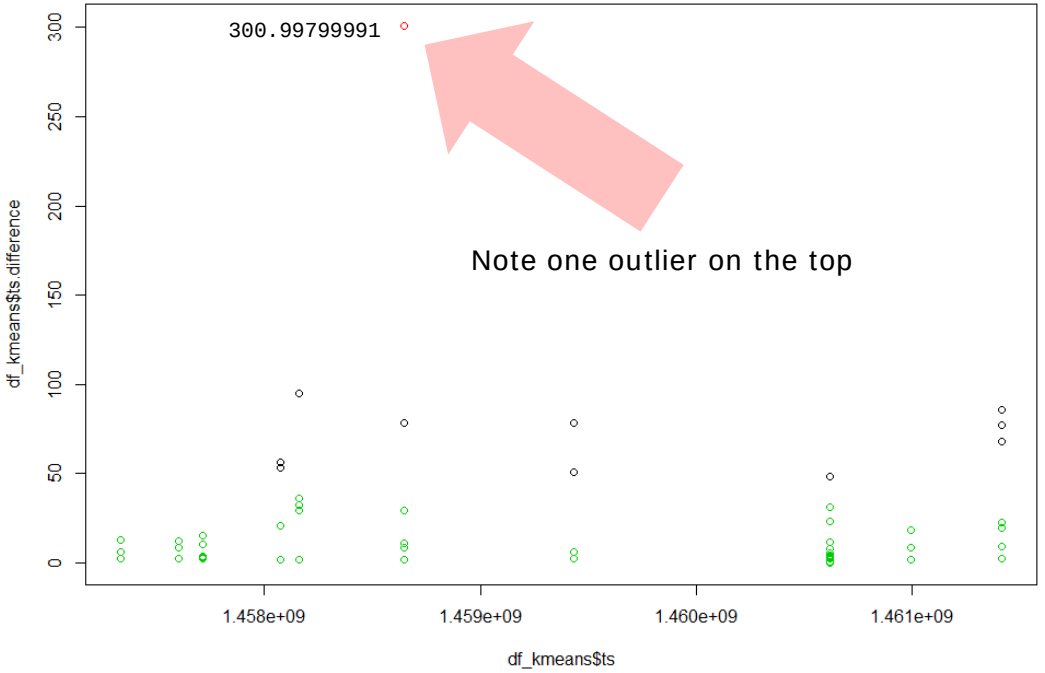
Anomaly and Outlier Detection using K-means Clustering Algorithm

How to detect anomalies in data?

```
# select complete cases (without NAs) from a test dataset for page p09
df_kmeans <- df_test[complete.cases(df_test) & df_test$id == "p09",]
# plot inter-arrival times mapped to the timeline
plot(x = df_kmeans$ts, y = df_kmeans$ts.difference)
# compute 3 clusters for visualization using K-means algorithms
kmeans.result <- kmeans(df_kmeans$ts.difference, 3)
# plot inter-arrival times and color the dots assigned to different clusters
plot(x = df_kmeans$ts, y = df_kmeans$ts.interarrivaltime, col = kmeans.result$cluster)
```

red cluster: 1 point
black cluster: 10 points
green cluster: 49 points

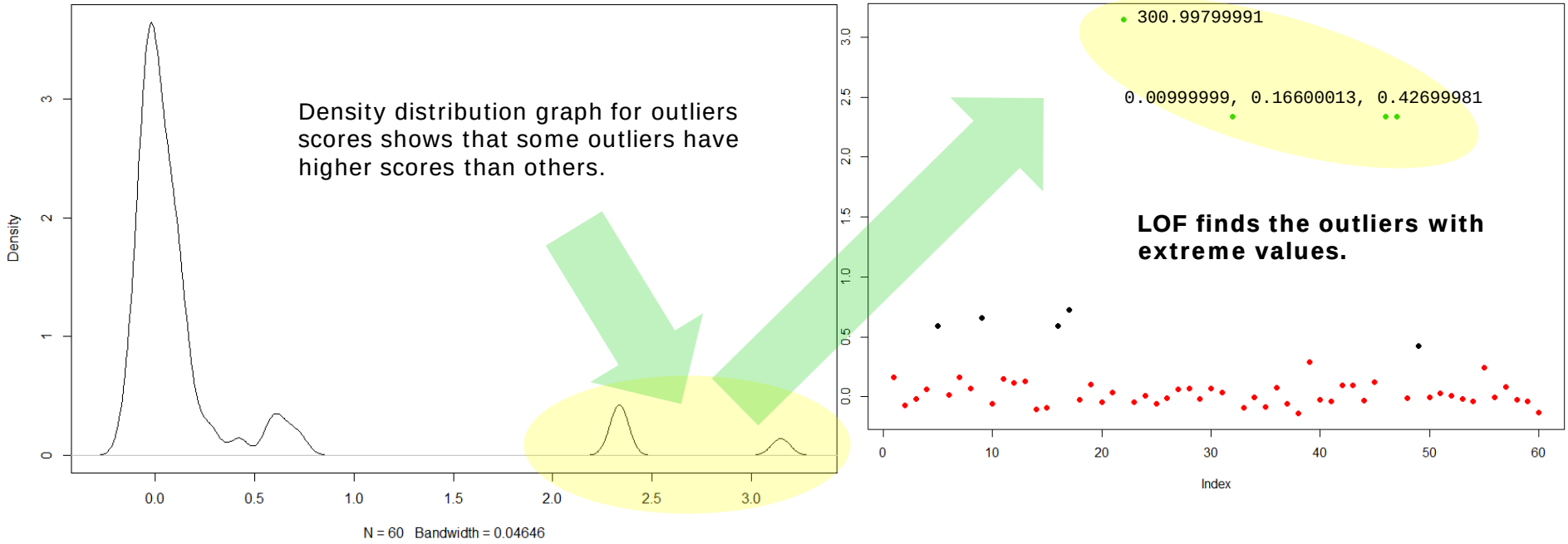
Visual analysis shows that the **red** cluster contains an outlier. However, K-Means algorithm does not provide scoring for the outliers, therefore additional computational steps are required to score the outliers. Very small values (e.g. 0.00999999, 0.16600013) are to be found in **green** cluster. However, higher number of clusters may resolve this issue.



Outlier Detection using LOF (local outlier factor) Algorithm

How to detect anomalies in data?

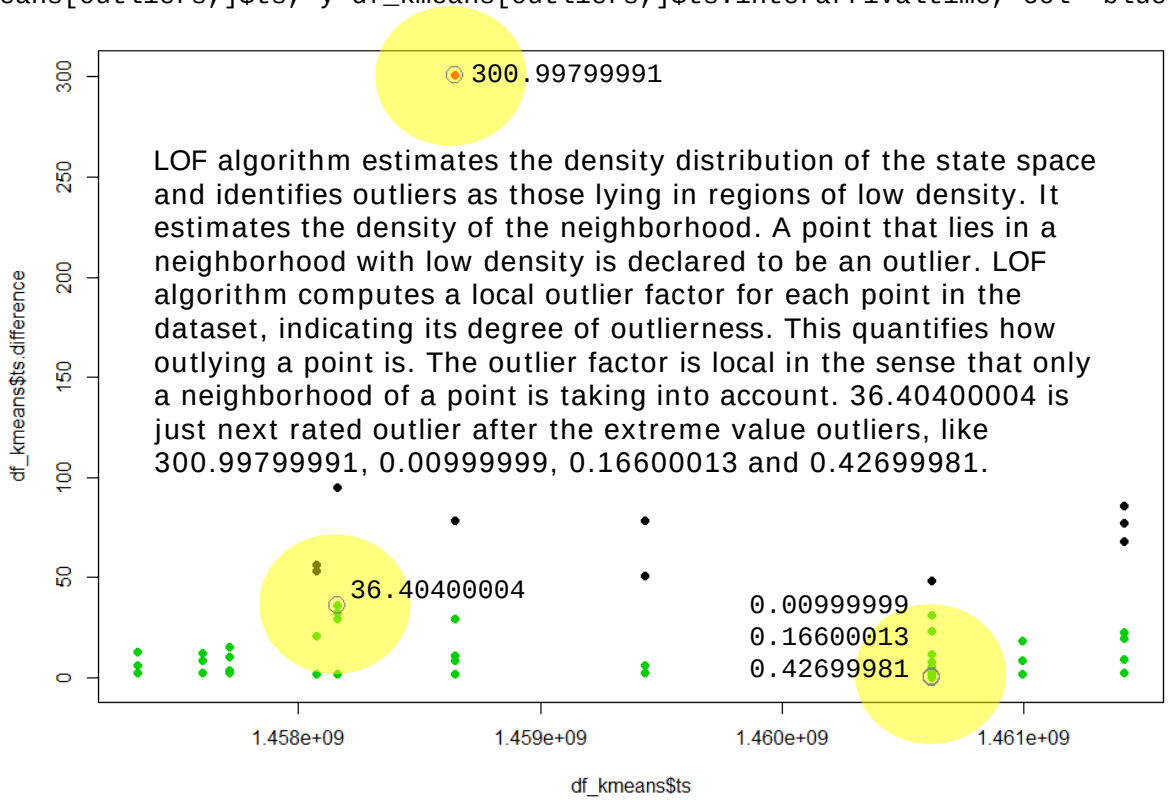
```
# load required package "DMwR" containing implementation for the LOF algorithm
# https://cran.r-project.org/web/packages/DMwR/DMwR.pdf
library(DMwR)
# lofactor function produces a vector of local outlier factors with outlier scores
outlier.scores <- lofactor(df_kmeans$ts.interarrivaltime, k=3)
# plot density distribution of outlier scores for inter-arrival times for p09 (plot on the left)
plot(density(log(outlier.scores[complete.cases(outlier.scores)])))
# compute 3 clusters for visualization using K-means algorithms
kmeans.outliers <- kmeans(outlier.scores, 3)
# plot outlier scores (plot on the right), note that log function "zooms in" the small values
plot(log(outlier.scores), col=kmeans.outliers$cluster, pch=19, cex=1)
```



Outlier Detection using LOF (local outlier factor) Algorithm

The graph below shows visually the top 5 the outliers found using LOF algorithm.

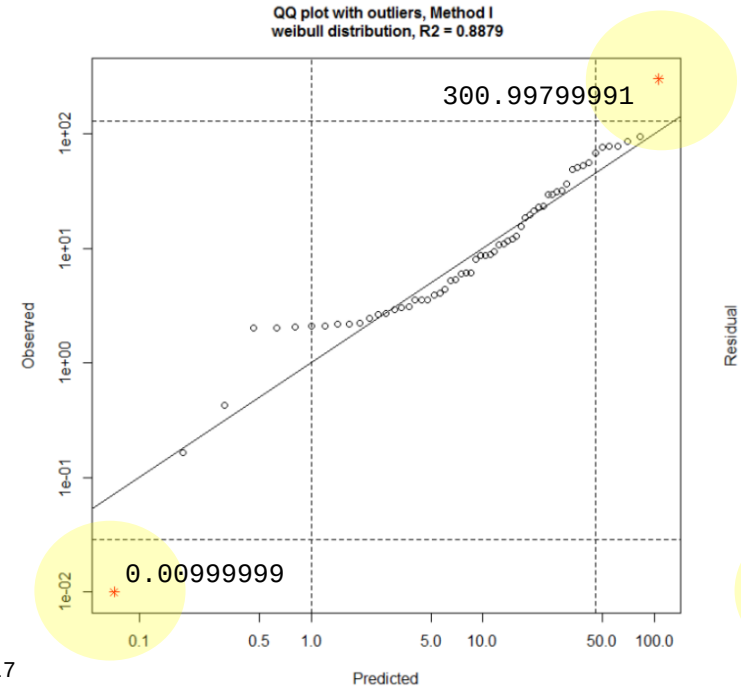
```
# get top 5 outliers found by LOF algorithm
outliers <- order(outlier.scores, decreasing=T)[1:5]
# plot outliers, note that not only extreme values have been ranked at top 5 outliers
plot(x=df_kmeans$ts, y=df_kmeans$ts.interarrivaltime, col=kmeans.result$cluster, pch=19, cex=1)
# highlight the outlier points
points(x=df_kmeans[outliers,]$ts, y=df_kmeans[outliers,]$ts.interarrivaltime, col="blue", pch=21, cex=2)
```

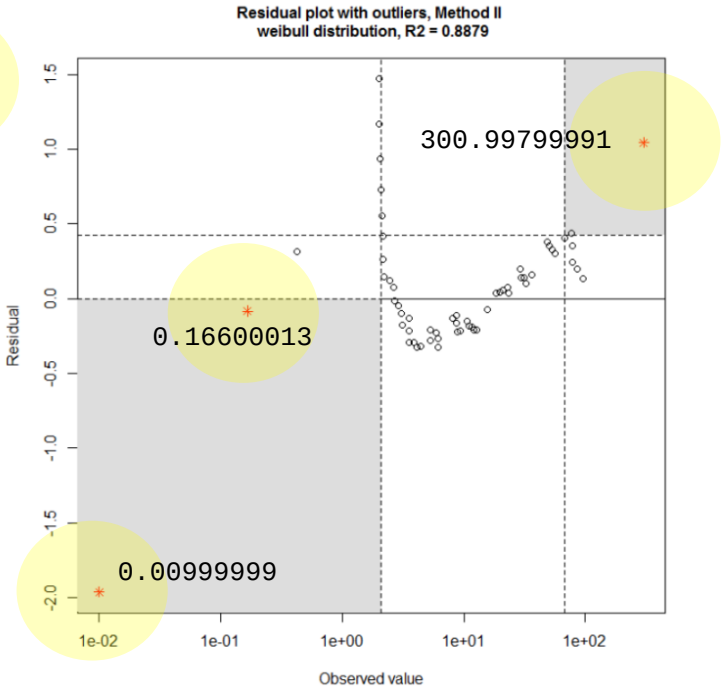


Extreme Value Analysis and Weibull Distribution

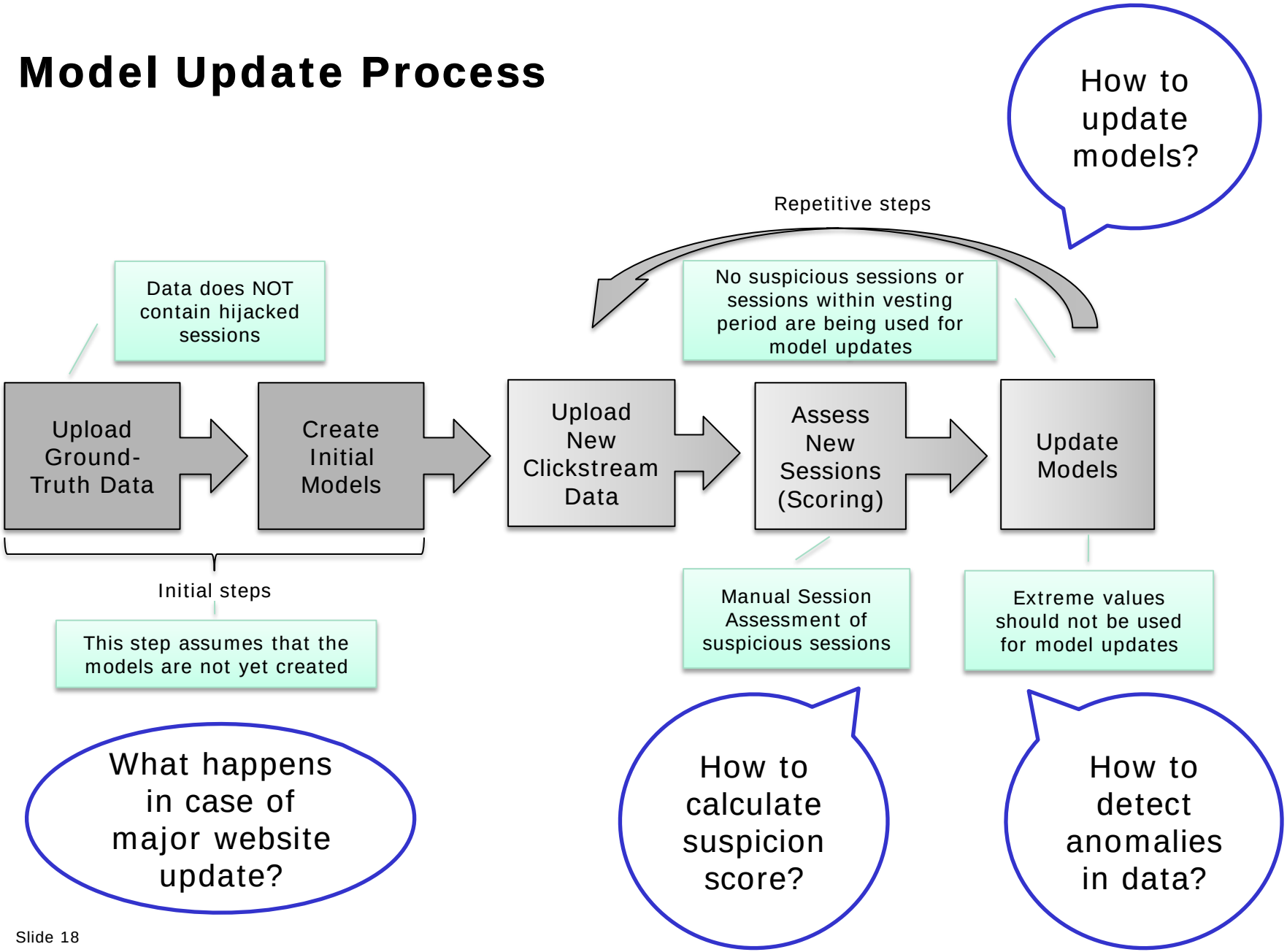
How to detect anomalies in data?

```
# load required package “extremevalues”
# https://cran.r-project.org/web/packages/extremevalues/extremevalues.pdf
library(extremevalues)
# copy data frame and extract one-dimensional vector with ts.difference values
yy <- df_kmeans; y <- yy$ts.interarrivaltime;
# compute outliers using first method and weibull distribution
K <- getOutliers(y, method='I', rho=c(0.5, 0.5), distribution='weibull', FLim=c(0.1,0.9))
# compute outliers using second method and weibull distribution
L <- getOutliers(y, method='II', alpha=c(0.5, 0.05), distribution='weibull', FLim=c(0.1,0.9))
# plot outliers found using first and second methods
outlierPlot(y, K, mode="qq"); outlierPlot(y, L, mode="residual");
```

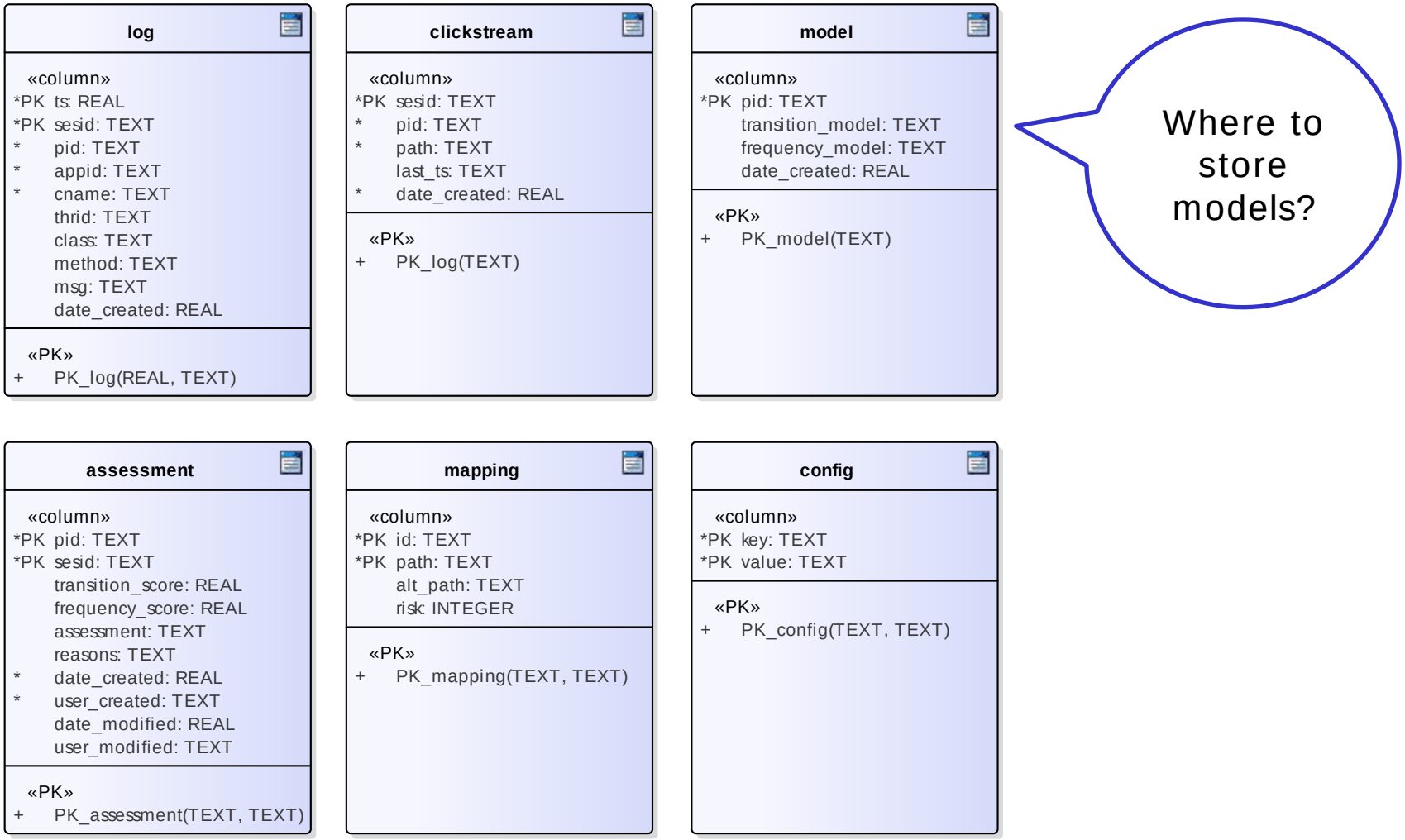




Model Update Process



Data Model Design



Serialized Transition and Frequency Models

How to store models?

"INSERT INTO model (pid, transition_model, frequency_model) VALUES('C0001000',
'A\n2\n197125\n131840\n787\n22\n14\n22\n0\n0\n0\n1\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n14\n22\n0\n0.2608695652173913\n
82609\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0.0434782608695652\n0\n0.08695652173913043\n0.4347826086956522\n14\n22\n0\n0\n0\n1\n0\n0\n0\n
0\n0\n0\n0\n14\n22\n0\n0\n0.0625\n0.03125\n0\n0.09375\n0.46875\n0\n0\n0.21875\n0.03125\n0\n0.0625\n0.03125\n0\n0\n0\n0\n0\n0\n0\n0\n14\n
66666667\n0\n0\n0\n0.1666666666666667\n0.1666666666666667\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n14\n22\n0.0666666666666667\n0\n0\n0\n
\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0.0666666666666667\n14\n22\n0\n0\n0\n0\n0\n0\n0\n0\n0.07407407407407407\n0\n0\n0\n0\n0\n0\n0.7
481481481481\n0\n0\n14\n22\n0\n0\n0\n0\n0\n0\n0.8\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0.2\n14\n22\n0\n0\n0.5\n0\n0\n0\n0\n0\n0.5
n0\n0\n14\n22\n0\n0\n0\n0\n0.0303030303030303\n0\n0.0303030303030303\n0\n0\n0.696969696969697\n0\n0.0303030303030303\n0\n0\n0\n0.0
n0.1212121212121212\n14\n22\n0\n0\n0\n0.3333333333333333\n0.3333333333333333\n0\n0\n0\n0\n0\n0.3333333333333333\n0\n0\n0\n0\n0\n
0\n0\n0\n0\n0\n0\n0\n0\n0.8333333333333334\n0\n0\n0\n0.1666666666666667\n0\n0\n0\n0\n0\n0\n14\n22\n0\n0\n0\n0.5\n0\n0\n0\n0\n0\n0\n
\n14\n22\n0\n0\n0\n0\n0\n0\n1\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n14\n22\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n
04\n0\n0\n0\n0.16\n0\n0\n0\n0.12\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0.04\n0\n0\n14\n22\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n1\n0\n
n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n1\n0\n0\n0\n14\n22\n 0.333333 n
0\n0\n0\n14\n22\n0\n0.1666666666666667\n0.166666666666 0\n0\n0 6666
667\n14\n22\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n 0\n0\n 0\n0\n0\n0 6\n1\n26
2153\n5\nnames\n16\n22\n262153\n6\nX1.p01\n262153\n6\nX 6215 n6\nX1.p09\n262153\n6\nX1.p10\n262153
\n7\nX1.p100\n262153\n7\nX1.p101\n262153\n7\nX1.p102\n26 3\n7\nX1.p106\n262153\n7\nX1.p107\n262153\n6
\nX1.p11\n262153\n6\nX1.p16\n262153\n6\nX1.p17\n262153\n 27\n1026\n1\n262153\n9\nrow.names\n16\n22\n2
62153\n3\np01\n262153\n3\np03\n262153\n3\np04\n262153\n 4\np100\n262153\n4\np101\n262153\n4\np102\n
262153\n4\np103\n262153\n4\np104\n262153\n4\np105\n2621 62153\n3\np17\n262153\n3\np21\n262153\n3\np2
5\n262153\n3\np26\n262153\n3\np27\n1026\n1\n262153\n5\n
'A\n2\n197125\n131840\n787\n22\n14\n21\n1\n1\n1\n1\n2\n1 2\n1\n1\n1\n1\n1\n2\n1\n3\n3\n1\n2\n1\n3\n1\n
1\n2\n14\n21\n1\n1\n0\n1\n0\n1\n1\n1\n1\n1\n3\n3\n4\n1\n1 2\n1\n1\n0\n3\n1\n1\n0\n14\n21\n1\n1\n0\n1\n
0\n1\n1\n1\n2\n2\n1\n1\n1\n1\n1\n2\n0\n1\n4\n1\n0\n14\n21 4\n21\n1\n1\n0\n1\n1\n1\n0\n1\n1\n1\n1\n1\n1\
n1\n1\n1\n1\n1\n1\n1\n0\n14\n21\n0\n1\n0\n0\n0\n0\n0\n0\n n0\n0\n0\n5\n0\n5\n1\n0\n0\n11\n5\n0\n0\n3\n
14\n21\n0\n1\n0\n0\n0\n0\n0\n1\n0\n0\n1\n1\n1\n0\n0\n0\n0 n0\n0\n0\n0\n0\n0\n14\n21\n0\n0\n3\n0\n0\n0\
n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n2\n0\n0\n0\n1\n14\n21\n0\n n0\n0\n0\n0\n0\n0\n1\n0\n0\n0\n0\n0\n0\n0\n0
\n0\n0\n0\n0\n0\n0\n14\n21\n0\n0\n0\n0\n0\n0\n1\n0\n0\n0\ 0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n14\n21
\n0\n0\n0\n0\n0\n0\n0\n1\n0\n0\n1\n1\n1\n0\n0\n0\n0\n1\n0 2\n0\n0\n0\n14\n21\n0\n0\n0\n0\n0\n0\n0\n0\n
0\n0\n0\n0\n2\n0\n0\n0\n0\n2\n0\n0\n0\n14\n21\n0\n0\n0\n0 n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n1\n0\
n0\n0\n0\n0\n14\n21\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0\n0 62153\n3\np01\n262153\n3\np05\n262153\n3\np10
\n262153\n3\np11\n262153\n3\np03\n262153\n3\np26\n262153\n3\np27\n262153\n3\np04\n262153\n4\np102\n262153\n3\np25\n262153\n3\np09\n262153\n3\np08\n262153\n3\np21\
n262153\n3\np16\n262153\n3\np17\n262153\n4\np107\n26215 62153\n4\np106\n262153\n4\np103\n1026\n1\n2
62153\n9\nrow.names\n16\n21\n262153\n52\nUVennp4oWOGm jdgpDsG7fWijriB8P0qg-
5g1VhsU5BZaKVqoqNZnP_uR\n262153\n52\nN8jFC7djZ7KOLB6p DiWb2yUvbDzoc7B6aq1G4XgbtVS5cASxBBfxOA1u3\
n262153\n52\nvcdNd9W9nQELu-
AsWSoJZ8SvQSvE4gU7acAAXsM3CVCmaX3M7Jte\n262153\n52\n 3\n52\nTOBRArXbzFQM1i-
f3OKx9jMsXtaSUdiZzJV0Hk03VFq_P0e3SDfx\n262153\n52\n0iJRL I4H17dDCXQoux12zMv2MlFbJ3plRXhyD7eKGGyfpW
8f9HpS0\n262153\n52\nK2JqPhTLhkcpuC8vM0WBp-iu4uOmn-un 46dbPCevKyqwQ8w9Zcj-
LW7g\n262153\n52\ndwhq_hQOfwSONfPyA9S9HwitRh7xdBYQiBr FQoROn1u-
662J9ZwK_T\n262153\n52\nEvxranVMQh_6GCI8wKewmZiT4DHj 60OrHFm0vtU_BzOLIRpKxccsZRcbb\n262153\n52\
nsYx0S520FdDK8x6RP6vQUQbgdFTxH7C3kCkoBqsx4AOUSdMvbVAw\n262153\n52\nnvB0rG3v438HCRNykTtxEFSHLHIps4LOO2FexiR_kto7ggL4HqNt\n262153\n52\nvTN1DVwnOHtkXb6LuDRQb
9fDoPKrpubvZoJgSzIbFDW7tlQb3wNF\n262153\n52\n7OKLBJhHcyTeVVBAWlPfjuPErsnOpqTLi8Viba7c5pz1XRH_ko9R\n262153\n52\n8J6fnXML7MQVPFnAMSJUgJwk4sk6d2Aa0TaMX6hxuHkmrS
wkg6NY\n262153\n52\nKcJ0zTCEG2ixqeSNlXYaRv7CmqetG_uBxo2_MiKMo4kPDiHXi-gy\n1026\n1\n262153\n5\nclass\n16\n1\n262153\n10\ndata.frame\n254\n')"

	p01	p02	p04	p05	p08	p103	p12	p14	p243	p250
p01										
p02	1									
p04		1							0.17	
p05			1			0.08				1
p08						0.17				
p103				0.6	1	0.50		0.3		
p12						0.17				
p14						0.08	0.5	0.3		
p243				0.2					0.83	
p250				0.2						
p27							0.5	0.3		

	p01	p02	p04	p05	p103	p250	p08	p12	p14	p27	p243
Session 1	1	1	1	3	7	1	1	1	3	1	0
Session 2	1	1	2	2	5	0	1	1	0	1	6

Scoring

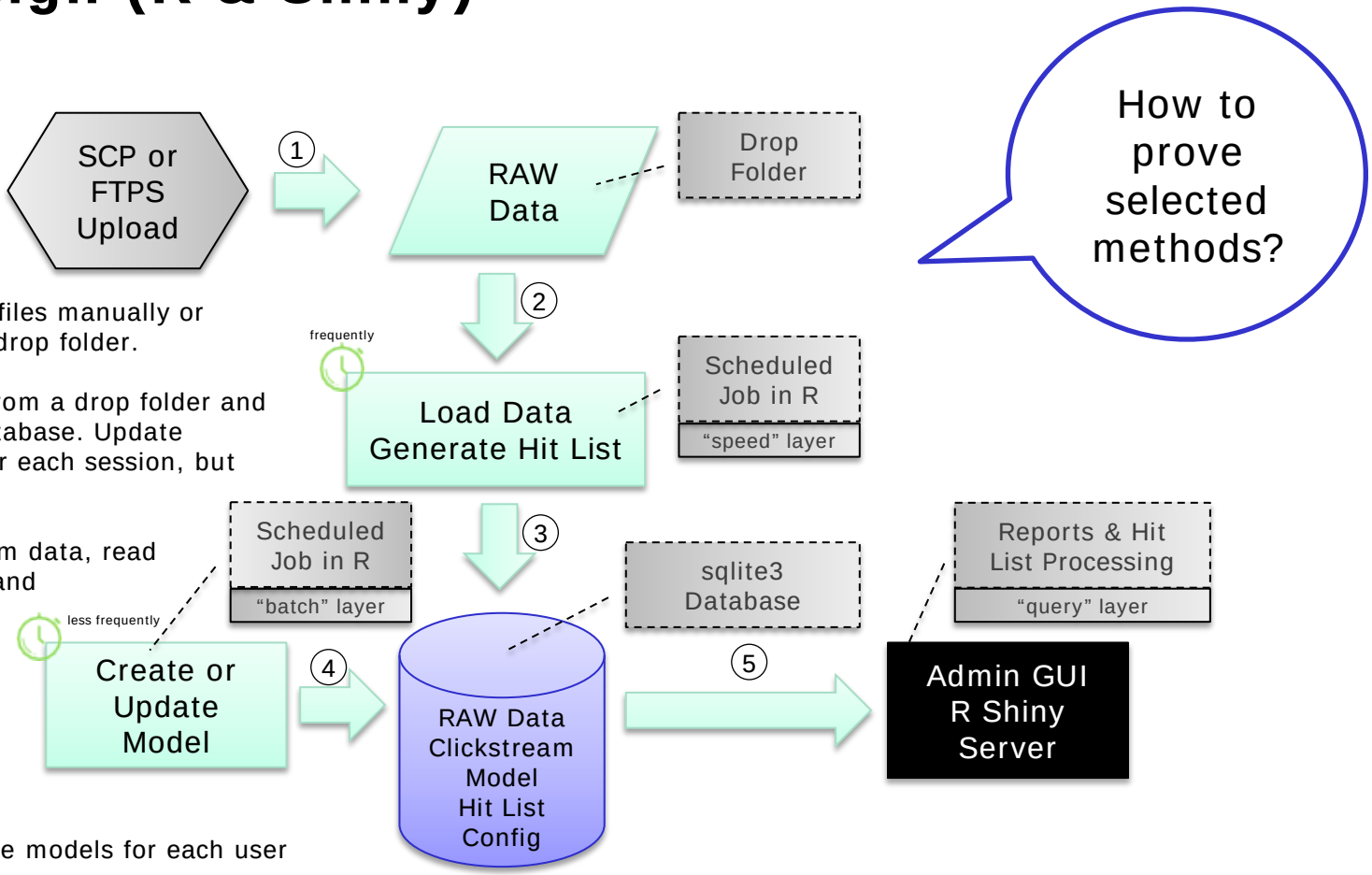
How to score user sessions?

Session suspicion scores are calculated based on each model. Final score is a sum of all weighted scores.

$$\#high\ risk\ transaction\ violations \times high\ risk\ page\ level +$$
$$\#medium\ risk\ transaction\ violations \times medium\ risk\ page\ level +$$
$$\#low\ risk\ transaction\ violations \times low\ risk\ page\ level$$
$$= \boldsymbol{score\ based\ on\ a\ model}\ (\boldsymbol{not\ normalized})$$

Page risk levels:
3 – high,
2 – medium and
1 – low

PoC Design (R & Shiny)



(1) Upload the log files manually or automatically to a drop folder.

(2) Read log files from a drop folder and store in a sqlite database. Update clickstream data for each session, but not the model.

(3) Read clickstream data, read computed models and generate hit list.

(4) Create or update models for each user

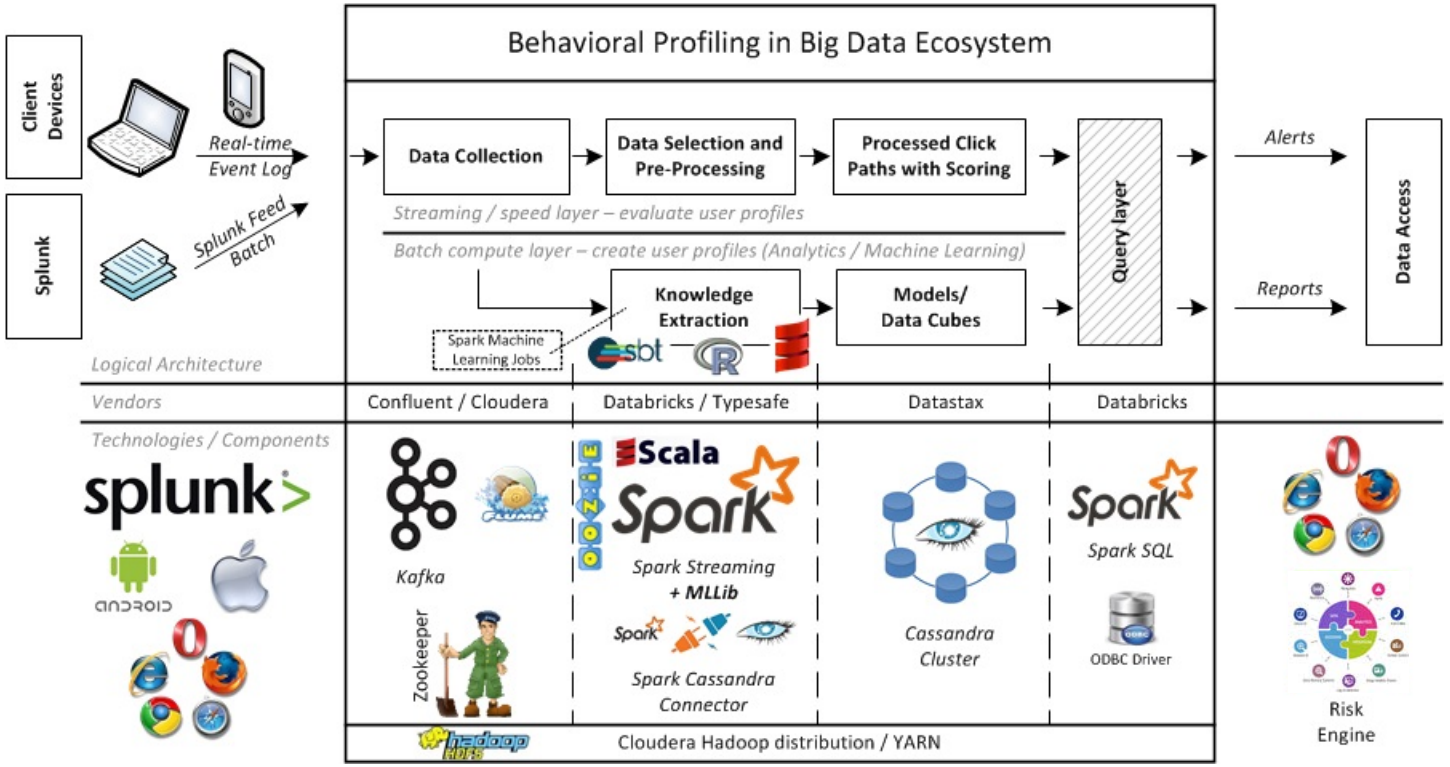
(5) R GUI for user session assessment

Vesting period of 5 days until a "normal" session is used for model update. "Normal" sessions from past 6 months are in scope for model update.

Target Production Design (Big Data Ecosystem)

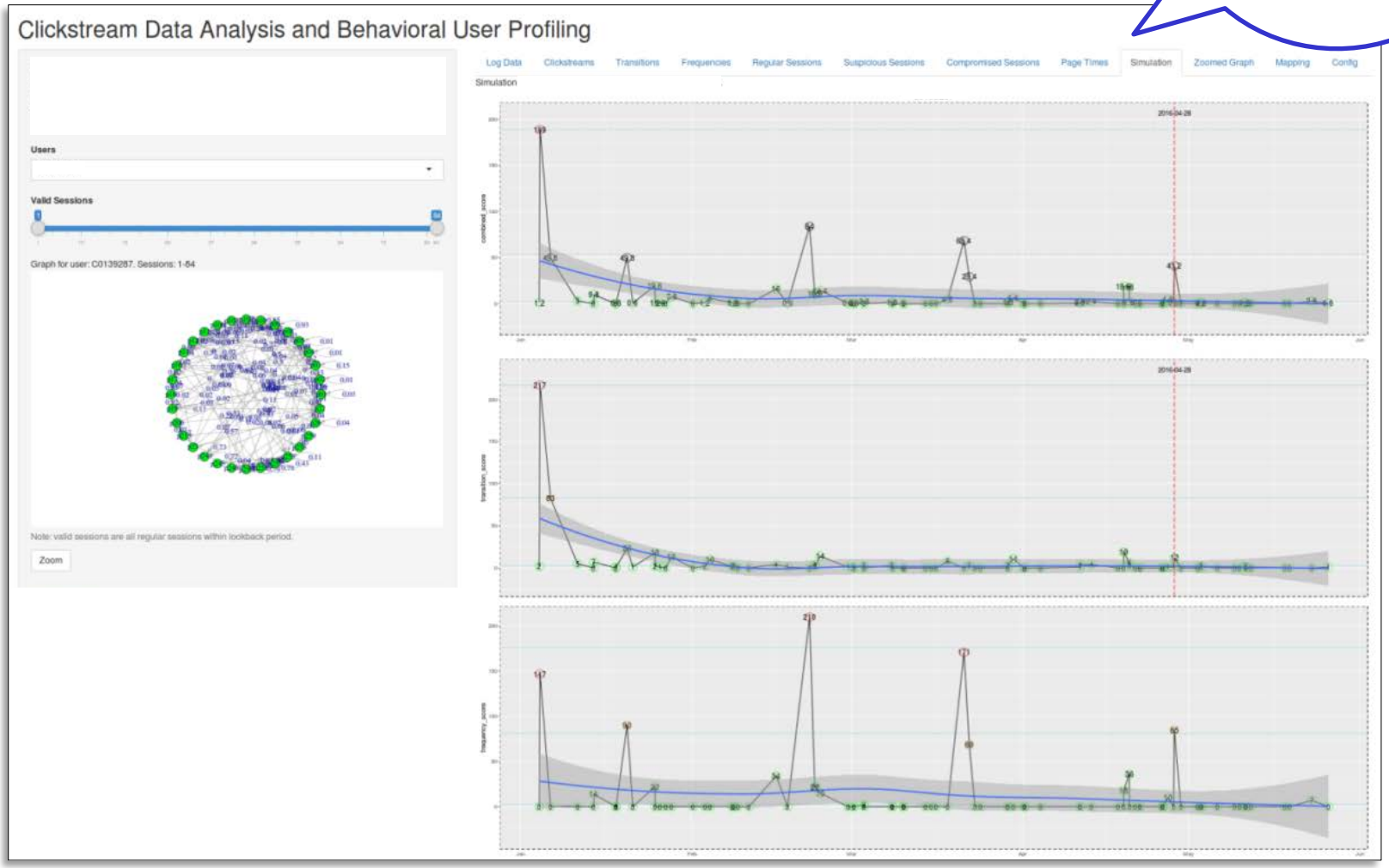
+ easily scales vertically and horizontally
+ high availability or fault-tolerarance by the core design of components
+ suits for near-realtime processing (micro jobs run in 500 ms time-interval)
+ utilizes well-integrated components from big data ecosystem
+ Spark provides powerful REPL environment for debugging
+ can be deployed on a cloud, like Amazon, Azure or Google

- requires knowledge of Java, Scala, R or Python for development
- runs stable only on Linux systems, though Windows is support
- packaging and deployment process is required
- “on-the-fly” changes cannot be implemented easily
- no referential integrity in NoSQL databases
- initial setup is quite time-consuming

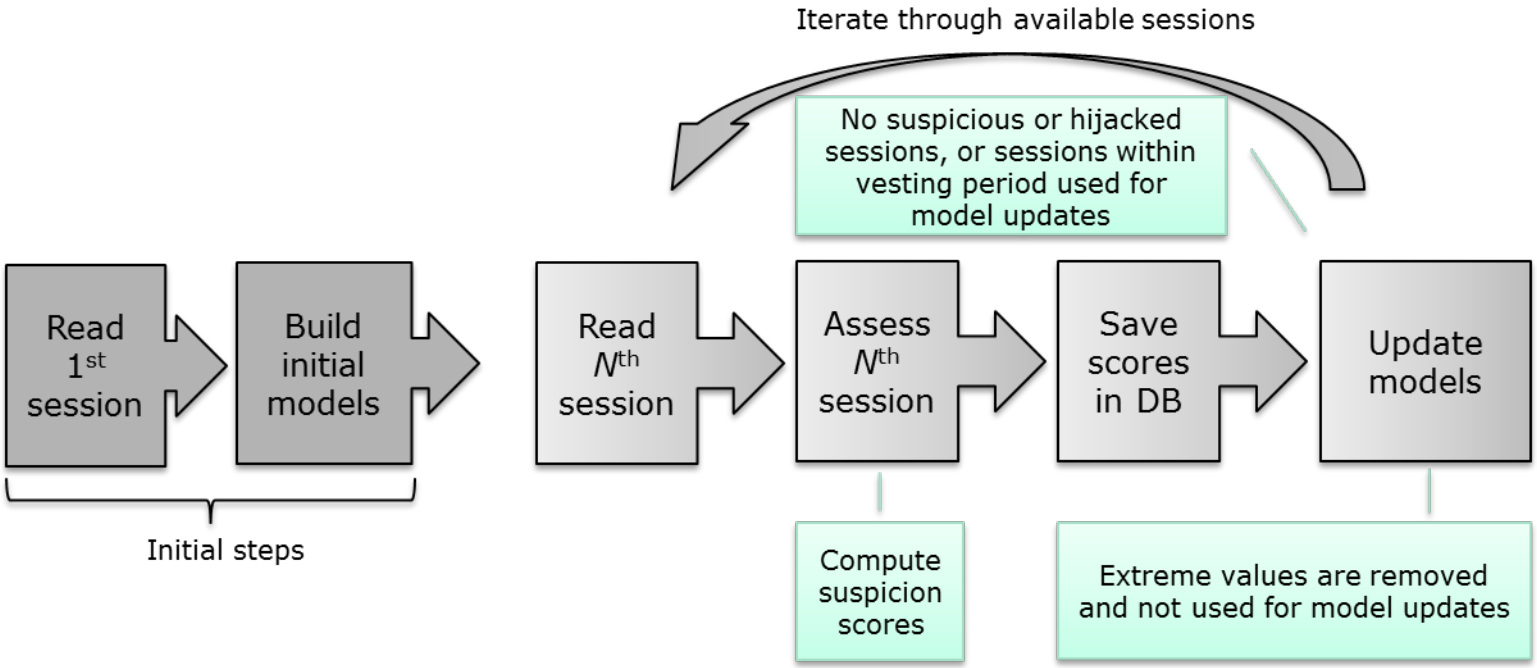


Shiny App (GUI)

How to visualize results?

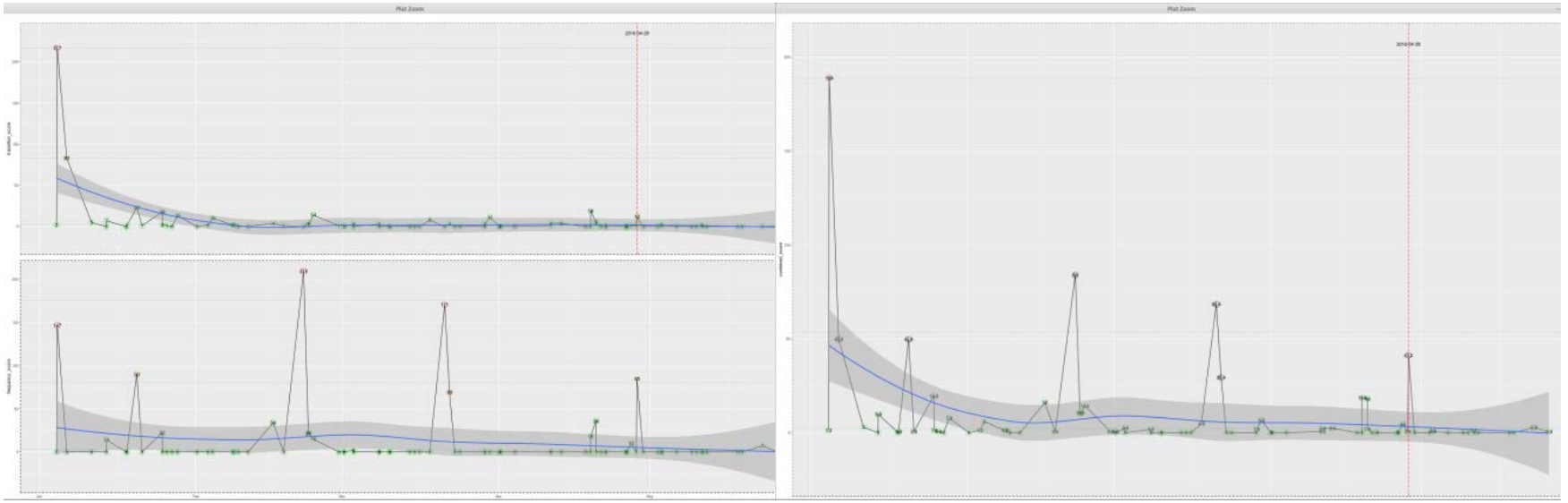


PoC - Experiments



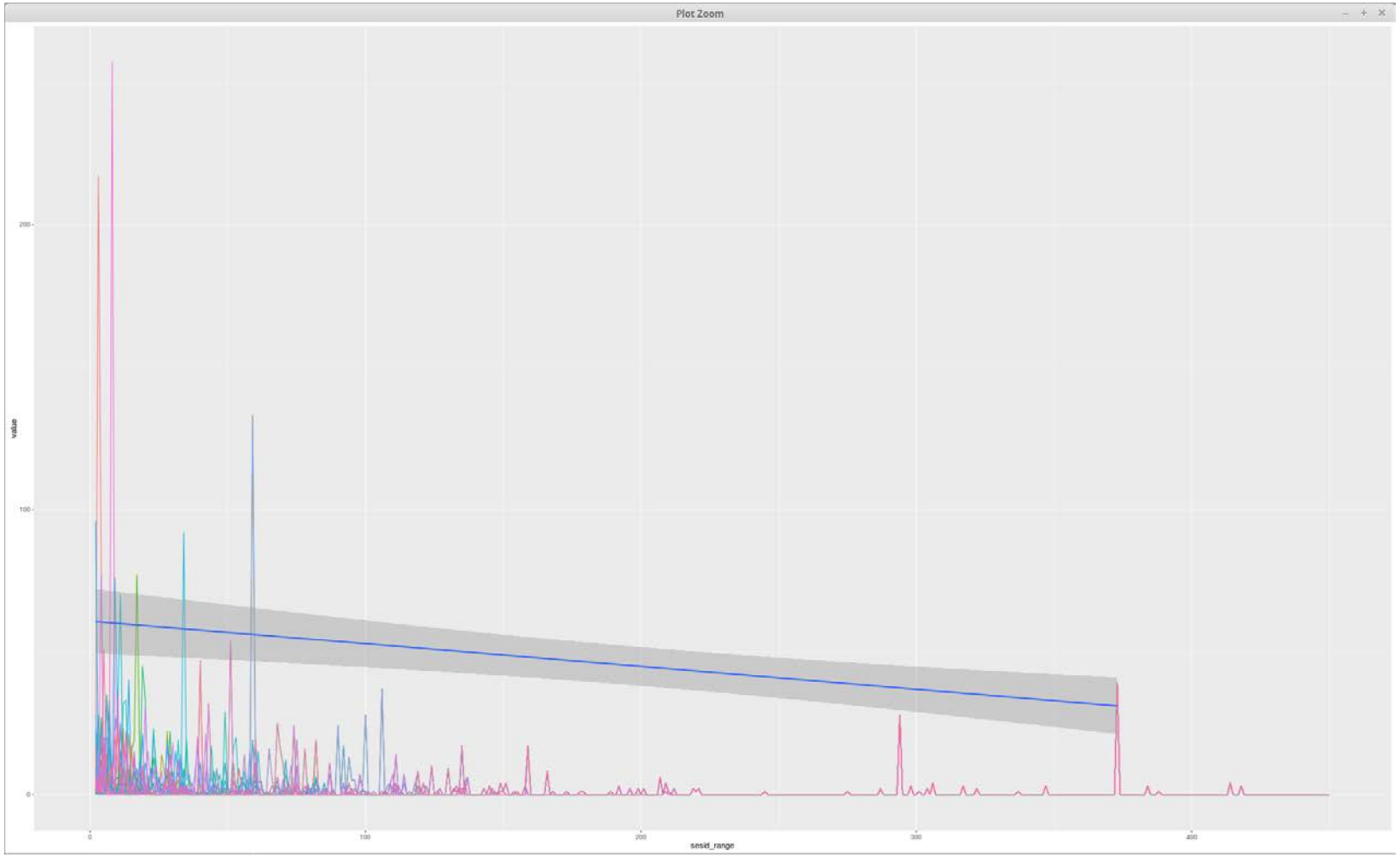
Test Results for a Sample User

Compromised session marked with a red dotted line.

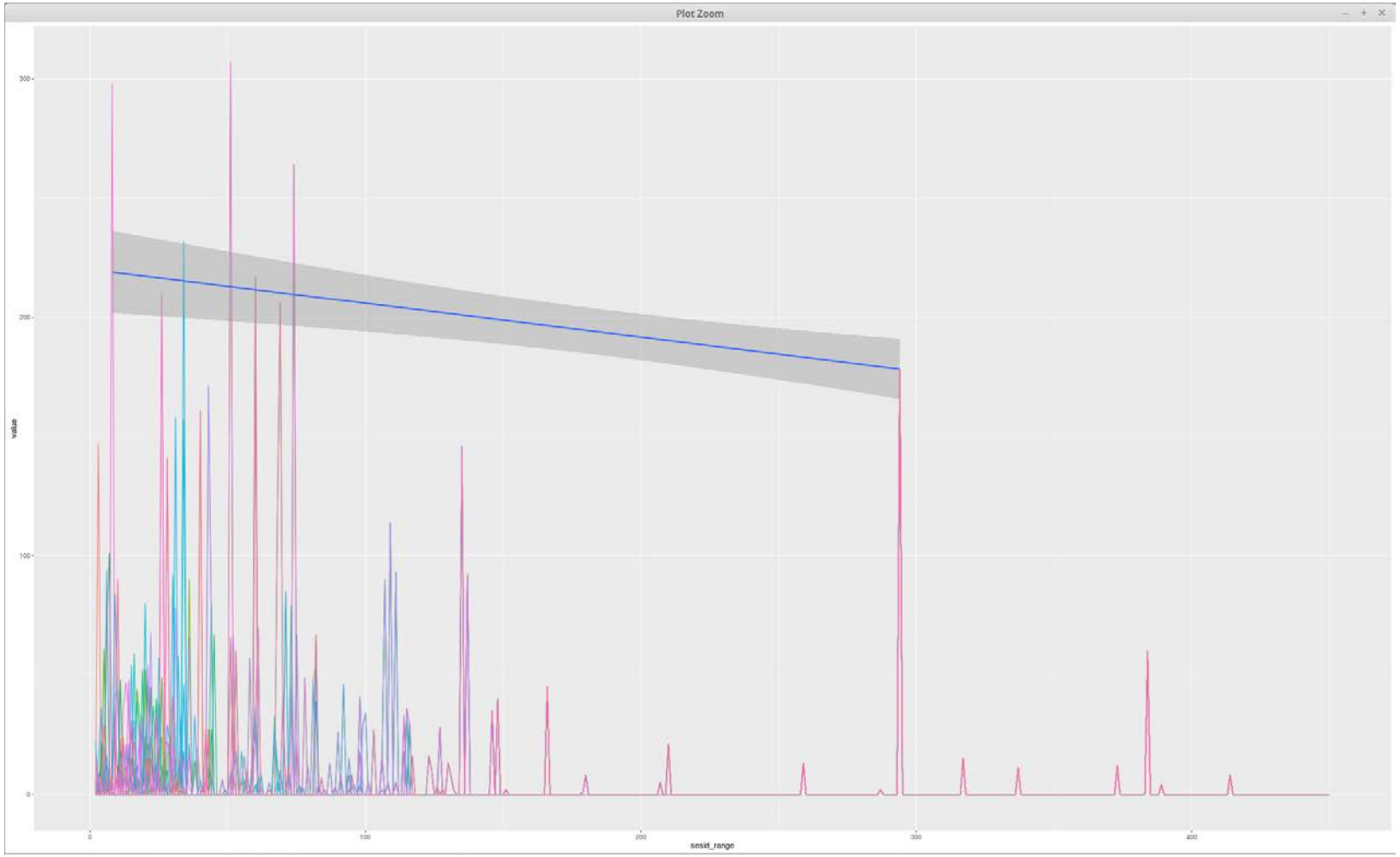


Two graphs on the left show development of transition (graph on the top) and frequency (graph on the bottom) scores. The graph on the right combines values from both, transition and frequency scores.

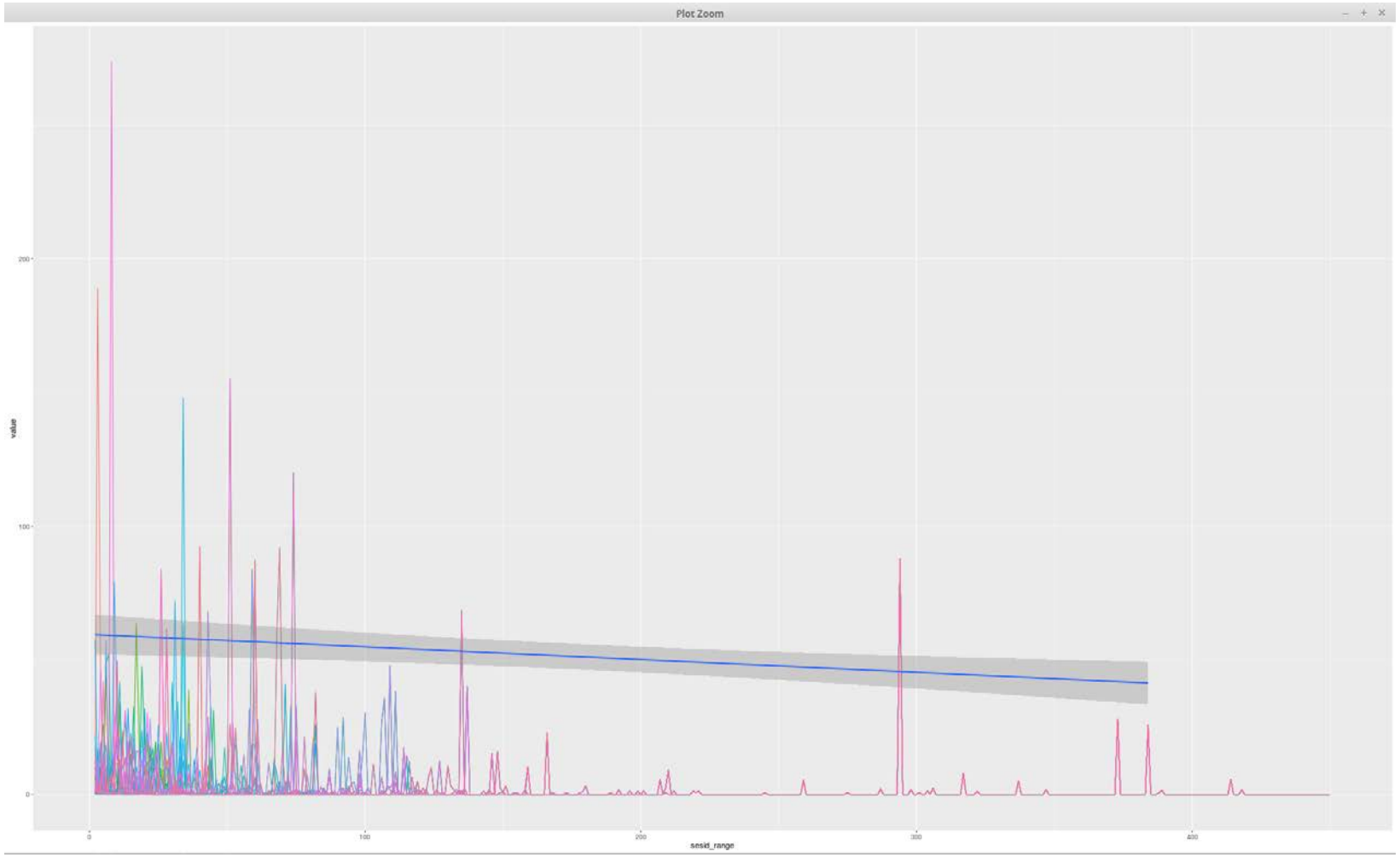
Transition Scores for All Tested Users



Frequency Scores for All Tested Users



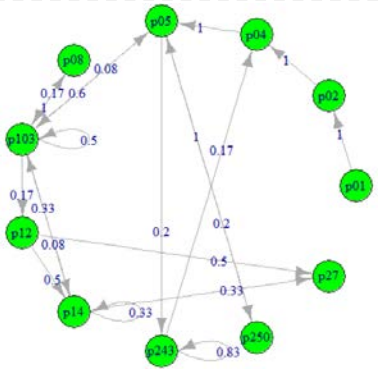
Combined Scores for All Tested Users



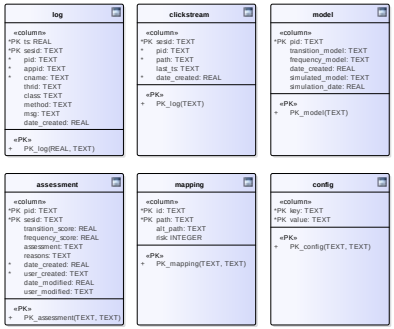
Summary

Suggested approach against MitB attacks to prevent financial fraud is to analyze the user behavior on a website, create mathematical models based on clickstream data and use these models to find anomalies or outliers in user behavior in order to prevent unauthorized payments before they take place.

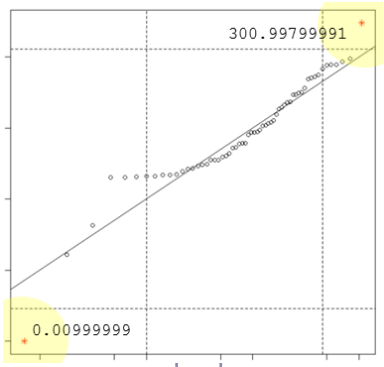
Clickstream



Persistence



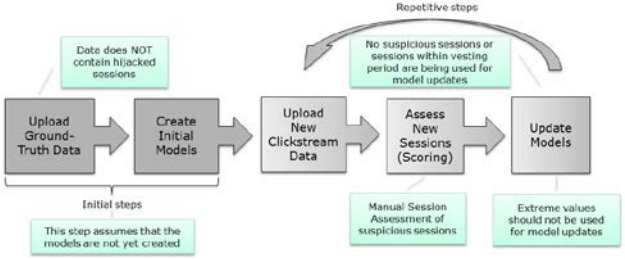
Outlier Detection



Models

	p01	p02	p04	p05	p08	p103	p12	p14	p243	p250
p01										
p02	1									
p04		1							0.17	
p05			1			0.08				1
p08						0.17				
p103				0.6	1	0.50		0.3		
p12						0.17				
p14						0.08	0.5	0.3		
p243				0.2					0.83	
p250				0.2						
p27							0.5	0.3		

Model Update Process



Scoring