Unsupervised Anomaly Detection for Payment Systems at Transaction Level

Eduard Betz[1], Ioana Duca-Radu[2]
European Central Bank

Seminar on Quantitative Analysis of Financial Market Infrastructures
17th Payment and Settlement System Simulation Seminar
29-30 August 2019, Helsinki, Finland

29 August 2019

[1,2] eduard.betz@ecb.int, ioana.duca@ecb.int. European Central Bank, Sonnemannstraße 20, 60341 Frankfurt, Germany.

Motivation

Goal: Develop a framework to detect anomalous activity in payment systems at individual transaction level.

Several purposes:

- detect fraudulent payments (**our main area of interest**)
- reveal liquidity stress at a granular level
- complement existing monitoring activities

Challenges:

- high dimensionality
- presence of categorical attributes (e.g. BIC information)
- anomalous characteristics may be hidden in subspaces of the data
- lack of anomaly labels $\rightarrow$ need unsupervised framework

Context

Most methods focus on a two-step procedure:

1. **learning a pattern of normality** using large amounts of data
2. identifying anomalies as **deviations** from learned patterns

* e.g. replicator/autoencoder neural networks (Hawkins et al. (2002), Triepels et al. (2017), 1-class SVMs (Tax and Duin (2004))

Alternative: **isolation-based** anomaly detection:

- iteratively disect data space to isolate anomalies
- isolation forest (Liu et al. (2008), Liu et al. (2012))
- efficent, has been applied successfully to large data sets

* e.g. cloud data centers (Calheiros et al. (2017))

This paper

Contibutes to **isolation-based** anomaly detection literature by:

- applying the isolation forest to payments data at transaction level
- working on a mixed-type data set with categorical variables
- developing a model selection framework based on output stability

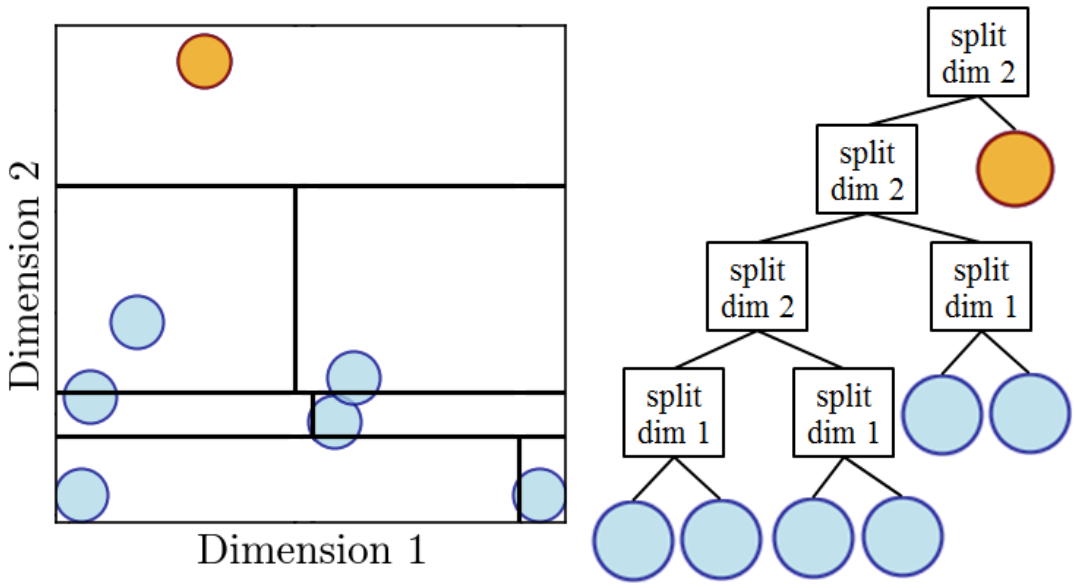
Overview

1. Motivation ✓
2. Isolation forest model
3. TARGET2 data
4. Methodology for training / evaluation
5. Model calibration
6. Model selection based on output stability
7. Conclusion

Isolation forest model

Idea: Anomalies are **few** and exhibit **different characteristics** than normal data $\rightarrow$ can be isolated in **fewer** partition steps.

Figure 1: Schematic presentation



Isolation tree mechanism:

1) draw subsample from overall data

2) Recursively partition subsample:

- randomly select a split dimension $q \in \{1, 2\}$
- select a random split value $\min(x^q) < x^q_{\text{split}} < \max(x^q)$
- use x^q_{split} to split the data into two subsets

until all data instances are isolated

Isolation forest model

Number of splits as anomaly measure:

- once built, an isolation tree can return the number of splits necessary for the isolation of a data point
- anomalies likely need fewer splits to be isolated
- data points can be ranked by their respective number of splits

Isolation forest:

- construct many isolation trees to form a forest
- for each data point, an anomaly score between 0 and 1 can be computed based on the number of splits retrieved from each tree

Key parameters:

- t: number of trees in the forest
- ϕ: subsample size for each tree

Isolation forest model

Model advantages:

- only few parameters to optimize
- makes no distributional assumption about the input data set
- no distance/density calculations needed to obtain anomaly score
- provides an interpretable anomaly score for each transaction
- subsampling counters effects of swamping & masking

TARGET2 data

Customer payment transactions sent by **1 participant** over the period 01/06/2017 to 28/06/2018 $\rightarrow$ **roughly 700,000 transactions**.

Table 1: Transaction level input data

Variable	**Data type**	**Further details**
Submission day	Integer	Takes values in $\{1, 2, ..., 31\}$
Submission month	Integer	Takes values in $\{1, 2, ..., 12\}$
Settlement time	Float	Indicates hour, minute, second, and millisecond as numeric value
Settlement delay	Float	Time difference between submission and settlement time (seconds)
Transaction value	Float	Amount of transaction in euro
Originator BIC information field	String	BIC accounts
Receiver BIC information field	String	BIC accounts
Beneficiary BIC information field	String	BIC accounts

Data source: TARGET2.

TARGET2 data — Categorical variables

Low-cardinal categorical variables:

- majority of machine learning models need numeric data
- typically to allow for computation of distance & density metrics
- low-cardinal categorical data often transformed to binary variables

High-cardinal (our case) categorical variables:

- among the most challenging data types (Micci-Barreca, 2001)
- often disregarded, as transforming to binary variables is infeasible

Proposal: associate unique BICs to integers:

- joint association of BICs in originator, receiver, beneficiary attributes
- transformed attributes take values in $[0, \text{number unique BICs} - 1]$
- suitable since no distance/density computations needed

Methodology for training/evaluation

- training on 1 year of transactions, evaluating 1 week of transactions
- then shift both training/evaluation windows by 1 week and repeat

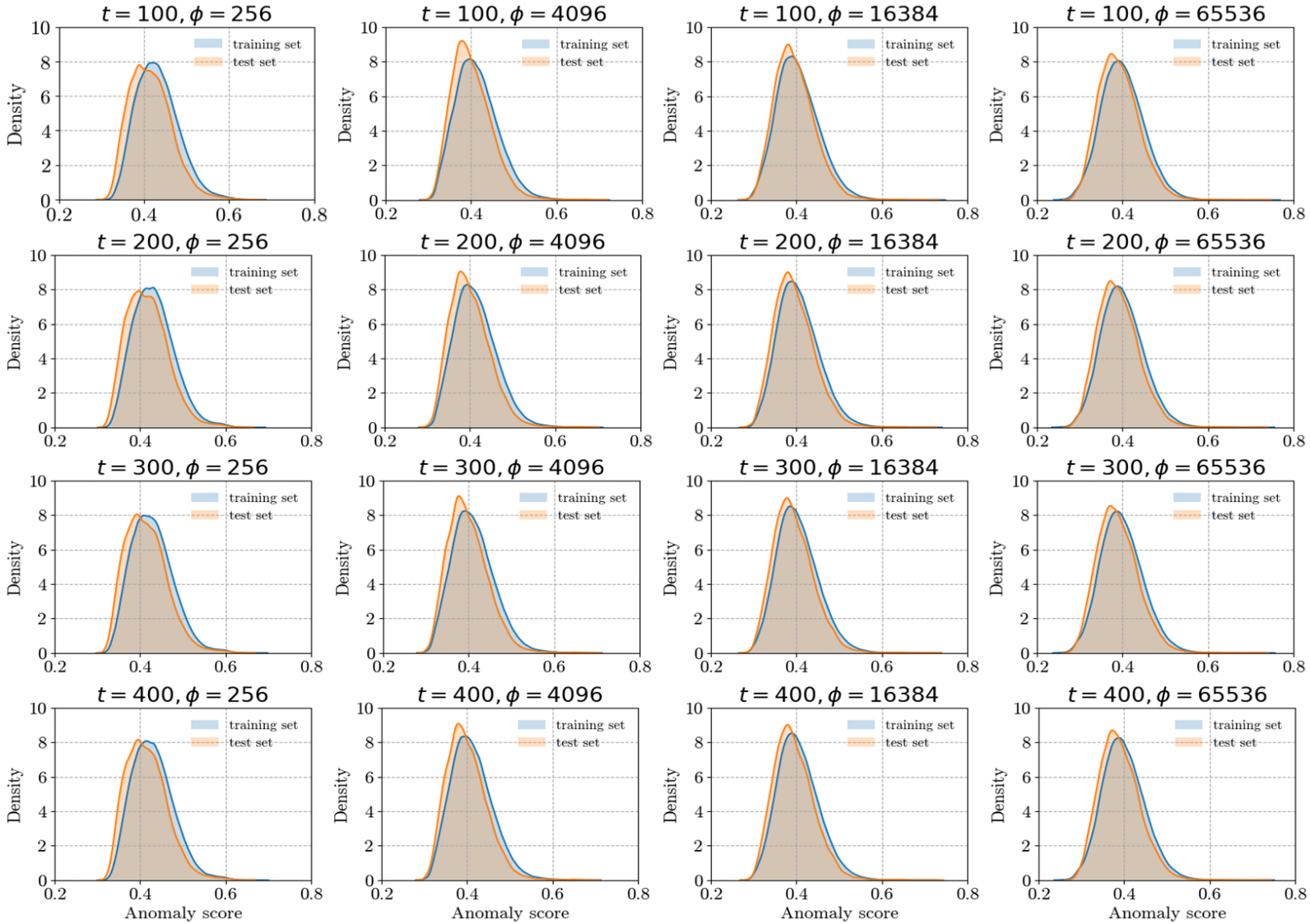
Table 2: Sliding window framework

	Window 1	**Window 2**	**Window 3**	**Window 4**
Training set start date	01 June '17	08 June '17	15 June '17	22 June '17
Training set end date	31 May '18	07 June '18	14 June '18	21 June '18
Number of transactions†	650,000	650,000	650,000	650,000
Test set start date	01 June '18	08 June '18	15 June '18	22 June '18
Test set end date	07 June '18	14 June '18	21 June '18	28 June '18
Number of transactions†	15,000	15,000	15,000	15,000

† For reasons of confidentiality, the exact number of transactions has been approximated to a range of 650,000 and 15,000 for training and test sets. Data source: TARGET2.

Model calibration — Number of isolation trees

Figure 2: Impact of number of trees t and subsample size ϕ on anomaly score



Note: Probability density functions of anomaly scores are estimated using Gaussian kernel density estimation.
Data source: TARGET2.

Model calibration — Subsample size

- set $t = 100$, investigate further the impact of the sub-sample size:

Table 3: Model specifications

Model	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**	**M10**
Sample size (ϕ)	128	256	512	1,024	2,048	4,096	8,192	16,384	32,768	65,536
Sample size (%)†	0.02%	0.04%	0.08%	0.16%	0.32%	0.63%	1.26%	2.52%	5.04%	10.08%

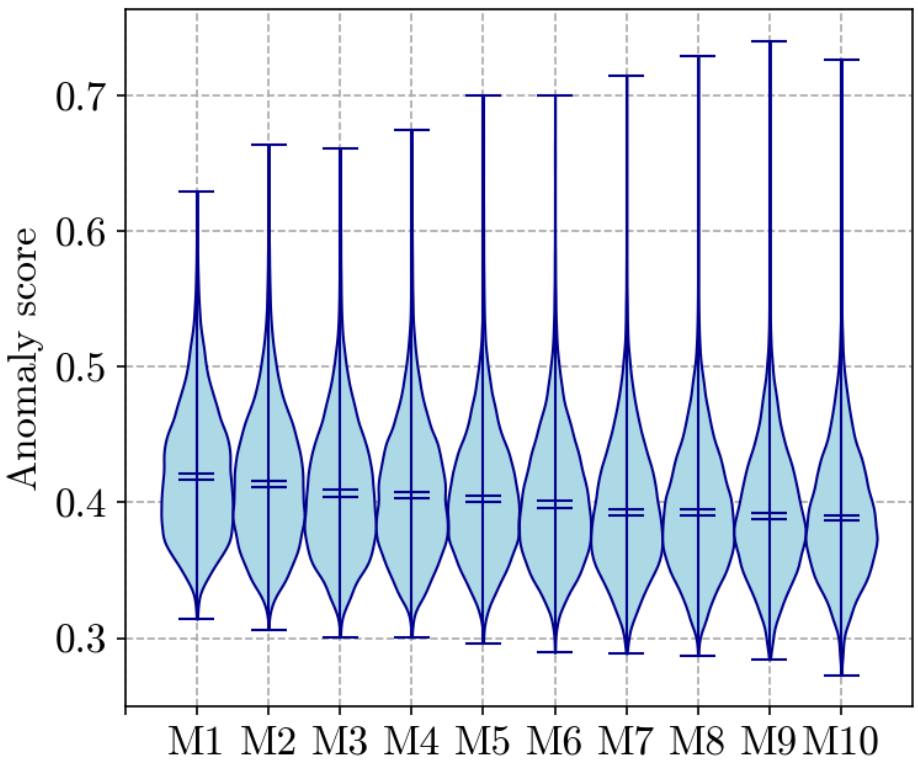
† Note: Sample size computed as share relative to the (approximate) number of transactions in training set (see Table 2). Data source: TARGET2.

Sample size matters for **detection ability** and **granularity**:

- based on $t = 100$ constructed trees, **M1** can cover at most 2% of transactions, while **M10** can cover all transactions many times
- **trade-off**: higher ϕ may improve detection ability and granularity, but increases possible effects of swamping & masking

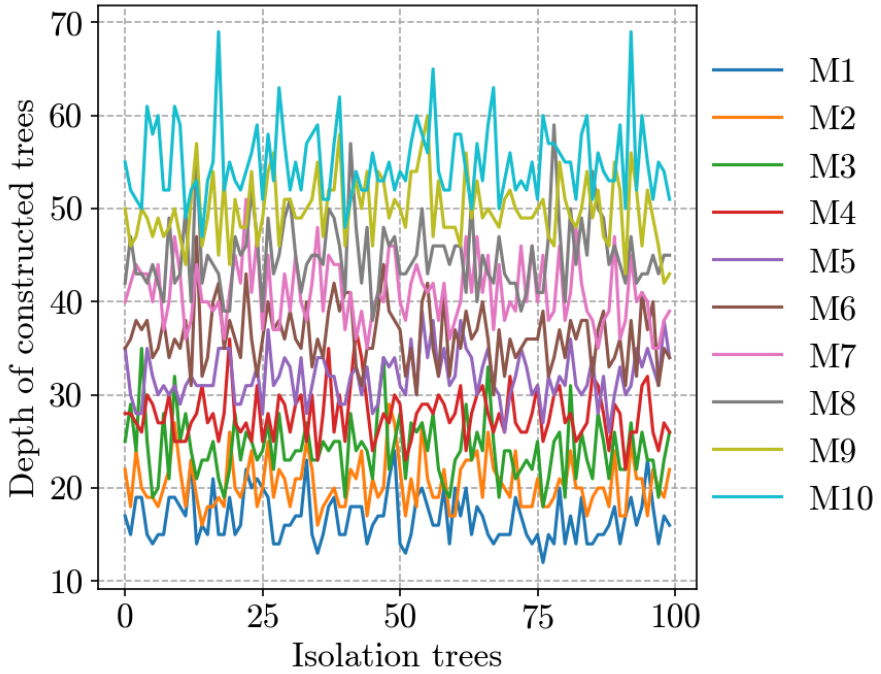
Model calibration — Subsample size

Figure 3: Distribution of anomaly scores across models



Note: Mirrored probability density functions of anomaly scores from models M1 to M10 for the test set. Approximated using Gaussian kernel density estimation. Data source: TARGET2.

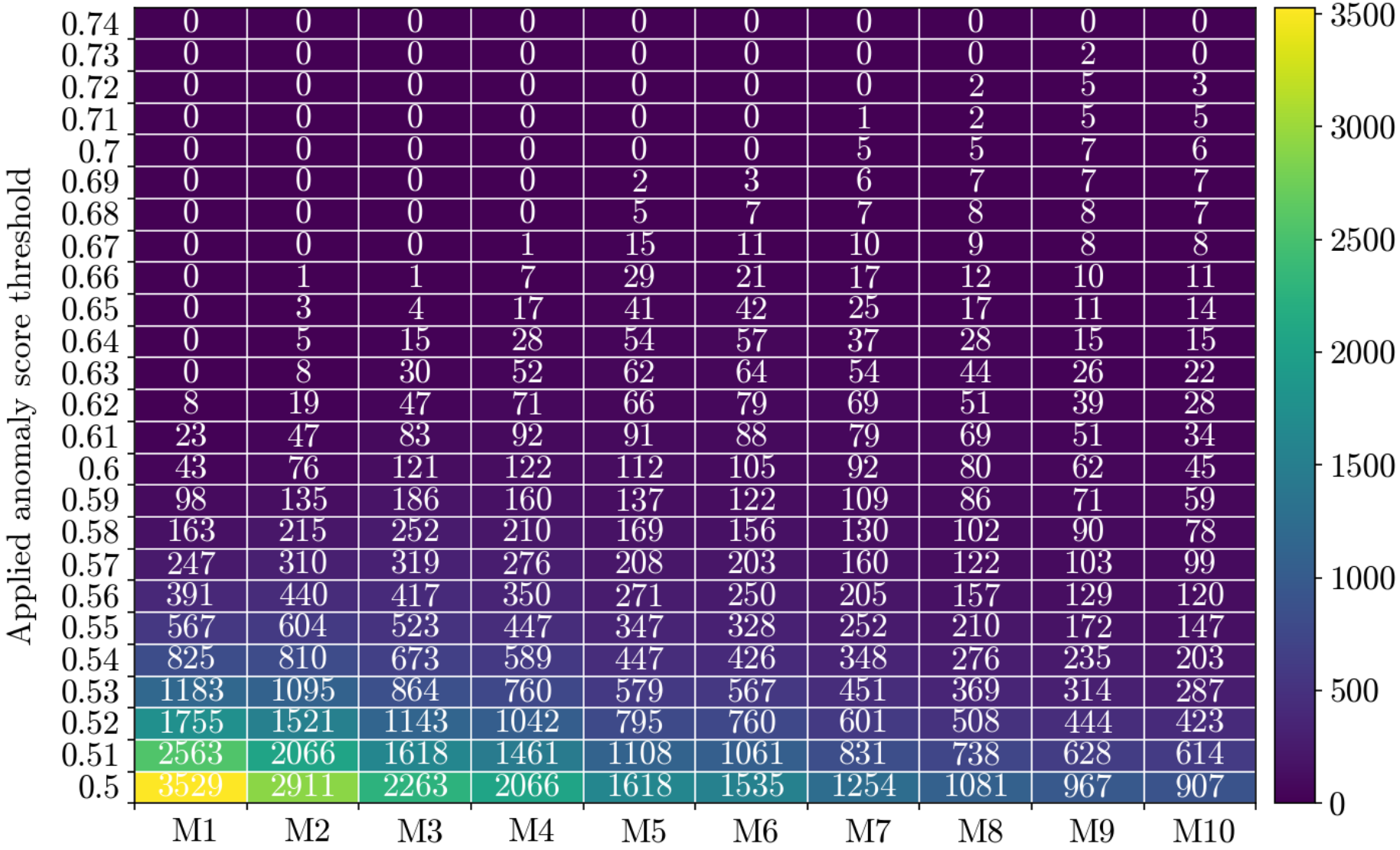
Figure 4: Depth of constructed isolation trees across models



Note: Trees are allowed to fully grow with the theoretical maximum tree depth set to $d_{\max} = \phi - 1$ (see Liu et al. (2012)). Data source: TARGET2.

Model calibration — Number of anomalies

Figure 5: Anomaly score threshold and detected anomalies



Note: Map indicating the number of identified anomalies (transactions having anomaly score $s \geq s^*$) across model specifications M1 to M10 and various threshold values $s^* \in (0, 1)$. Data source: TARGET2.

Model calibration — Overlap of anomalies

Figure 6: Overlap map

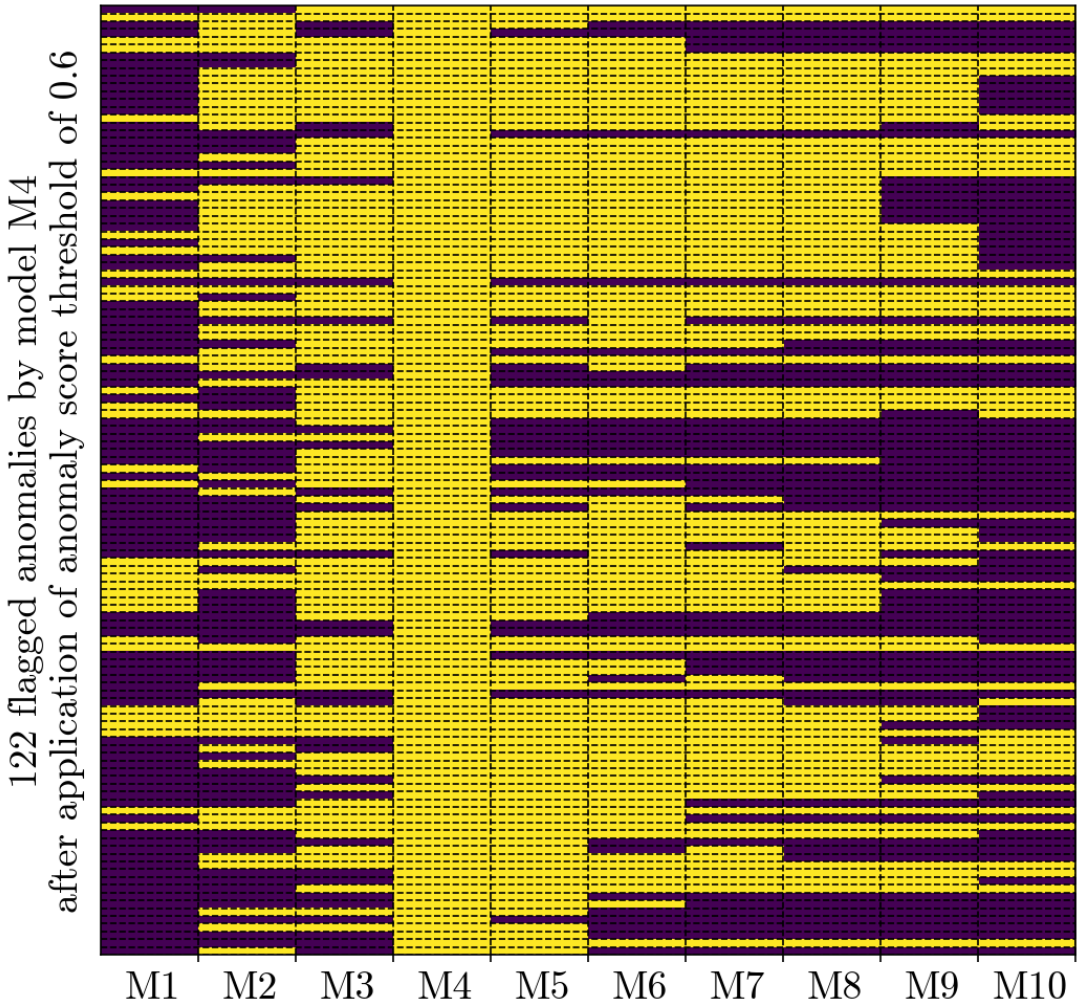


Table 4: Overlap share

Model	Overlap of detected anomalies relative to model M4
M1	26.23%
M2	49.18%
M3	**74.59%**
M4	**100.00%**
M5	**81.15%**
M6	**76.23%**
M7	**68.03%**
M8	**63.11%**
M9	48.36%
M10	35.25%

Note: Comparison of the overlap of individual anomalous transactions returned by models M1 to M10 under the application of an anomaly score threshold of $s^* = 0.60$. Under s^*, model M4 returns the largest number of anomalies (122 transactions) and is used as baseline. The figure on the left is a binary colour map with each rectangle representing an anomalous transaction returned by model M4. Yellow areas indicate an overlap, while purple areas indicate no overlap. Data source: TARGET2.

Model selection framework

Proposal: output stability as the main criterion for model selection.

Recall: Isolation forest has fundamentally **random elements**:

- random draws of ϕ transactions to form subset for tree construction
- random selection of split attribute
- random selection of split value for selected attribute

Idea: Repeated application of a model with the same parameters, same anomaly score threshold and on the same transaction set, should ideally detect the **same transactions**.

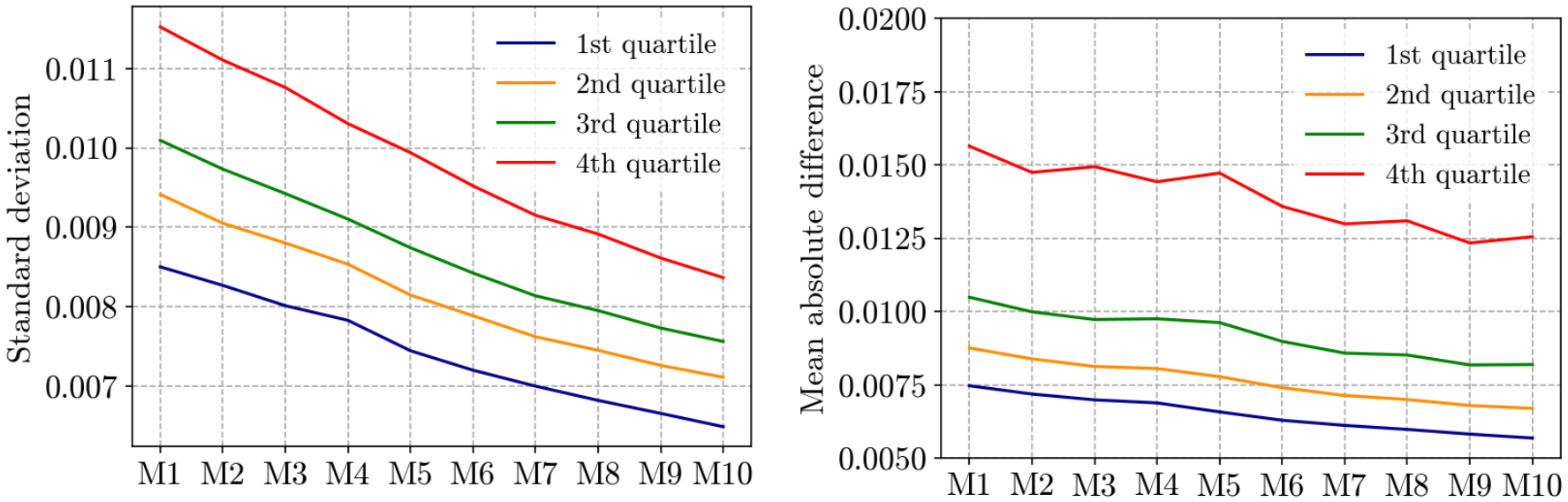
Model selection framework

Investigate **stability of model outputs** as follows:

1. select an anomaly score threshold (here: $s^* = 0.60$)
2. run all models 100 times with randomly chosen RNG seeds
3. investigate **2 output stability indicators**:
   - 3.1 distributional properties of anomaly scores
   - 3.2 overlap of detected anomalies per model

Model selection framework — (1) Distributional properties

Figure 7: Standard deviation and mean absolute difference of anomaly scores



Note: Standard deviation and mean absolute difference of anomaly scores produced for each transaction by models M1 to M10 with 100 different RNG seeds. Transactions have been ordered and grouped into quartiles. Data source: TARGET2.

Model selection framework — (2) Overlap of anomalies

Table 5: Overlap of detected anomalies after 100 runs

Model	Number of anomalies detected in baseline result	Average number of overlapping anomalies in alternative results	Standard deviation of number of overlapping anomalies in alternative results	Share of average number of overlapping anomalies
M1	114	12.88	12.47	11.30%
M2	201	74.45	24.34	37.04%
M3	224	120.63	23.90	53.85%
M4	182	126.94	12.32	69.75%
M5	152	117.32	8.14	77.18%
M6	128	101.91	5.97	79.62%
M7	**104**	**88.79**	**4.35**	**85.38%**
M8	94	76.71	4.97	81.61%
M9	72	57.01	3.20	79.18%
M10	60	45.90	3.83	76.50%

Notes: Comparison of the overlap of individual anomalous transactions returned by models M1 to M10 under anomaly score threshold $s^* = 0.60$. Anomalous transactions resulting from the first seed have been chosen as the baseline anomaly set. Anomalies obtained from other seeds are then compared relative to this baseline anomaly set to assess repeated detection ability, i.e. overlapping anomalies. Data source: TARGET2

Conclusion

What we do:

- apply the **isolation forest** to TARGET2 transaction data
- incorporate **categorical** BIC information
- introduce a model selection framework based on **output stability**

Next steps:

- further investigate variations in the number of trees t and the applied anomaly score threshold s^*
- expand input data set with additional attributes that might benefit anomaly detection
- investigate extended forms of the model (e.g. Hariri et al. (2018))
- benchmark the detection performance against alternative models commonly applied in anomaly detection

References I

Calheiros, R. N., Ramamohanarao, K., Buyya, R., Leckie, C., and Versteeg, S. (2017). On the effectiveness of isolation-based anomaly detection in cloud data centers. *Concurrency and Computation: Practice and Experience*, 29(18):e4169.

Hariri, S., Kind, M. C., and Brunner, R. J. (2018). Extended isolation forest. *arXiv preprint arXiv:1811.02141*.

Hawkins, S., He, H., Williams, G., and Baxter, R. (2002). Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 170–180. Springer.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):3.

References II

Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1):27–32.

Tax, D. M. and Duin, R. P. (2004). Support vector data description. *Machine learning*, 54(1):45–66.

Triepels, R., Daniels, H., and Heijmans, R. (2017). Detection and explanation of anomalous payment behavior in real-time gross settlement systems. In *International Conference on Enterprise Information Systems*, pages 145–161. Springer.