Research Challenges for Credit Card Fraud Detection Systems

Computational Intelligence applications to Credit-Card Fraud Detection

Giacomo Boracchi

November, 20, 2015

giacomo.boracchi@polimi.it

Introduction

This talk focuses on a research collaboration with Universitè Libre de Bruxelles and Atos Wordline S.p.A

Reference Paper:

Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy

Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi and Gianluca Bontempi , *IEEE Transactions on Neural Networks and Learning Systems --14 pages, 2017*

Introduction

Today's talk focus: these systems use **data-driven models** (classifiers) in a real-world environment, where:

- processes are nonstationary
- training sets are biased and unbalanced

I will first describe the applications and how these systems work, and then I will illustrate the main research challenges

Introduction

Today's talk focus: these systems use **data-driven models** (classifiers) in a real-world environment, where:

- processes are nonstationary
- training sets are biased and unbalanced

I will first describe the applications and how these systems work, and then I will illustrate the main research challenges

Let's make this talk as interactive as possible, I'll be glad to answer your questions during the talk (if I am allowed to)

CREDIT CARD FRAUD DETECTION

A collaboration with…

Since November 2014 we started a collaboration with

- The Machine Learning Group at Université Libre de Bruxelles, Belgium (Prof. Gianluca Bontempi).
- Atos Wordline, a Belgium company that analyses about 600K credit card transactions everyday

ULB

worldline
e-payment services

The Fraud Detection System (FDS)

The Fraud Detection System (FDS):

- Performs security controls to prevent frauds
- Automatically analyzes all the authorized transactions and **alerts** the most suspicious ones
- Involves investigators that check the alerts and possibly block fraudulent cards

Fraud detection is challenging because:

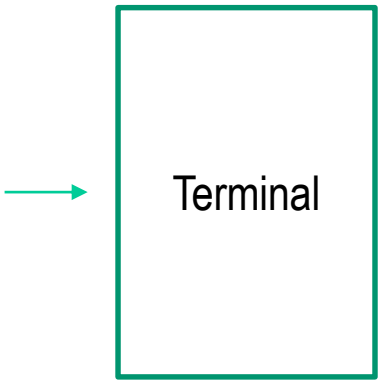
- New fraudulent strategies appear and genuine transactions might also change over time
- Genuine transactions far outnumber frauds (< 0.2%)
- Investigators that can actually check only few alerts

The **goal** of the project is to **improve** the precision of alerts automatically generated by the FDS

A CLOSER VIEW ON THE FDS

The levels of control in the Atos Worldline FDS

The Terminal



Purchase

The Terminal

Acceptance checks like:

- Correct PIN
- Number of attempts
- Card status (active, blocked)
- Card balance / availability

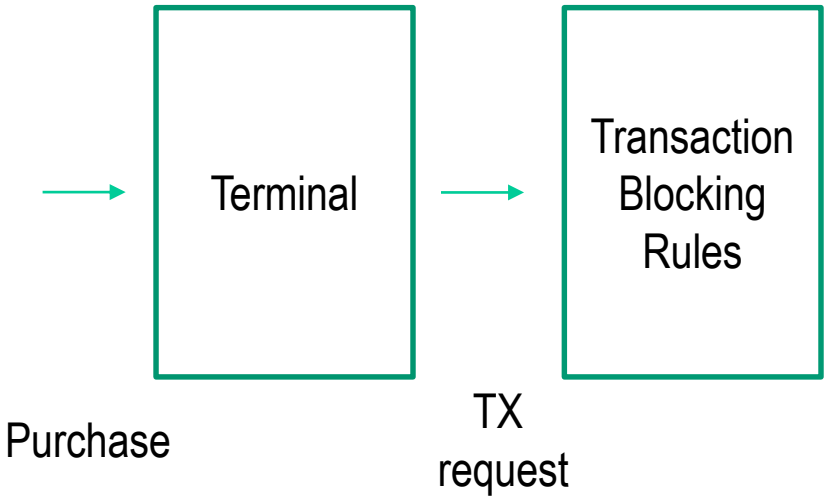
are immediately performed.

These checks are done in **real time**, and **preliminary filter** our purchases: when these checks are not satisfied, the card/transaction can be blocked.

Otherwise, a **transaction request** is entered in the system that include information of the actual purchase:

- *transaction amount, merchant id, location, transaction type, date time, …*

Summarizing



Transaction Blocking Rules

Association rules (if-then-else statements) like*

IF Internet transactions AND compromised website THEN deny the transaction

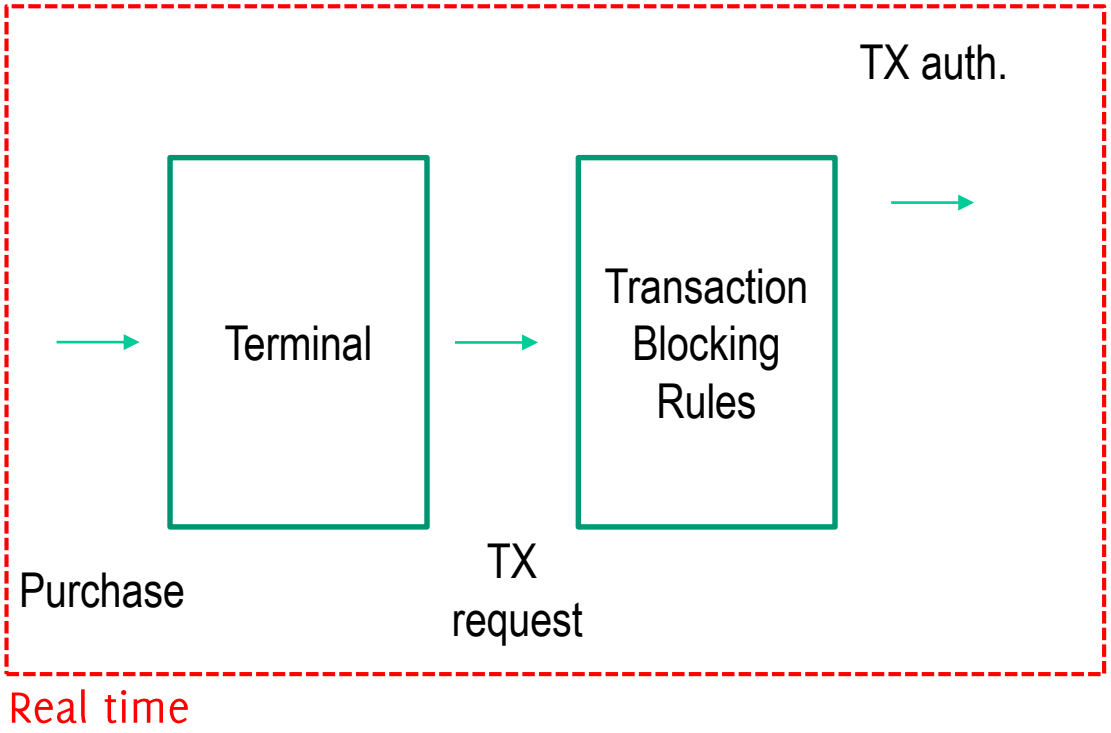
These rules:

- are **expert-driven**, designed by investigators
- are quite simple statement
- are easy to interpret
- have always «deny the transaction» as statement
- executed in real time

All the transaction RX passing these rules becomes **authorized transactions** and further analysed by the FDS

(*) Transaction blocking rules are confidential and this is just a reasonable example

Real Time Processing



Feature Augmentation

A feature vector $\boldsymbol{x}$ is associated to each authorized transaction.

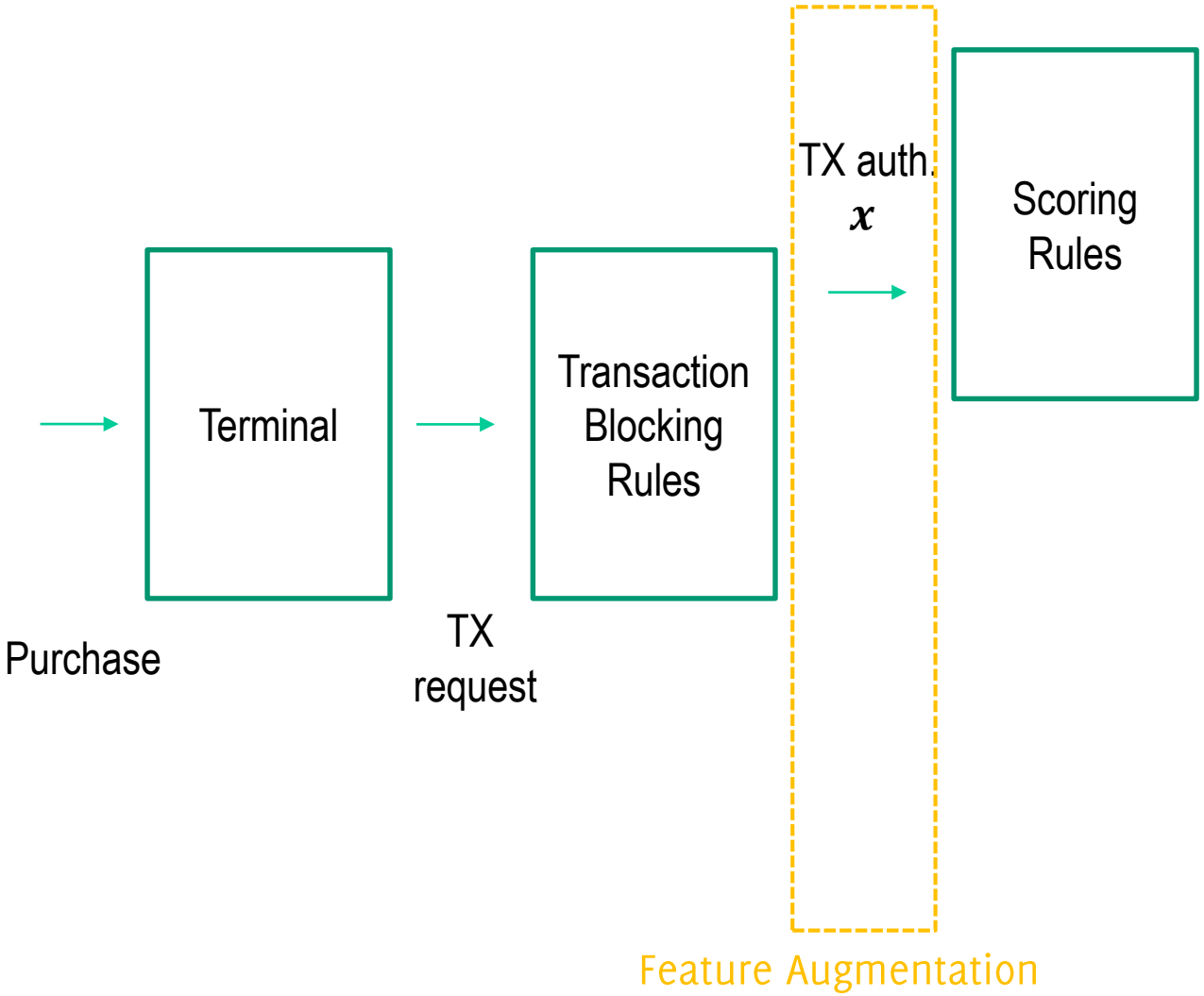
The components of $\boldsymbol{x}$ include data about the current transaction and customary shopping habits of the cardholder, e.g.:

- the average expenditure
- the average number of transactions per day
- the cardholder age
- the location of the last purchases
- …

and are very informative for fraud-detection purposes

Overall, about 40 features are extracted in near-real time.

Scoring Rules



Scoring Rules

Scoring rules are if-then-else statement that:

- Are **expert-driven**, designed by investigators.
- Operate on augmented features (components of $\boldsymbol{x}$)
- Assign a **score**: the larger the score the more risky the transaction (an estimate of the probability for $\boldsymbol{x}$ to be a fraud, according to investigator expertise)
- Feature vector receiving large scores are alerted
- Are easy to interpret and are designed by investigators
- Scoring rules operate in near-real time

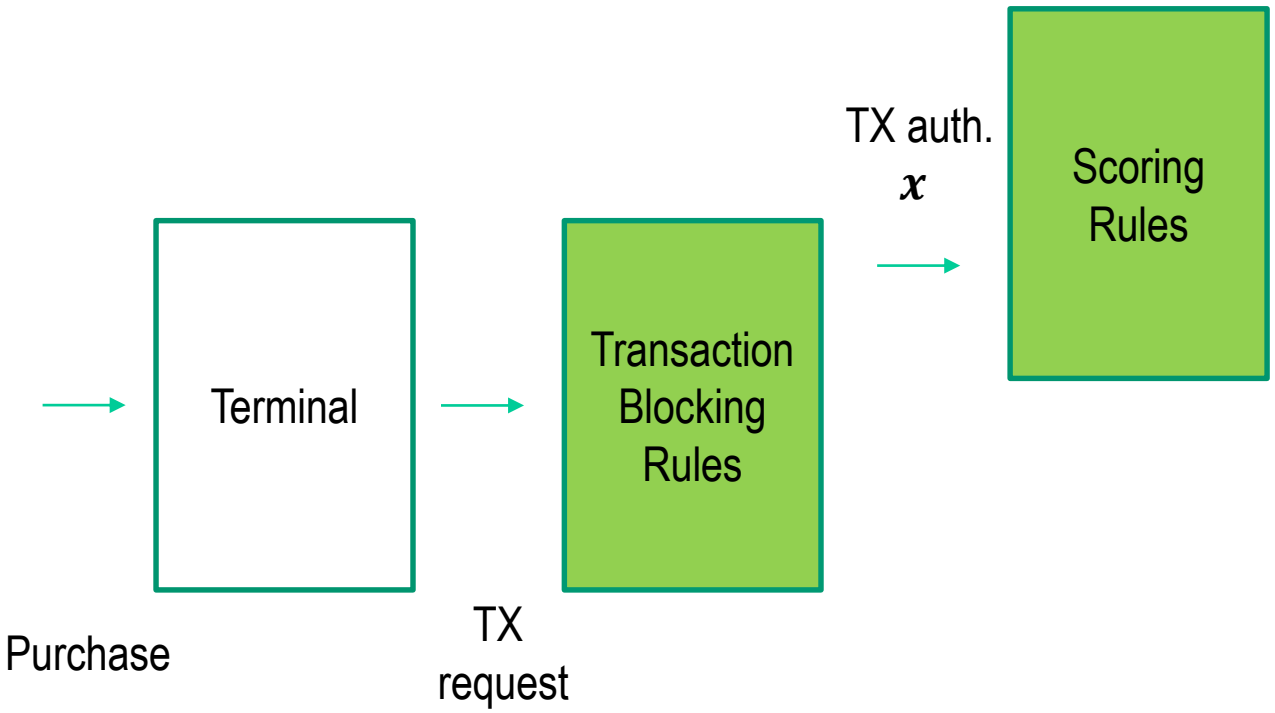
Scoring Rules

Examples* of scoring rules might be:

- *IF previous transaction in a different country AND less than 2 hours since the previous transaction, AND operation using PIN THEN fraud score = 0.95*
- *IF amount > average of transactions + 3σ AND country is a fiscal paradise AND customer travelling habits low THEN fraud score = 0.75*

(*) Scoring rules are confidential and these are just a reasonable examples

Expert Driven Models in the FDS



Expert-driven vs data-driven models

Scoring rules are an **expert-driven model**, thus:

- Can detect **well-known / reasonable** frauds
- Involve **few components** of the feature vector
- **Difficult** to **exploit correlation** among features

Expert-driven vs data-driven models

Scoring rules are an **expert-driven model**, thus:

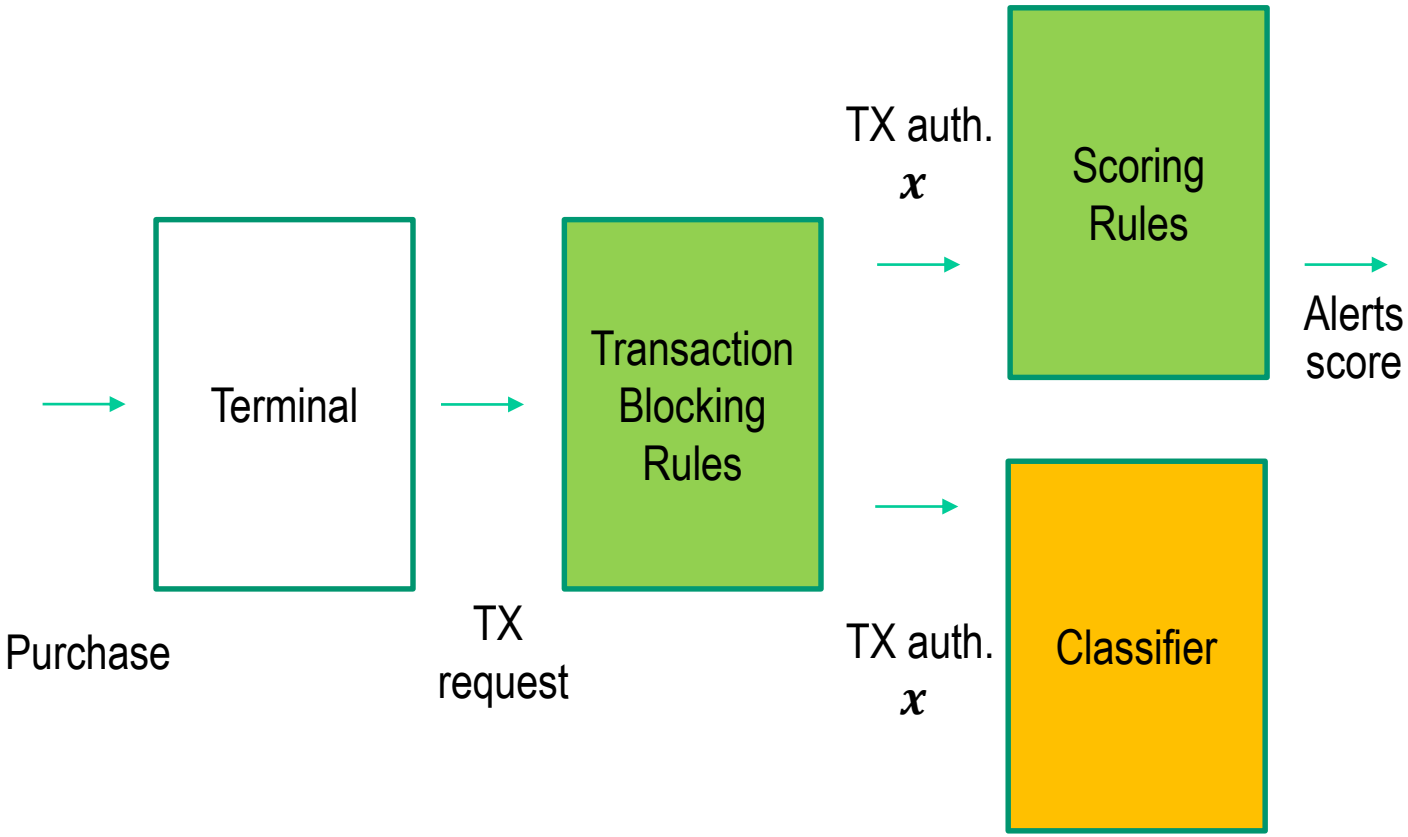
- Can detect **well-known / reasonable** frauds
- Involve **few components** of the feature vector
- **Difficult** to **exploit correlation** among features

Fraudulent patterns can be directly **learned from data**, by means of a **data-driven model** (DDM). This should:

- Simultaneously analyze **several components** of the feature vector
- Uncover **complex relations among features** that cannot be identified by investigator

.. as far as these are meaningful for separating frauds from genuine transactions

Alerts generation



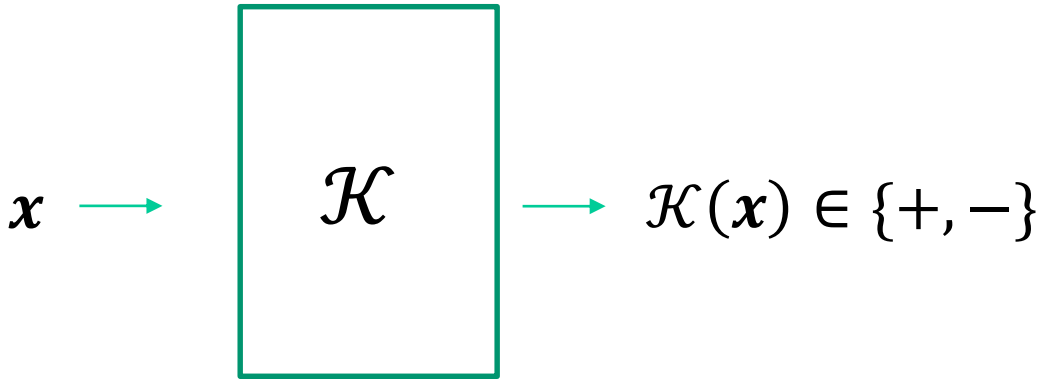
The Classifier in the FDS

A classifier $\mathcal{K}$ is learned from a **training set** that contains: **labeled feature vectors**

$$TR = \{(\boldsymbol{x}, y)_i, i = 1, \dots, N\}$$

where the label $y = \{+, -\}$, i.e., $\{«fraud», «genuine»\}$

In practice, the classifier $\mathcal{K}$ then can assign a label, $+ \; or \; -$ to each incoming feature vector $\boldsymbol{x}$



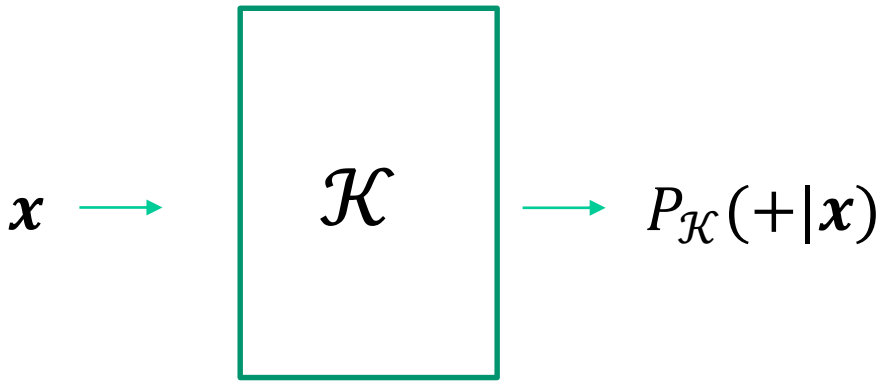
$\mathcal{K}$ considers transactions labeled as '+' as frauds

Alerts Reported to Investigators

It is not feasible to alert all transactions labeled as frauds

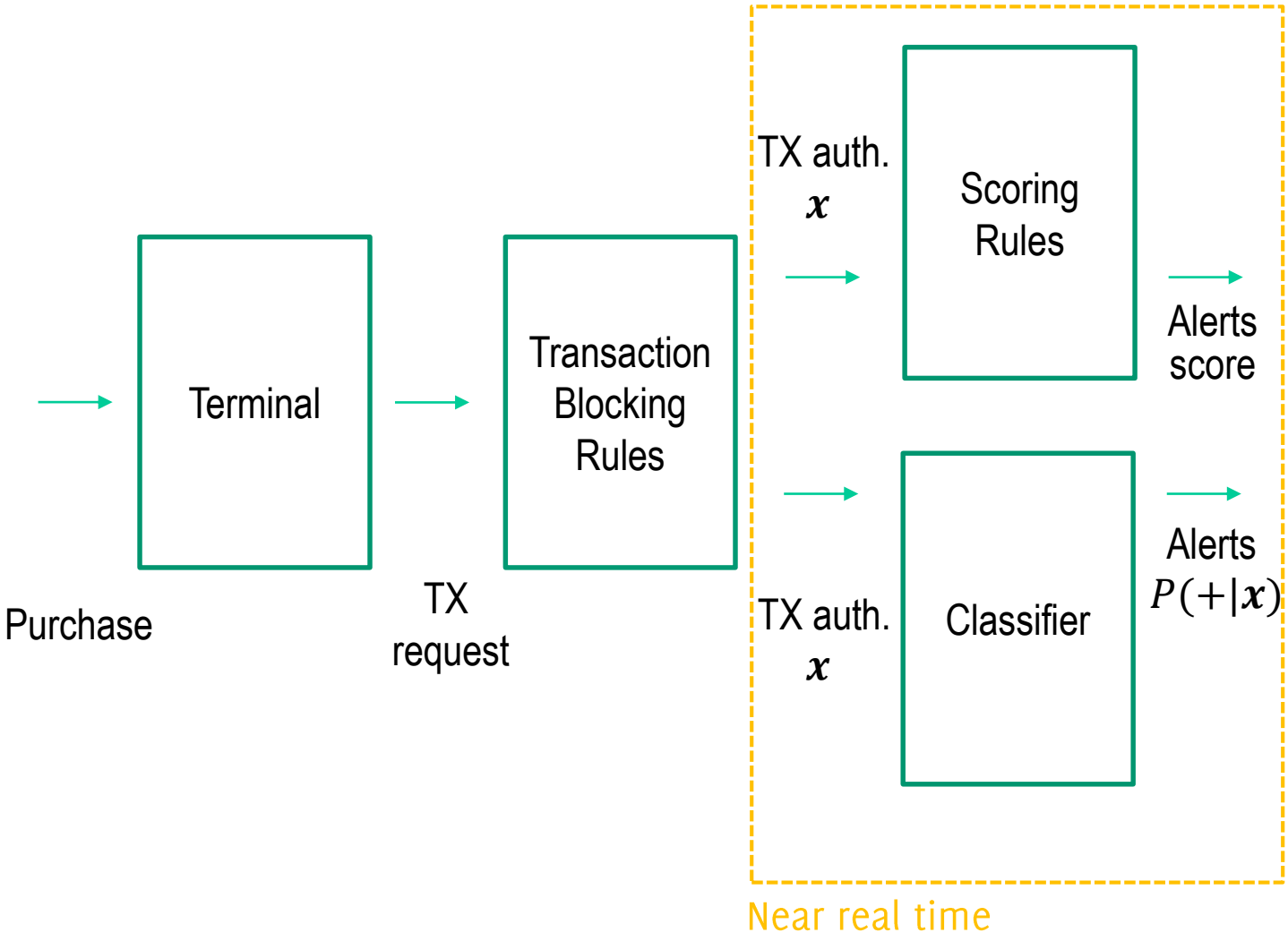
Only few transactions that are **very likely** to be frauds can be alerted.

Thus, the FDS typically consider $P_{\mathcal{K}}(+|\boldsymbol{x})$, **an estimate of the probability** for $\boldsymbol{x}$ to be a fraud according to $\mathcal{K}$

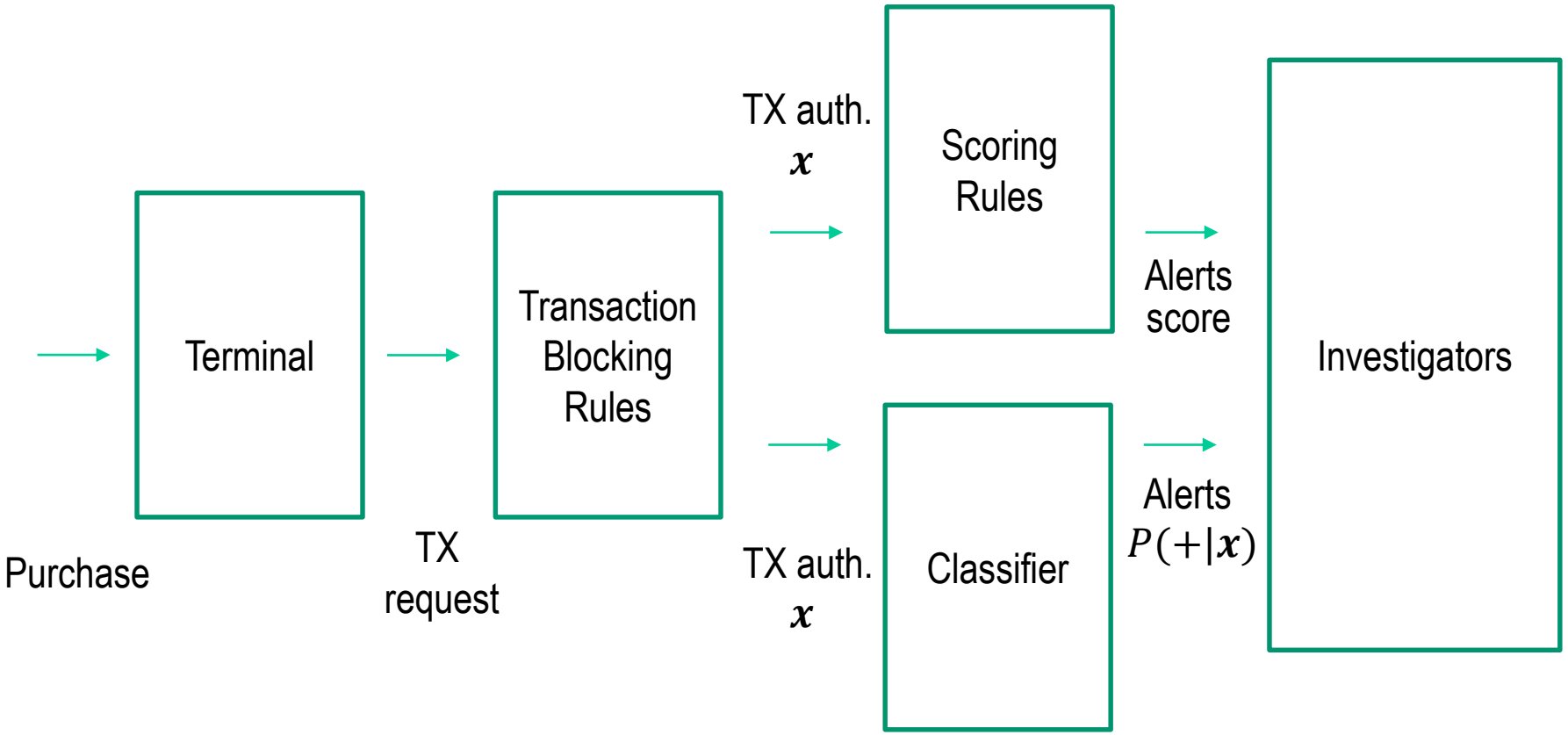


and only transactions yielding $P_{\mathcal{K}}(+|\boldsymbol{x}) \approx 1$ raise an alert

Near real time processing



Investigators



Investigators

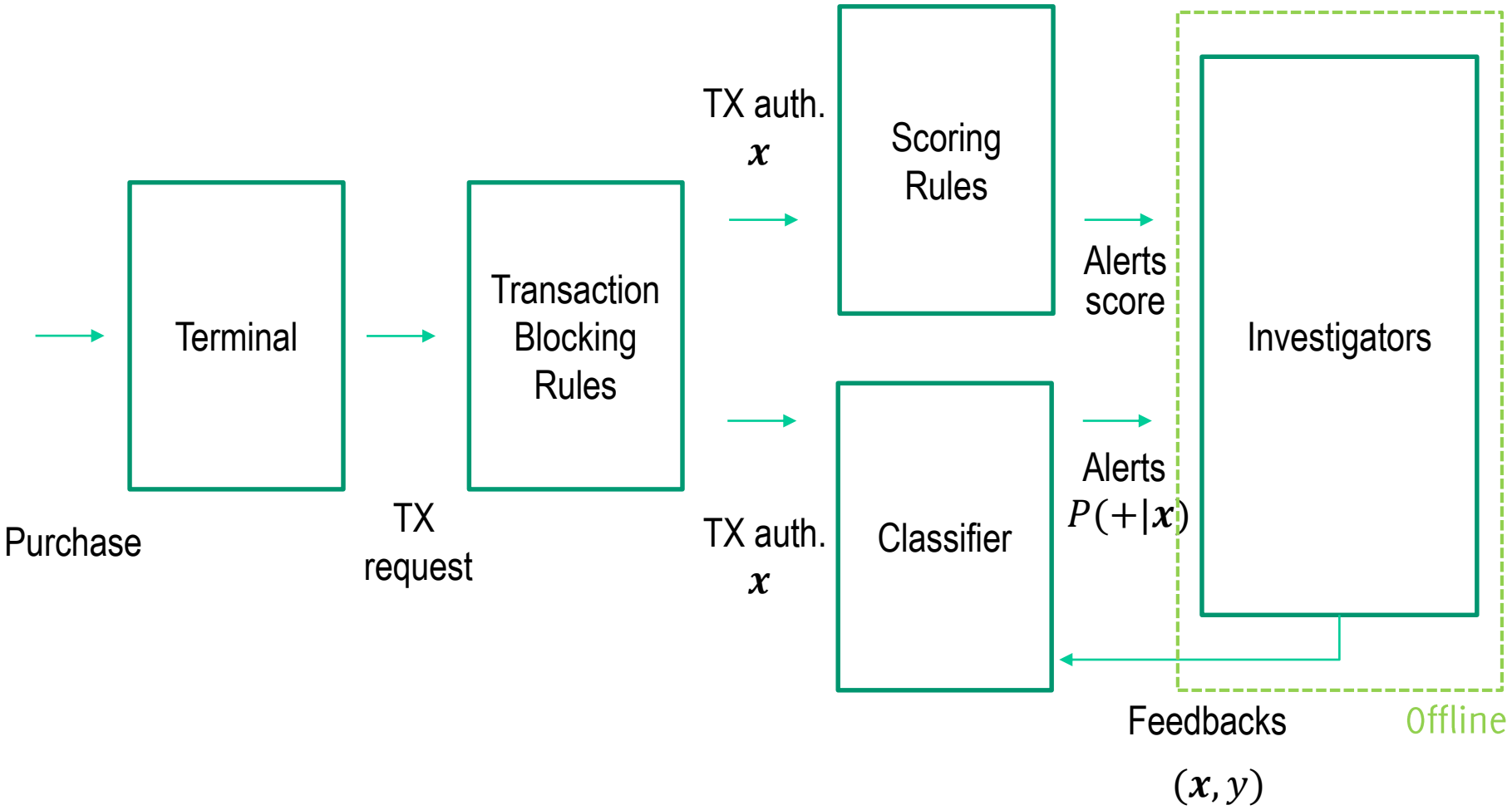
Investigators are **professionals** that are experienced in analyzing credit card transactions:

- they **design blocking/scoring rules**
- they **call cardholders** to check whether alerts correspond to frauds
- as soon as they detect a fraud, they block the card
- they annotate the **true label** of checked transactions

The labels associated to transactions comes in the form of **feedbacks** and can be used to re-train/update $\mathcal{K}$

Given the limited number of investigators, the large number of transactions, the multiple sources of alerts, etc … it is important to provide **very precise alerts**

Offline Processing



RESEACH CHALLENGES

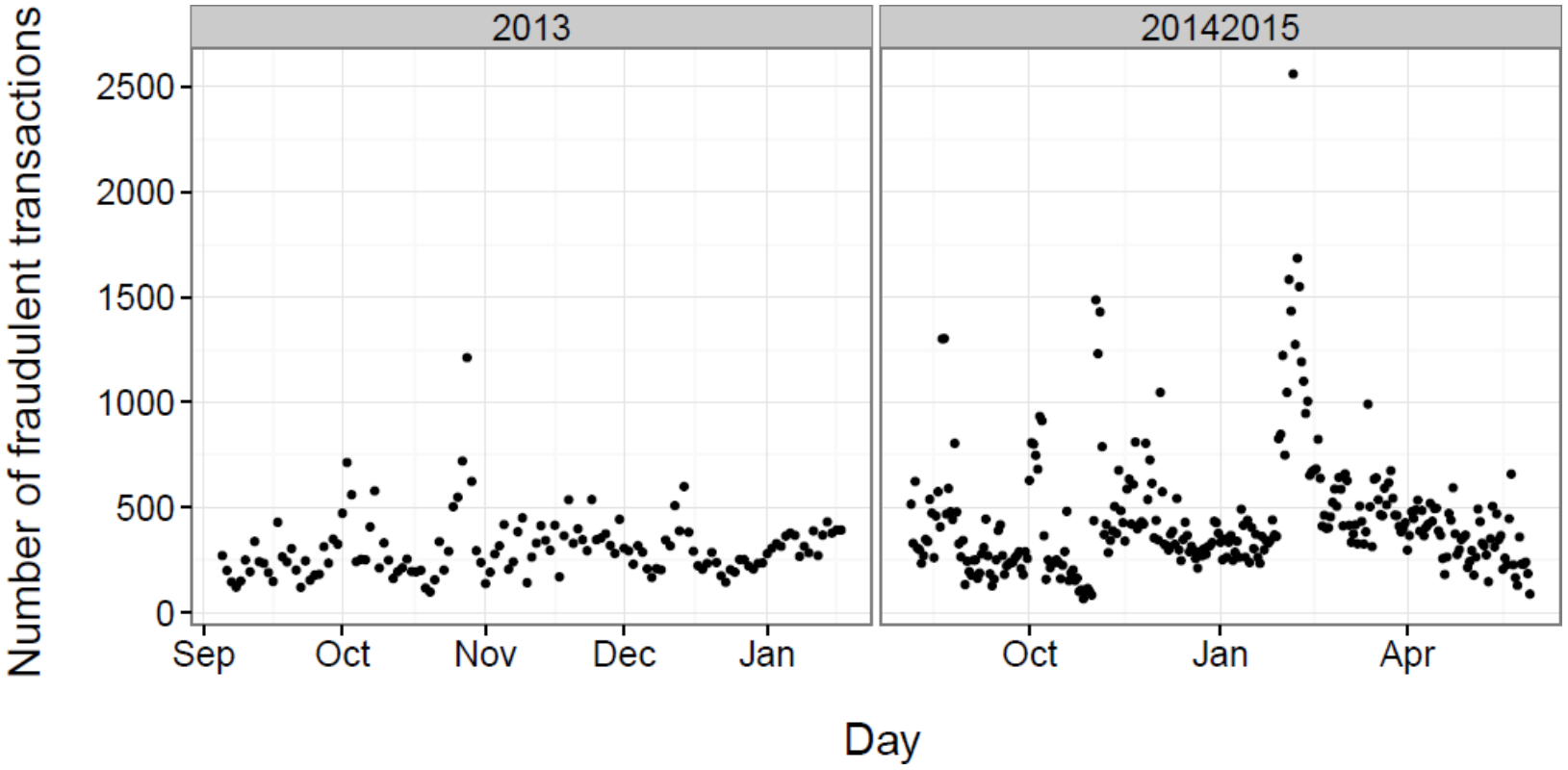
When using a Data-Driven Model in a FDS

First Research Challenges: Class Unbalance

In the real world, genuine transactions far outnumber the fraudulent ones.

The overall number of fraudulent transactions is less 0.2%

These are the statistics for our datasets



First Research Challenges: Class Unbalance

An unbalanced training set could lead $\mathcal{K}$ to consider all transactions as “genuine”, as this solution yield the smaller number of misclassified samples.

Main solutions:

- Resampling to balancing the proportion of classes
- Reweighting of training samples or cost-sensitive learning to assign different misclassification penalties

The best one also depends on the specific classifier in use.

In our experiments we used Random Forest (that are particularly effective in FDS) and can be easily combined with resampling methods.

Second Research Challenge: Concept Drift

In practice:

- Fraudsters constantly prepare new attacks
- Genuine purchases follow seasonality
- Everybody changes his own shopping habit over time

$\Rightarrow$ the process generating $\boldsymbol{x}$ is **nonstationary**

$$\boldsymbol{x} \sim \mathcal{X}_t$$

Concept Drift: a change in the data-generating process

The FDS become **obsolete** soon since:

- Expert-driven rules become inadequate and could not detect frauds or report to many false alerts
- $\mathcal{K}$ assumes $\boldsymbol{x}$ follow the same distribution of training data: when $\mathcal{X}$ changes, $\mathcal{K}$ becomes unfit

Adaptation in a FDS

Expert-driven rules are added/updated/removed by investigators according to the most recent trends

This is very important to timely :

- Include prior information in the FDS
- Detect specific (known) fraudulent patterns

In contrast, investigators **cannot manipulate** $\mathcal{K}$, which requires to be updated/retrained from data

Learning in Nonstationary Environment (NSE)

Learning methods for NSE is an important research topic in computational intelligence community

Two strategies for learning/adapting a DDM in a NSE

Learning in Nonstationary Environment (NSE)

Learning methods for NSE is an important research topic in computational intelligence community

Two strategies for learning/adapting a DDM in a NSE

- Active approach («Detect and React»)

Active approach **(our expertise)**:

- Monitor by a **change-detection test** the performance of $\mathcal{K}$ or distribution of $\boldsymbol{x}$ to **detect concept drift**
- After each detection **automatically** identifies suitable training data coherent with the current state of $\mathcal{X}$
- **Reconfigure** $\mathcal{K}$ only when a change is detected
- Provide information **when the change has occurred**

Learning in Nonstationary Environment (NSE)

Learning methods for NSE is an important research topic in computational intelligence community

Two strategies for learning/adapting a DDM in a NSE

- Active approach («Detect and React»)
- Passive approach (Continuous adaptation)

Passive approach:

- $\mathcal{K}$ is **steadily updated** on recent supervised samples
- No **change-detection information**

Learning in Nonstationary Environment (NSE)

Learning methods for NSE is an important research topic in computational intelligence community

Two strategies for learning/adapting a DDM in a NSE

- Active approach («Detect and React»)
- Passive approach (Continuous adaptation)

Which is the best depends on:

- Availability of supervised information,
- System resources
- Expected change rate/type
- Interest of having information about the change

Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

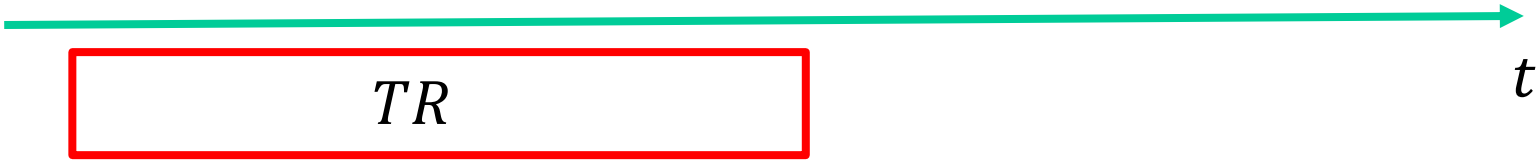
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days



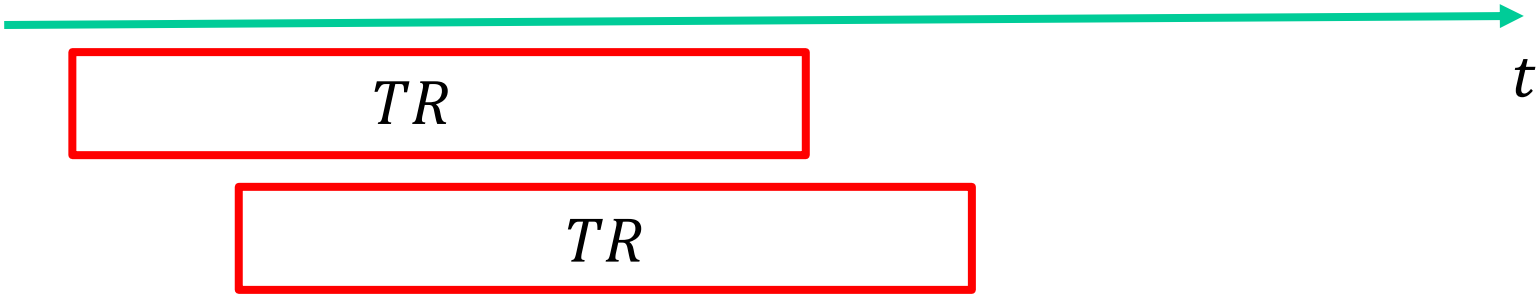
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days



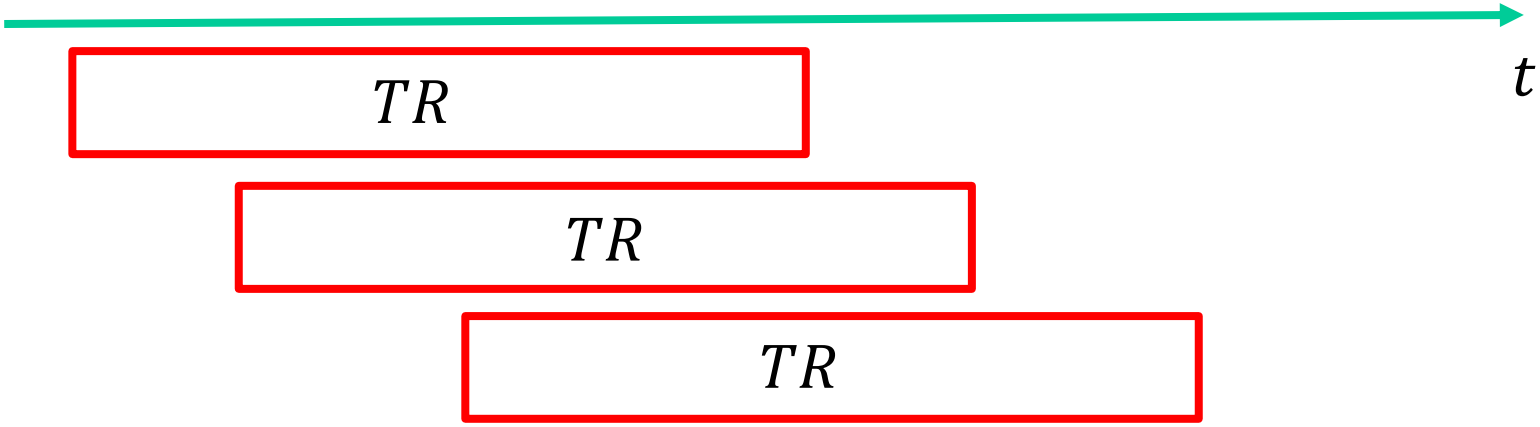
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days



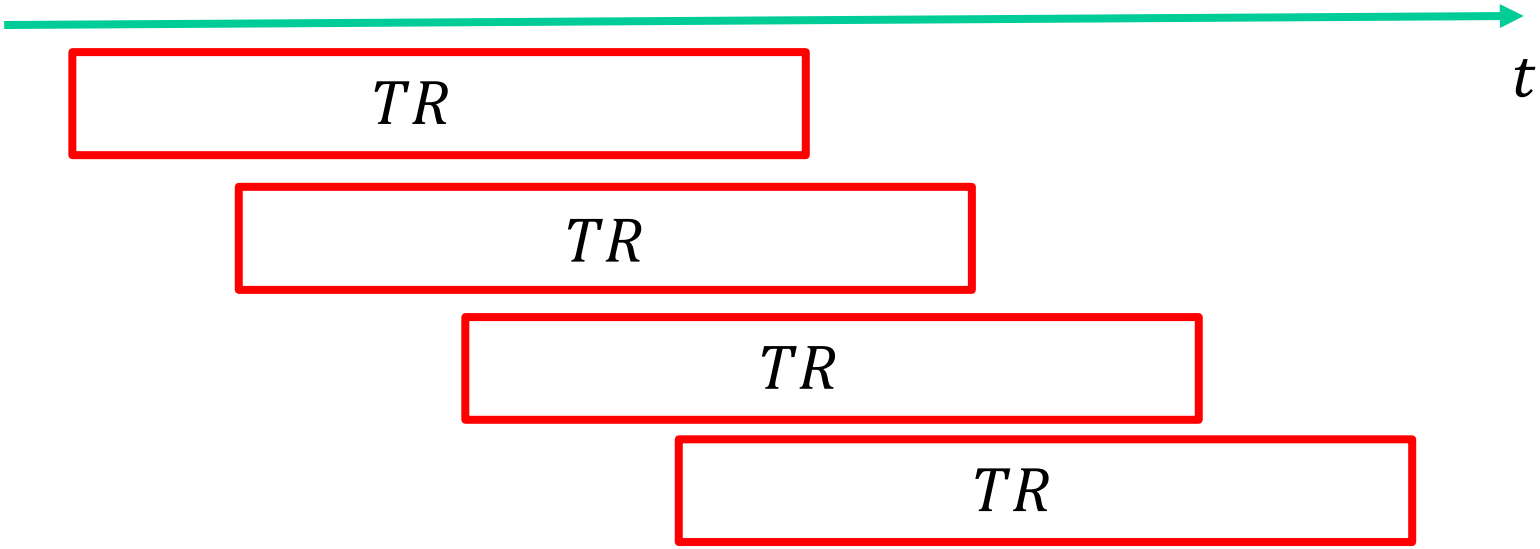
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Sliding Window Approach: use supervised information from the last δ days



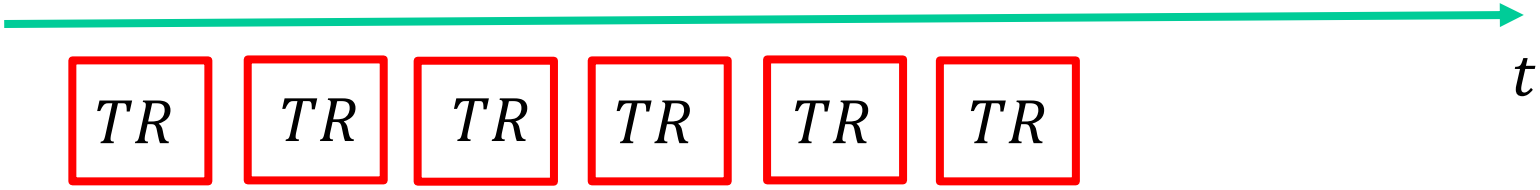
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Ensemble Approach: train a different classifier on each day and then aggregate their outputs



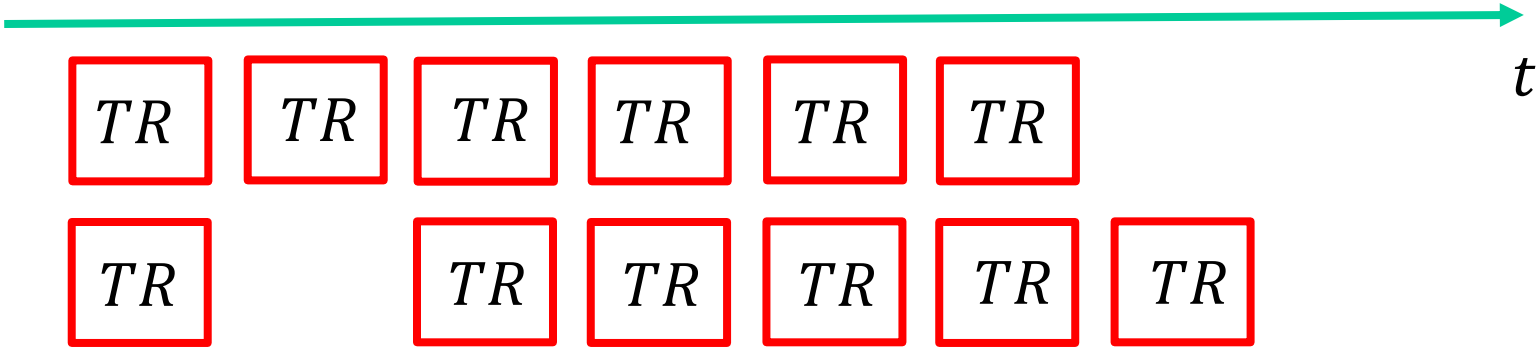
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

Ensemble Approach: train a different classifier on each day and then aggregate their outputs



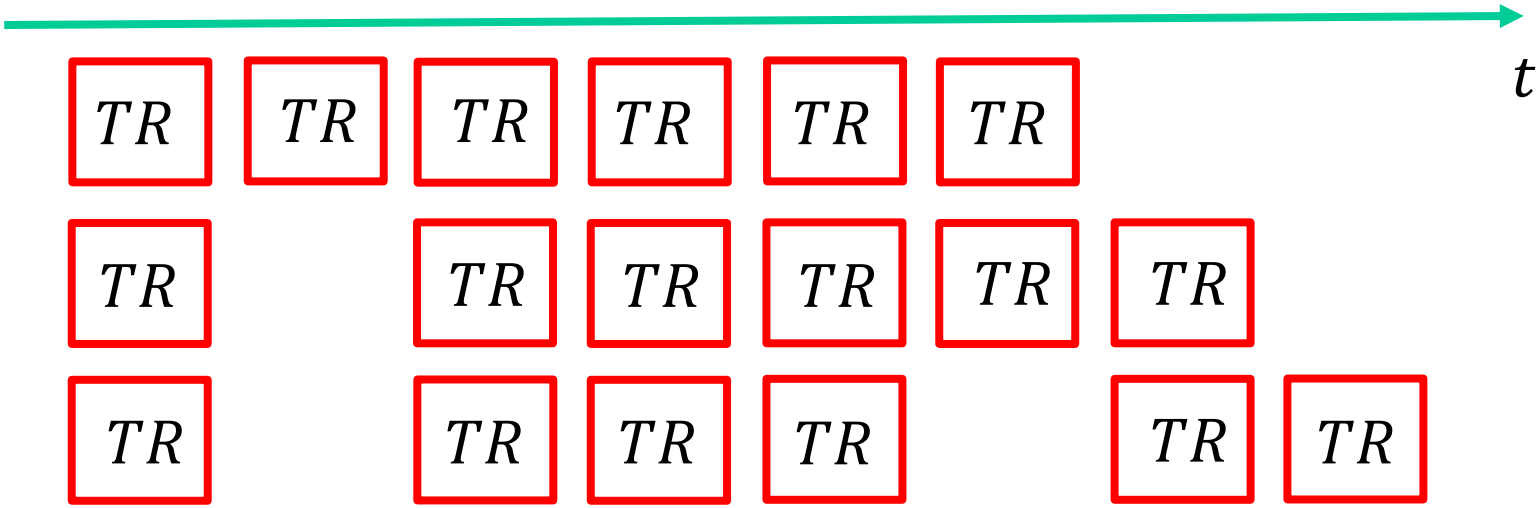
Adaptation in a FDS

In a FDS, the passive approach is the most suited.

In practice, $\mathcal{K}$ is **updated**, as soon as **enough** supervised samples are gathered.

Then, $\mathcal{K}$ becomes $\mathcal{K}_t$ and is updated (say) everyday

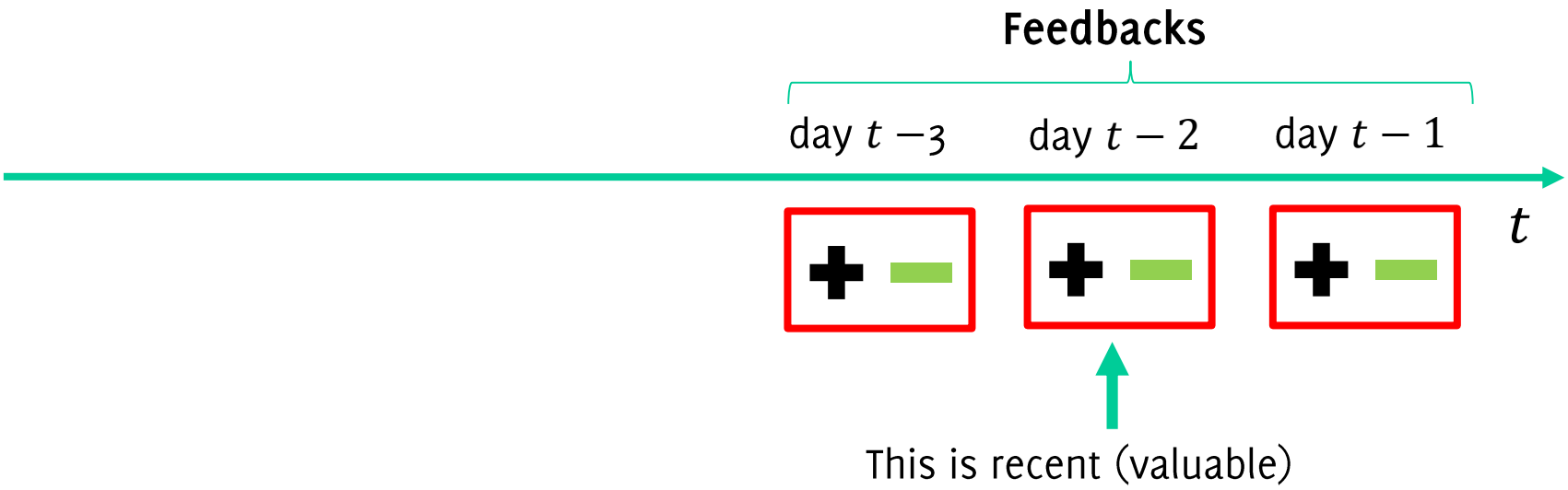
Ensemble Approach: train a different classifier on each day and then aggregate their outputs



Supervised Information in a FDS

The supervised information available in the FDS is:

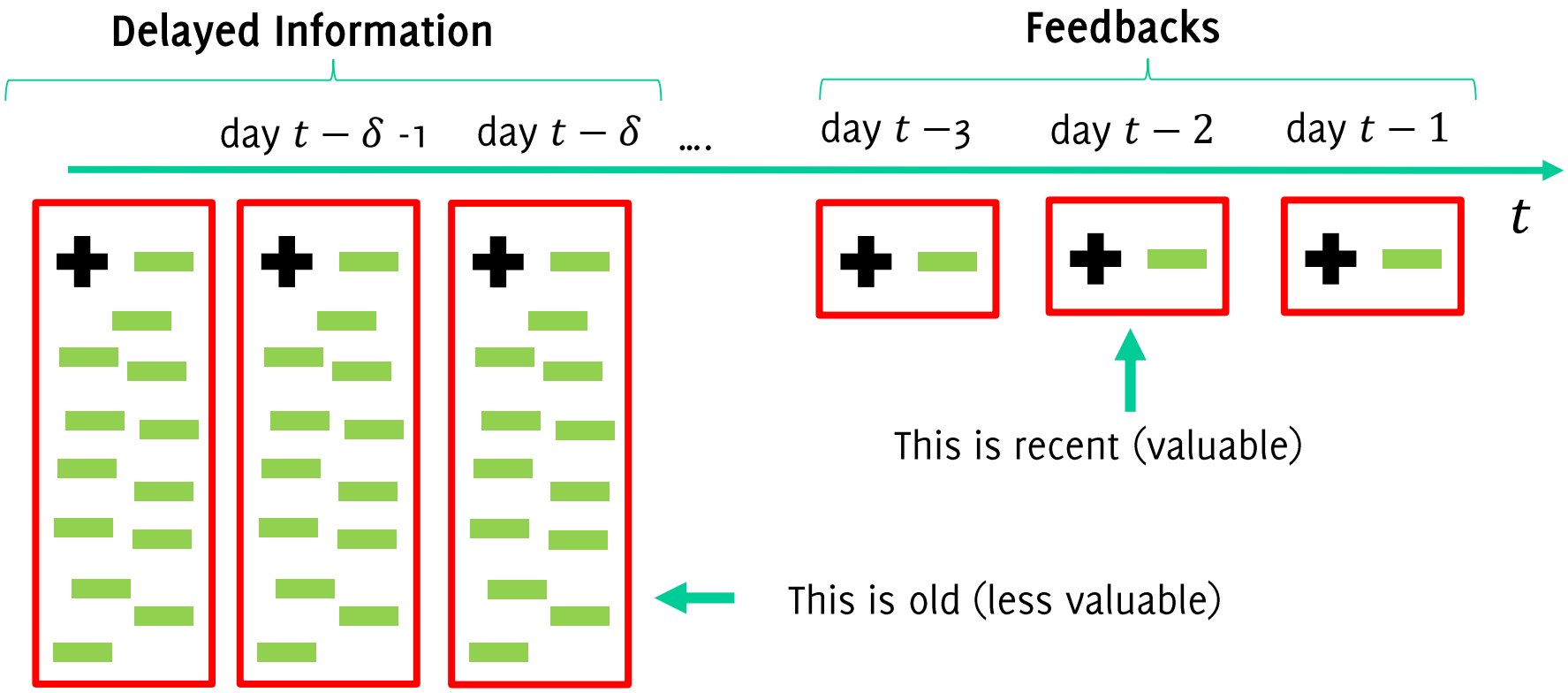
- **Few, very recent feedbacks** of yesterday's alert



Supervised Information in a FDS

The supervised information available in the FDS is:

- **Few, very recent feedbacks** of yesterday's alert
- **All the transactions** authorized **several days before** (verification latency)

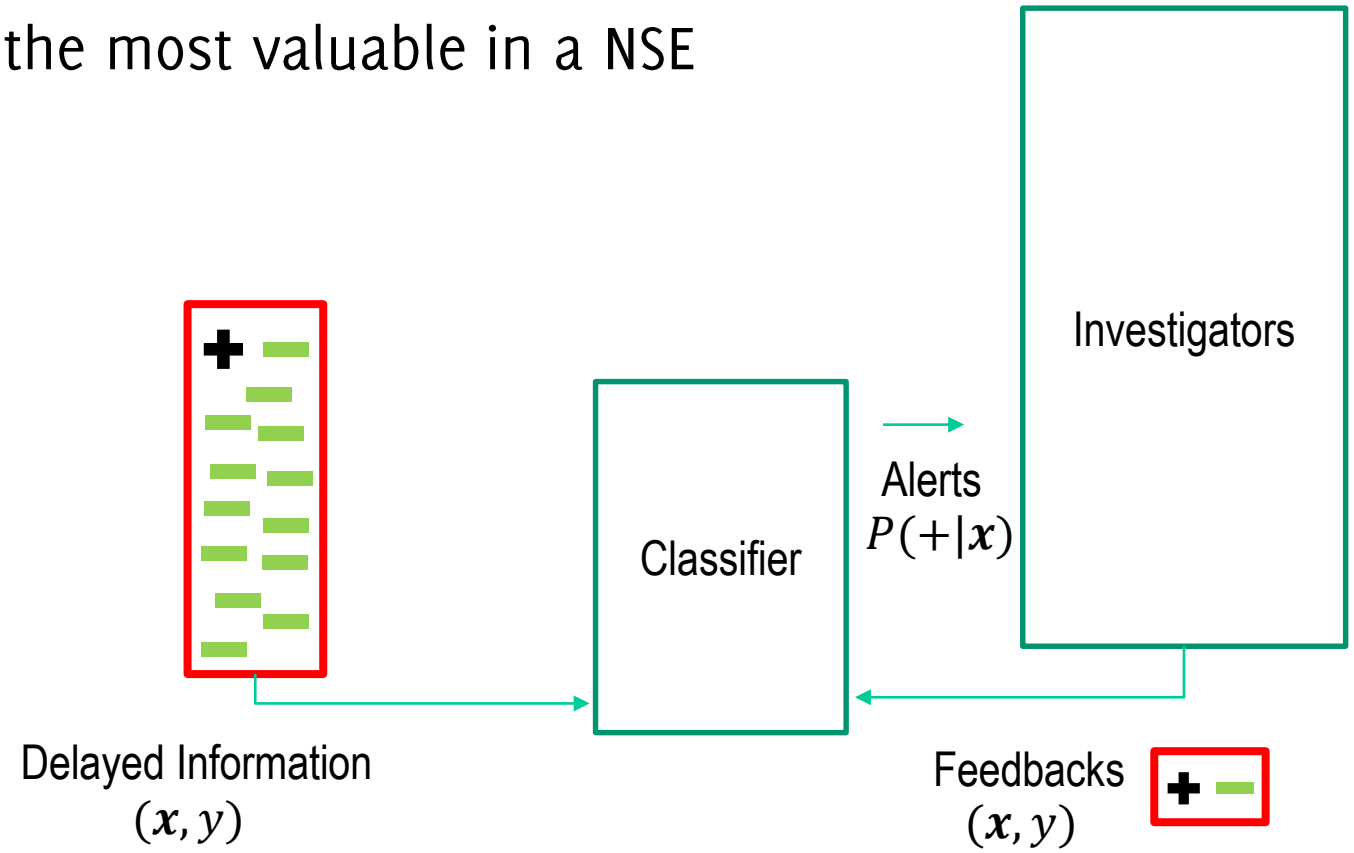


Third Research Challenge: Sample Selection Bias

The only recent supervised information is provided by the «alert-feedback» interaction

Feedbacks are somehow selected by $\mathcal{K}$ itself

Feedbacks are the most valuable in a NSE



Third Research Challenge: Sample Selection Bias

Why are feedbacks and delayed samples different?

- They have different class proportions
- Feedbacks are only highly suspicious transactions
- Feedbacks are more recent than the others

When training and testing distributions are different there is a **sample selection bias**

Main solutions in to correct sample selection bias

- Importance weighting
- Ensemble methods using unsupervised samples

An Effective Solution to Fraud Detection

“Feedback and delayed samples are different in nature and should be exploited differently”

An Effective Solution to Fraud Detection

"Feedback and delayed samples are different in nature and should be exploited differently"

Learn two separate classifiers from:

- Feedback (get a classifier $\mathcal{F}$)
- Delayed Samples (get a classifier $\mathcal{D}$)

Aggregate the outputs

$$P_{\mathcal{K}}(+|\boldsymbol{x}) = \alpha P_{\mathcal{F}}(+|\boldsymbol{x}) + (1-\alpha) P_{\mathcal{D}}(+|\boldsymbol{x})$$

An Effective Solution to Fraud Detection

"Feedback and delayed samples are different in nature and should be exploited differently"

Learn two separate classifiers from:

- Feedback (get a classifier $\mathcal{F}$)
- Delayed Samples (get a classifier $\mathcal{D}$)

Aggregate the outputs

$$P_{\mathcal{K}}(+|\boldsymbol{x}) = \alpha P_{\mathcal{F}}(+|\boldsymbol{x}) + (1-\alpha)\, P_{\mathcal{D}}(+|\boldsymbol{x})$$

Sliding Window

	Dataset 2013		Dataset 2014	
classifier	mean	sd	mean	sd
$\mathcal{F}$	0.609	0.250	0.596	0.249
$\mathcal{W}^D$	0.540	0.227	0.549	0.253
$\mathcal{W}$	0.563	0.233	0.559	0.256
$\mathcal{A}^W$	0.697	0.212	0.657	0.236

Ensembles

	Dataset 2013		Dataset 2014	
classifier	mean	sd	mean	sd
$\mathcal{F}$	0.603	0.258	0.596	0.271
$\mathcal{E}^D$	0.459	0.237	0.443	0.242
$\mathcal{E}$	0.555	0.239	0.516	0.252
$\mathcal{A}^E$	0.683	0.220	0.634	0.239

CONCLUDING REMARKS

Concluding Remarks

The majority of works addressing the fraud-detection problem in credit card transactions unrealistically assume that the class of each transaction is immediately provided for training the classifier.

Real-world FDSs address articulated classification problems which involve the following challenges:

- concept drift
- class unbalance
- sample selection bias due to alert-feedback interaction

Precision of the reported alerts is probably the most meaningful figure of merit in a real-world FDS.

Concluding Remarks

Our experiments on two vast datasets of real-world transactions show that, to get precise alerts, it is mandatory to assign larger importance to feedbacks during learning.

We propose an effective and novel learning strategy which separately trains classifiers on feedbacks and delayed supervised samples, and then aggregate their posteriors to identify alerts.

Further Information

Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy

Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi and Gianluca Bontempi , *IEEE Transactions on Neural Networks and Learning Systems --14 pages, 2017*