Faculty of Computer Science

Programme “Master of Data Science”

Moscow
2022

SpiderNet: Fully Connected Residual Network for Fraud Detection

Student: Sergey Afanasiev

Supervisor: PhD, Alexey Masyutin

Agenda

1. Introduction
2. About fraud detection
3. SpiderNet: problem formulation
4. B-tests and W-tests
5. Experiment Setup
6. Results
7. Conclusions

With the development of high technologies, the volume of fraud is increasing

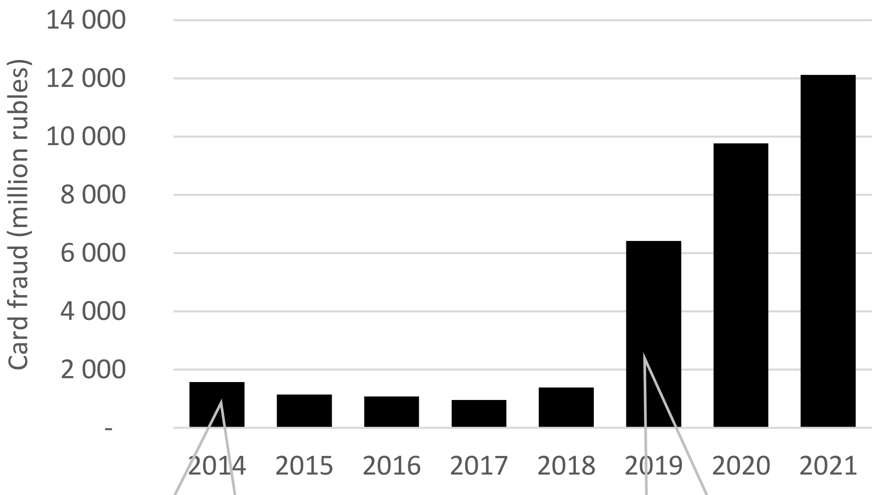
Partner fraud. In 2018, eight Indian banks incurred $1.3 billion in losses in a fraud case involving Kingfisher Airlines founder Vijay Mallya [1].

Internal fraud. The Agricultural Bank of China faced losses of $497 million after being defrauded by employees of billionaire Guo Wengui [2].

Hacker fraud. In 2019, the FBI issued an official announcement that global losses from fraudulent Business Email Compromise (BEC) reached $26 billion during the period from June 2016 to July 2019 [3].

Social engineering. Losses of Russian banks' clients from card fraud reached 12,13 billion rubles in 2021 [4].

Volume of card fraud in Russia*



*Bank of Russia: http://www.cbr.ru/analytics/ib/

Previously, skimming was the main fraud problem. Skimming was defeated with EMV technology (cards with a chip). Since 2013, Russian banks have been issuing cards with chips.

Social engineering has replaced skimming and has become an even greater threat to banks and their clients.

[1] https://www.theguardian.com/world/2020/apr/20/kingfisher-airlines-tycoon-vijay-mallya-loses-appeal-extradition-india
[2] https://www.reuters.com/article/us-china-corruption-tycoon-idUSKBN1900DL
[3] https://www.ic3.gov/Media/Y2019/PSA190910
[4] http://www.cbr.ru/analytics/ib/operations_survey_2021/

Social engineering is a worldwide problem

“If you close one phone fraud company, there will be five more”

“Phone scams and pet Halloween costumes are the only growing industries in America”

© The Simpsons (Episode 2, Season 33)

Theory of differential association

If the environment of the individuals is dominated by criminals, then they learns their values and behaviors and becomes a criminals themself. The strength and influence of connections depend on the characteristics of the individual's communication with criminals:

- **Frequency** - how often and regularly
- **Duration** - how long
- **Priority** - from what age

Accordingly, the strongest influence on individuals is usually exerted by their relatives.

Edwin Hardin Sutherland
(1883 – 1950)

Fraud Triangle

The reasons necessary for a person to commit fraudulent actions:

- **Pressure** - a person must experience pressure (financial or otherwise)
- **Opportunity** - a person should have the opportunity to commit and hide an act of fraud for some time
- **Rationalization** - a person must justify his action to himself.

Donald Cressey
(1919 – 1987)

Crime and Punishment: An Economic Approach

Crime can be viewed as a specific market in which there is supply and demand:

Probability of not being caught

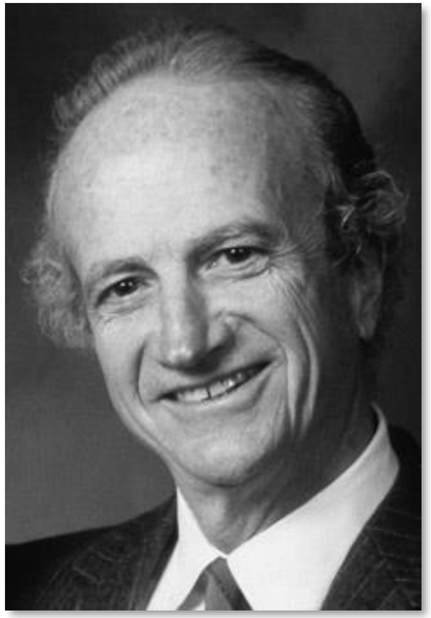
Proceeds of crime

Income from fair business

$$(1-\pi)\times U(W_C) - \pi \times S \;>\; U(W_L)$$

Probability of being caught

Sanctions



Gary Becker
(1930 – 2014)

Anti-fraud tools

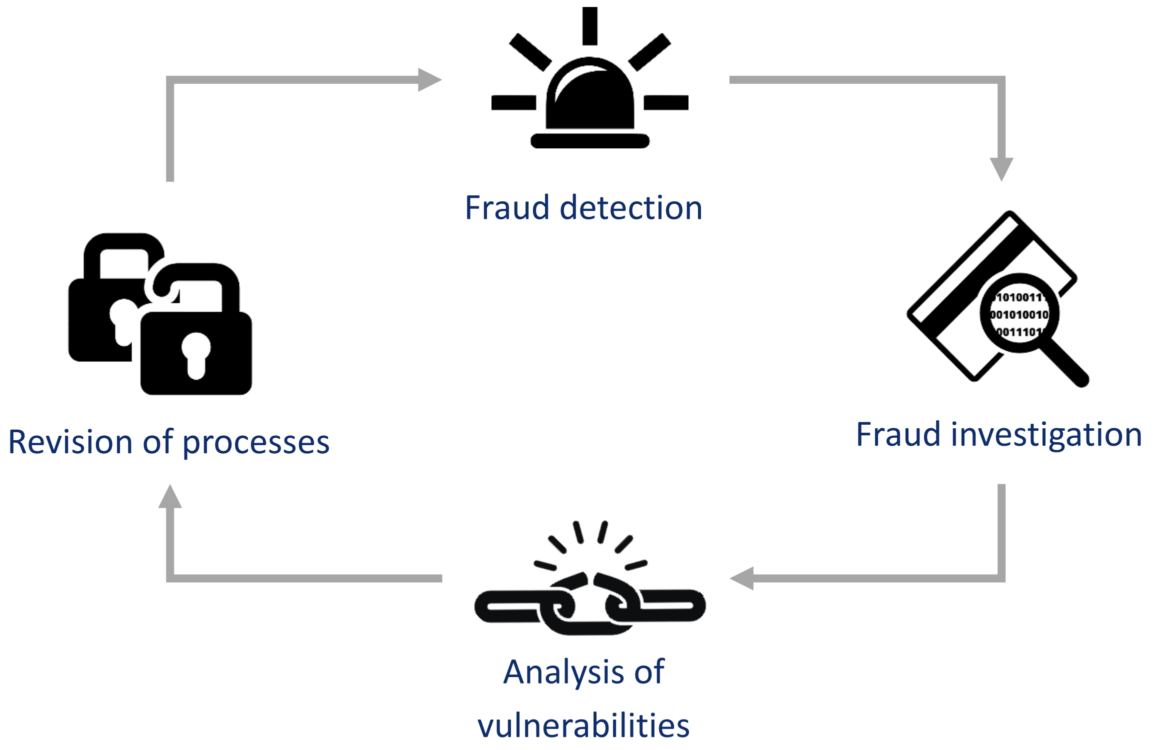
Directive	Detective	Preventive

- Instructions for employees
- Clients/Partners contract
- Antifraud newsletters

- Fraud detection systems
- Video surveillance systems
- Underwriting and auditing

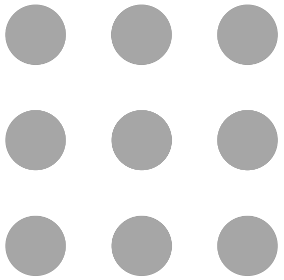
- Safes, logins/passwords
- Biometric systems
- Access restrictions

Principle of the cyclic scientific method

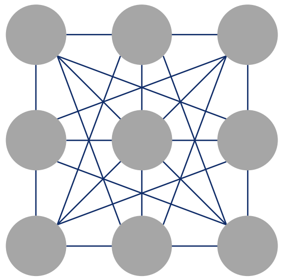


Scheme of developing strong rules for fraud detection

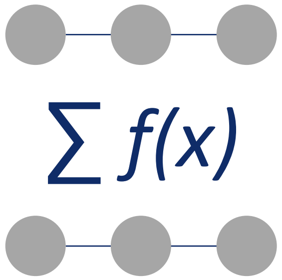
Features → **Rules** → **Strong Rules**



Analysts develop features for fraud detection based on the results of investigations (applications, behavior, anomalies, velocity, etc.)



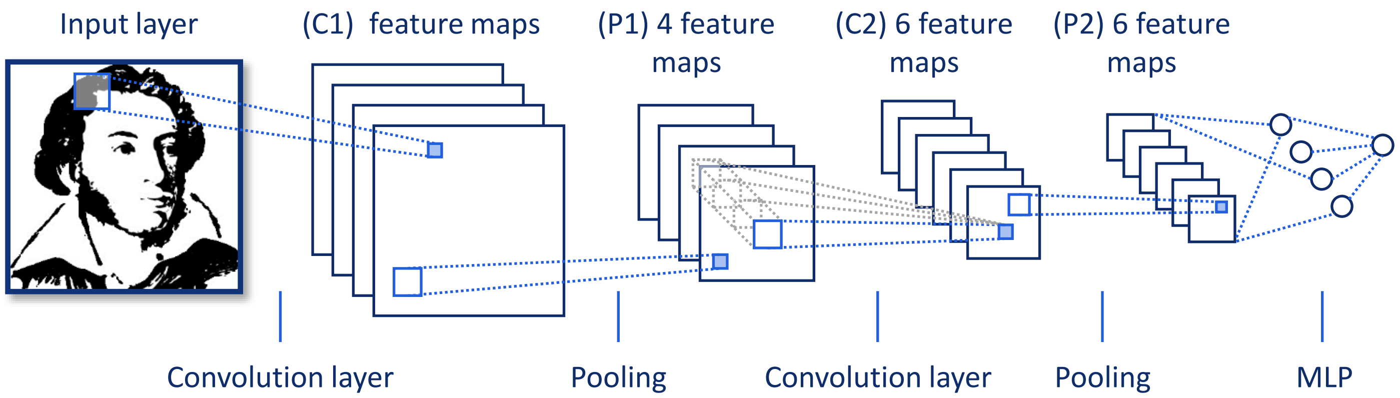
Analysts build rules from features using machine learning methods: Decision Trees, Branch & Bound, Association Analysis etc.

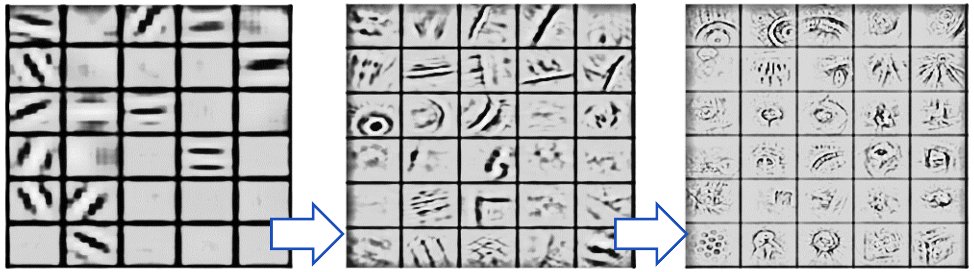


Strong rules are selected from the rules using the iterative method to be included in fraud detection system

Convolutional Network makes hierarchical features

LeNet-5





Feature Map1 Feature Map2 Feature Map3

The main ideas for developing an Antifraud Neural Network

1. The neural network can create strong features from weak ones — **Convolutional**
2. The neural network can choose top-strong features from different ones — **Pooling**
3. Strong features should be immediately forwarded to the output layers — **Skip-connection**

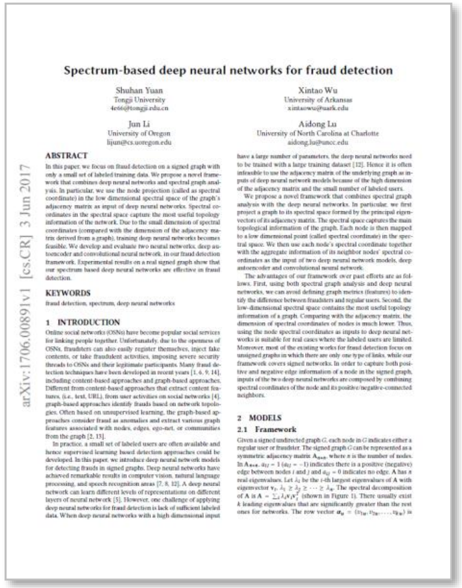
In previous anti-fraud works, NN-architectures were transferred from the CV and NLP domains

2016 CNN


Credit Card Fraud Detection Using Convolutional Neural Networks

Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang

2017 CNN-LSTM

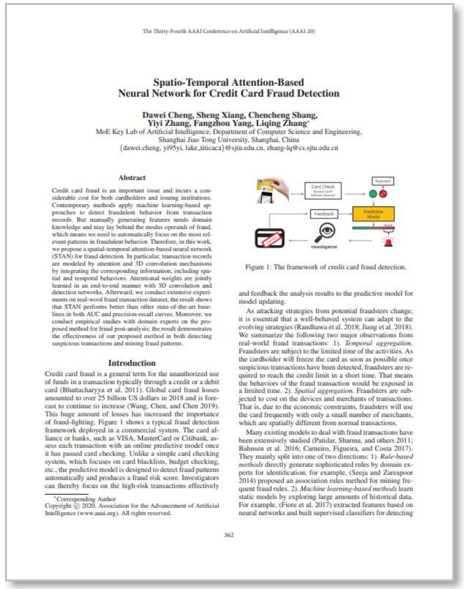

Spectrum-based deep neural networks for fraud detection

Shuhan Yuan, Xintao Wu, Jun Li, Aidong Lu

2018 DenseNet


Multi-Scale DenseNet-Based Electricity Theft Detection

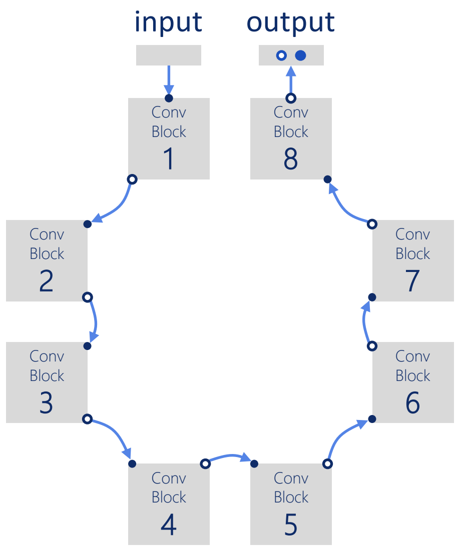
2020 Attention-CNN


Spatio-Temporal Attention-Based Neural Network for Credit Card Fraud Detection

Dawei Cheng, Sheng Xiang, Chencheng Shang, Yiyi Zhang, Fangzhou Yang, Liqing Zhang

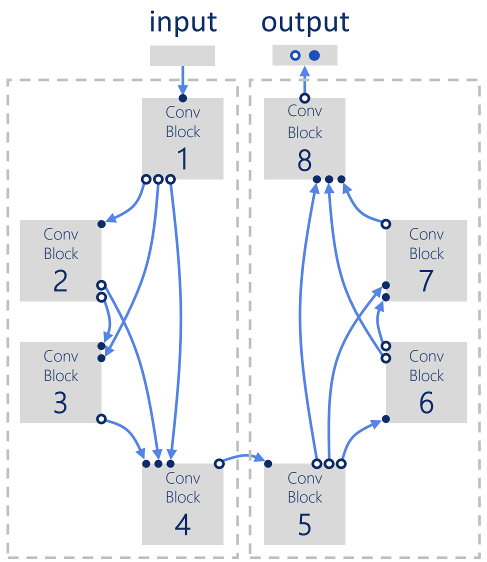
CNN architectures designed for fraud detection

1D-CNN



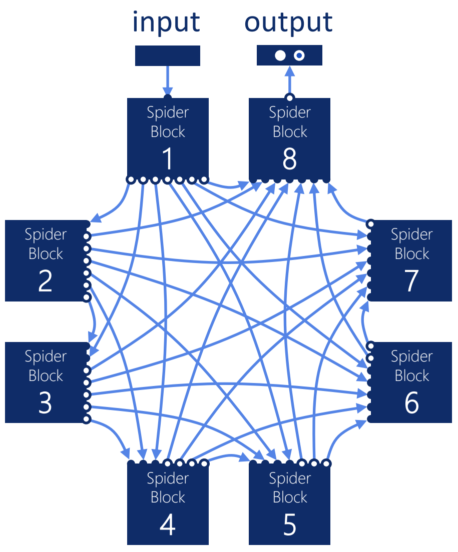
Classical CNN hasn’t skip-connections, therefore strong features aren’t forwarded to the network output directly

F-DenseNet



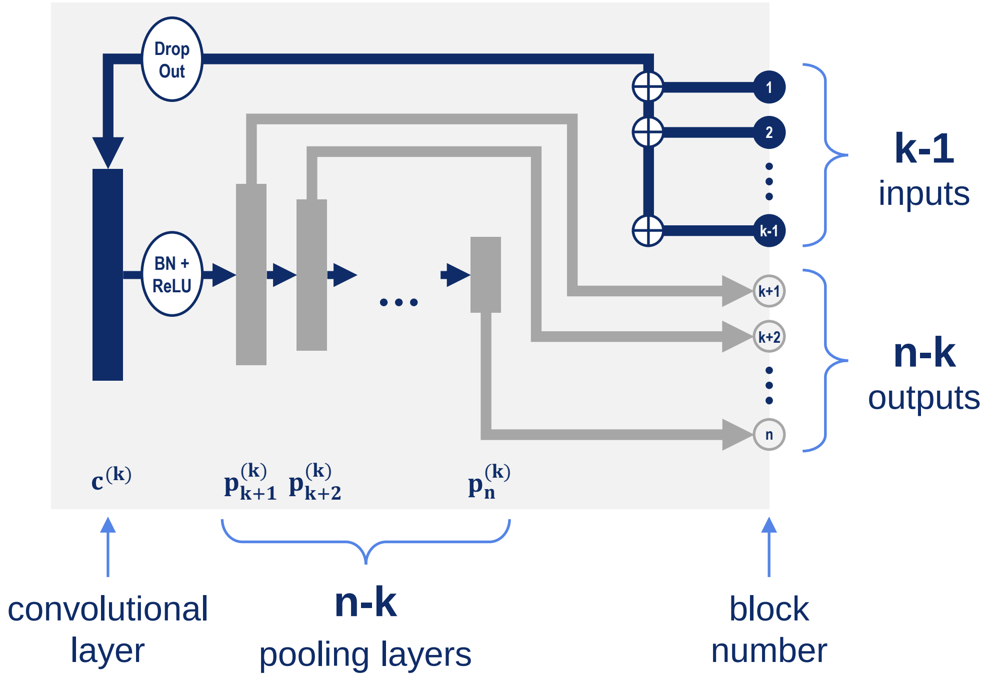
DenseNet has a bottleneck, which will prevent strong features from being forwarded to the network output

SpiderNet

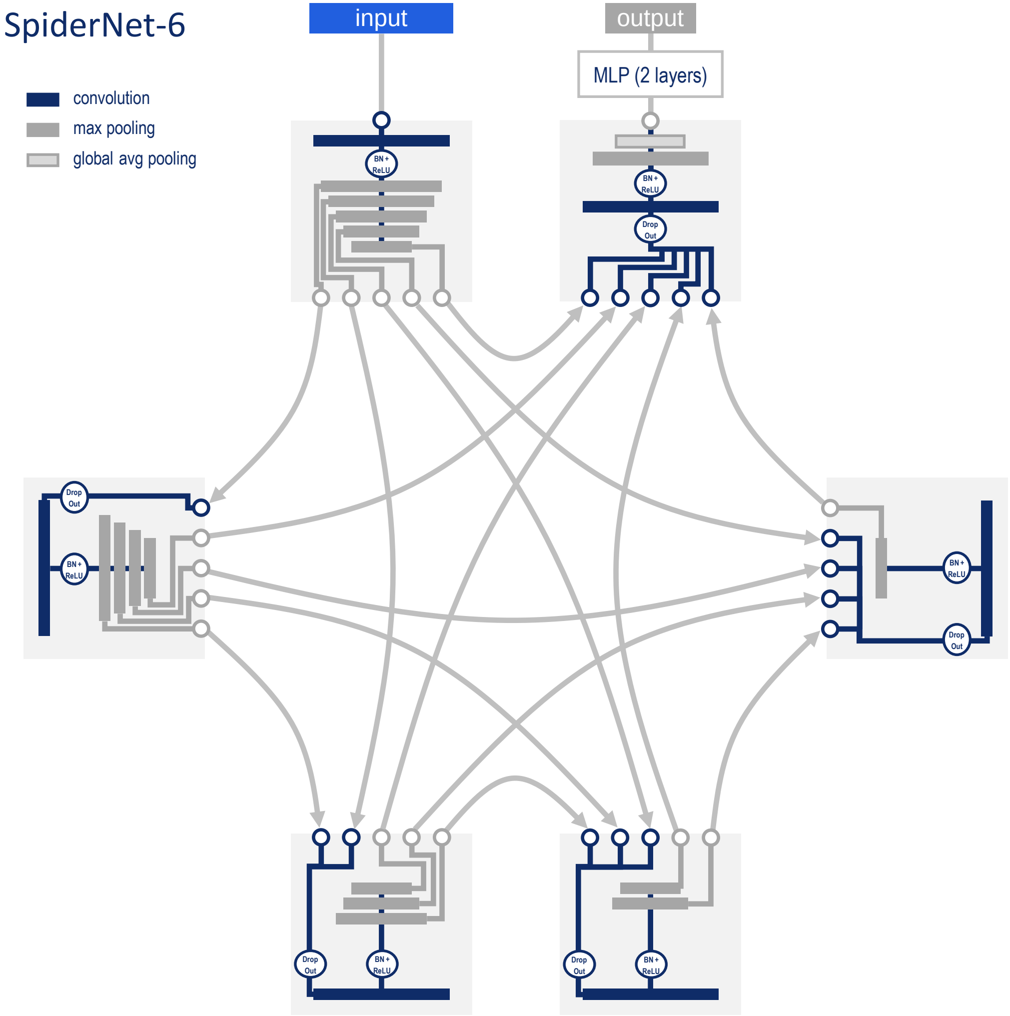


SpiderNet has skip-connections between all layers of the network and doesn't contain bottlenecks

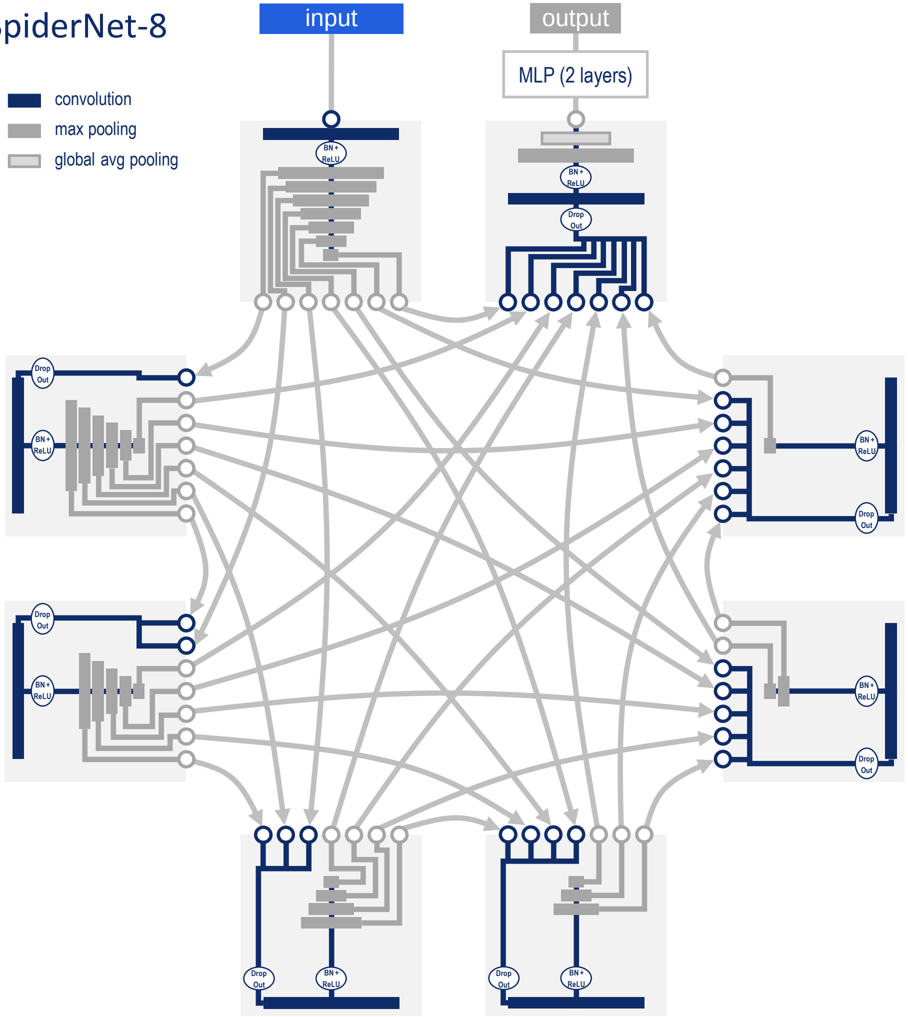
Scheme of the kth Spider-block with convolutional layer and n-k pooling layers (n is the total number of Spider-blocks)



SpiderNet-6



SpiderNet-8



Benford’s Law

1881
Simon Newcomb
Astronomer

Newcomb found that logarithmic reference books contain the digits "1" more than digits "2", the digits "2" more than digits "3," etc.

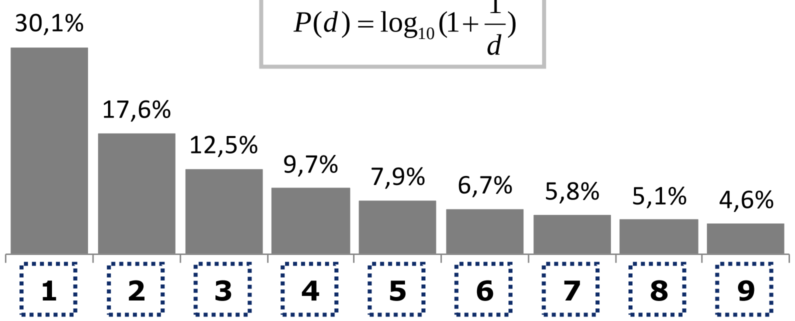
1938
Frank Benford
Physicist

Benford analyzed different reference book with data and he calculated empirical law of distribution of first digits

1993
Mark J. Nigrini
Accounting

Nigrini developed tests for financial audit and revealed the embezzlement of $2 million from Treasury of the Arizona state

$$P(d) = \log_{10}(1 + \frac{1}{d})$$



B-tests and W-test for internal fraud detection

Statistic for B-test:

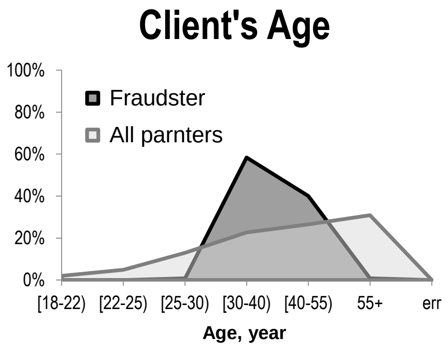
$$S = \frac{1}{2}\sum_{i=1}^{n}|a_i - b_i|$$

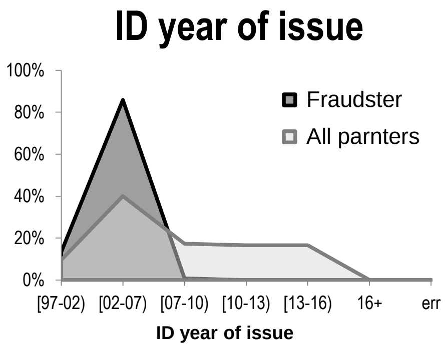
where a_i, и b_i are the compared distributions,
n is the number of quantiles in the distribution

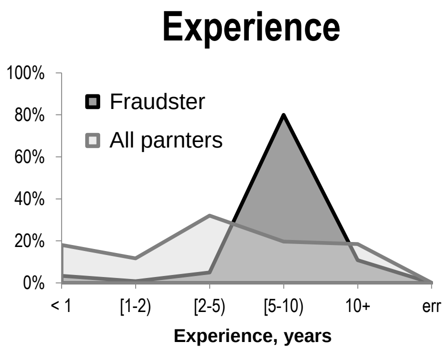
Statistic for W-test:

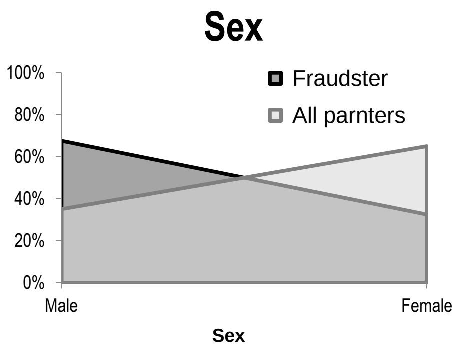
$$W_p(\mu, \nu) = inf_{\gamma\in\Gamma(\mu,\nu)} E_{(x,y)\sim\gamma}[||x - y||]$$

where $E[Z]$ denotes the expected value of a random variable Z and the infimum is taken over all joint distributions of the random variables X and Y with marginals μ and ν respectively









Characteristics of private and public datasets

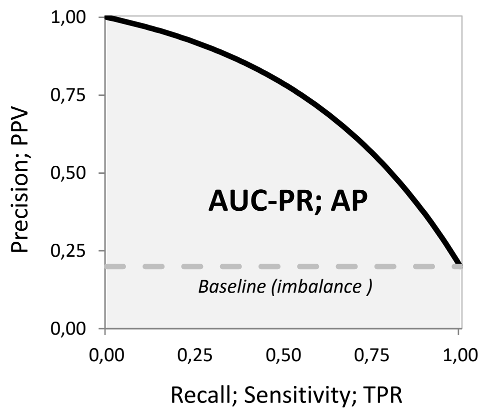
	Private Data	Public Data
Source	Russian Bank	Ant Financial Services Group
Type of data	POS credits	Payments
Type of fraud	Internal	Transactional
Period time	03.2014-10.2019	09.2017-11.2017
All observations, #	1 880 499	990 006
Fraud observations, #	5 327	12 122
Fraud ratio, %	0.283	1.224
All features, #	509	297
Selected features, #	163	128

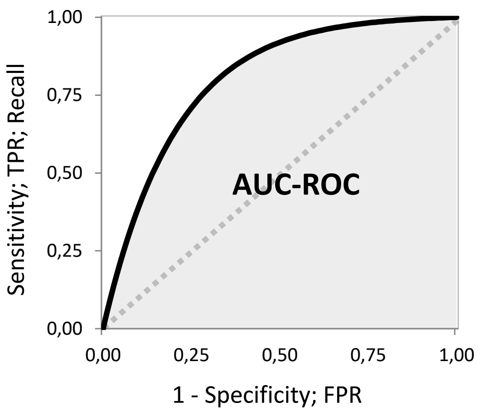
Evaluation metrics

Public Dataset

$AUC\ PR\ (Avg\ Precision)$ – recommended for imbalanced data

$AUC\ ROC$ – banking standard (Gini = 2 * AUC_ROC – 1)





Private Dataset

$AUC\ PR\ (Avg\ Precision)$ – recommended for imbalanced data

$AUC\ ROC$ – banking standard (Gini = 2 * AUC_ROC – 1)

$$PL = \sum_{i=1}^{k} PL^{(i)} \cdot b_i$$

where $PL^{(i)}$ is prevented loss for the ith fraud partner;
k is the number of the first k partners with the highest model probability of fraud;
b_i is binary variable showing the event for the ith partner: 1 (fraud), 0 (no fraud).

$$PL^{(i)} = P_{(T_l - T_a)} \cdot \frac{DR - DR_0}{1 - DR_0}$$

where T_l is the whole considered period of loss;
T_a is the period on which the model works;
$(T_l - T_a)$ is the period after the model is triggered (for our sample, it is 90 days – the empirical period for which the bank's Security detects fraud without using the model);
DR is the Default-Rate for loans issued by the partner for the period $(T_l - T_a)$;
DR_0 is a "zero target" for Default-Rate in which the loan portfolio has zero profit;
$P_{(T_l - T_a)}$ is the partner's loan portfolio for the period $(T_l - T_a)$.

Hyperparameters and tricks

Random Forest

- Train/Val/Test (80%:10%:10%)
- Feature selection: low fill rate, cross-correlation matrix
- Tuning hyperparameters by Optuna library on 5-fold cross-validation: *max_depth, N_estimators, class_weight*

1D-CNN

CNN-3, CNN-6, CNN-8

- Train/Val/Test (80%:10%:10%)
- Feature selection: low fill rate, cross-correlation matrix
- 3, 6 and 8 convolutional layers
- Tuning hyperparameters by Optuna library and manual GridSearch on 5-fold cross-validation: *l2_batch, n_filters, kernel_size, weight_decay, learning_rate, hidden, dropout*
- Decay learning rate scheduler
- BatchNorm + ReLU
- Fraud-rate leveling in batches
- Early stopping

1D-DenseNet

DenseNet-6, DenseNet-8

- Train/Val/Test (80%:10%:10%)
- Feature selection: low fill rate, cross-correlation matrix
- Two DenseNet-blocks with 3 and 4 conv layers in each block
- Tuning hyperparameters by Optuna library and manual GridSearch on 5-fold cross-validation: *initial_filters, initial_stride, k, conv_kernel_width, bottleneck_size, theta, transition_pool_stride, initial_conv_width, initial_pool_width, initial_pool_stride*
- Decay learning rate scheduler
- Fraud-rate leveling in batches
- Early stopping

F-DenseNet

F-DenseNet-6, F-DenseNet-8

- Train/Val/Test (80%:10%:10%)
- Feature selection: low fill rate, cross-correlation matrix
- Two blocks with 3 and 4 conv layers in each block
- Tuning hyperparameters by Optuna library and manual GridSearch on 5-fold cross-validation: *l2_batch, dropout, kernel_size, n_filters, hidden, weight_decay, learning_rate*
- Decay learning rate scheduler
- BatchNorm + ReLU
- Fraud-rate leveling in batches
- Early stopping

SpiderNet

SpiderNet-6, SpiderNet-8

- Train/Val/Test (80%:10%:10%)
- Feature selection: low fill rate, cross-correlation matrix
- 6 and 8 Spider-blocks
- Tuning hyperparameters by Optuna library and manual GridSearch on 5-fold cross-validation: *l2_batch, n_filters, kernel_size, hidden, weight_decay, dropout, learn_rate, dropout_block_k (where k is a block number)*
- Decay learning rate scheduler
- BatchNorm + ReLU
- Fraud-rate leveling in batches
- Early stopping

Public dataset: quality of models for transactional fraud detection

The best results are highlighted in bold; 95% confidence intervals are shown in parentheses

#	Model	Public data (test sample)	
		AUC PR	AUC ROC
1	Random Forest	0.4881 (±0.003114)	0.9709 (±0.003572)
2	CNN-3	0.4462 (±0.003096)	0.9670 (±0.004674)
3	CNN-6	0.4908 (±0.003114)	0.9711 (±0.004780)
4	CNN-8	0.5099 (±0.003114)	0.9718 (±0.004511)
5	DenseNet-6 [3; 3]	0.4757 (±0.003111)	0.9669 (±0.004935)
6	DenseNet-8 [4; 4]	0.4854 (±0.003113)	0.9686 (±0.004661)
7	F-DenseNet-6 [3; 3]	0.5092 (±0.003114)	0.9708 (±0.005082)
8	F-DenseNet-8 [4; 4]	0.4968 (±0.003114)	0.9704 (±0.004780)
9	SpiderNet-6	**0.5375** (±0.003106)	**0.9721** (±0.004763)
10	SpiderNet-8	0.5160 (±0.003113)	0.9684 (±0.004744)

Private dataset: quality of models for internal fraud detection

The best results are highlighted in bold; 95% confidence intervals are shown in parentheses

#	Model	Private data (test sample)	
		AUC PR	AUC ROC
1	Random Forest	0.0650 (±0.001116)	0.9371 (±0.009253)
2	CNN-3	0.0527 (±0.001012)	0.9339 (±0.012978)
3	CNN-6	0.0644 (±0.001111)	0.9385 (±0.011432)
4	CNN-8	0.0708 (±0.001161)	0.9288 (±0.009605)
5	DenseNet-6 [3; 3]	0.0646 (±0.001113)	0.9315 (±0.009091)
6	DenseNet-8 [4; 4]	0.0691 (±0.001148)	0.9310 (±0.010545)
7	F-DenseNet-6 [3; 3]	0.0732 (±0.001179)	0.9263 (±0.014509)
8	F-DenseNet-8 [4; 4]	0.0575 (±0.001054)	0.9186 (±0.015820)
9	SpiderNet-6	**0.0948** (±0.001326)	**0.9484** (±0.008004)
10	SpiderNet-8	0.0680 (±0.001139)	0.9277 (±0.009588)

Private dataset: PL-quality of models for internal fraud detection

The best results are highlighted in bold

#	Model	Private data (test sample)	
		PL	Fraud, #
	Random classifier	$ 325 604	48
1	Random Forest	$ 2 079 527	208
2	CNN-3	$ 2 235 707	1 **312**
3	CNN-6	1 **$ 2 753 821**	280
4	CNN-8	$ 2 337 297	280
5	DenseNet-6 [3; 3]	$ 2 324 181	240
6	DenseNet-8 [4; 4]	$ 2 433 914	3 288
7	F-DenseNet-6 [3; 3]	$ 2 297 848	240
8	F-DenseNet-8 [4; 4]	3 $ 2 402 470	272
9	SpiderNet-6	2 $ 2 570 014	2 304
10	SpiderNet-8	$ 2 379 977	264
	Perfect classifier	$ 4 659 439	888

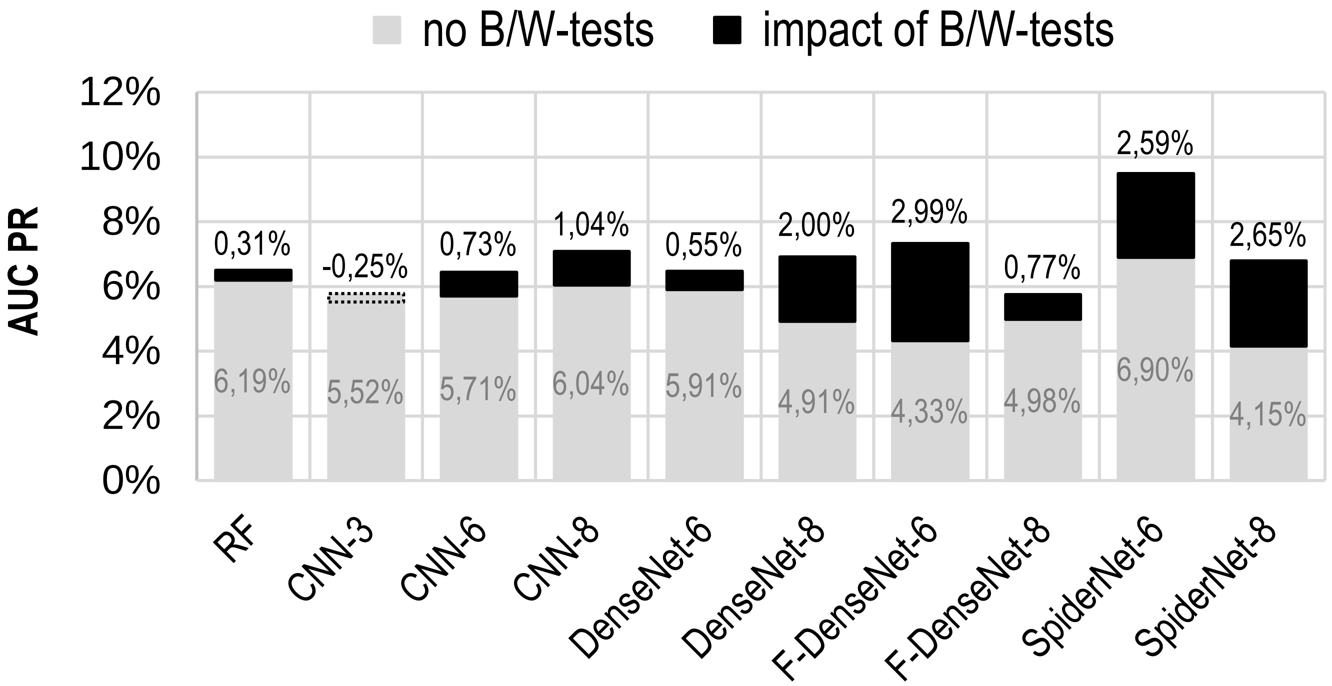
Sign tests for two pairs: 1) CNN-3 and SpiderNet-6; 2) CNN-6 and SpiderNet-6

The PL (recall) and Fraud (recall) metrics are normalized according to the perfect classifier

	CNN-3	SpiderNet-6	CNN-6	SpiderNet-6
Private data:				
AUC PR	0.0527	**0.0948**	0.0644	**0.0948**
AUC ROC	0.9339	**0.9484**	0.9385	**0.9484**
PL (recall)	0.4798	**0.5516**	**0.5910**	0.5516
Fraud (recall)	**0.3514**	0.3423	0.3153	**0.3423**
Public data:				
AUC PR	0.4462	**0.5375**	0.4908	**0.5375**
AUC ROC	0.9670	**0.9721**	0.9711	**0.9721**
p-value	0.015625		0.015625	

Influence of B-tests and W-tests on the AUC PR of the model

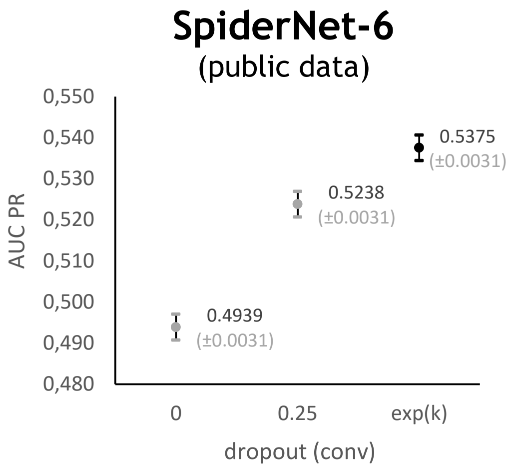
(Negative impact means a decrease in the model quality when adding B/W-tests)

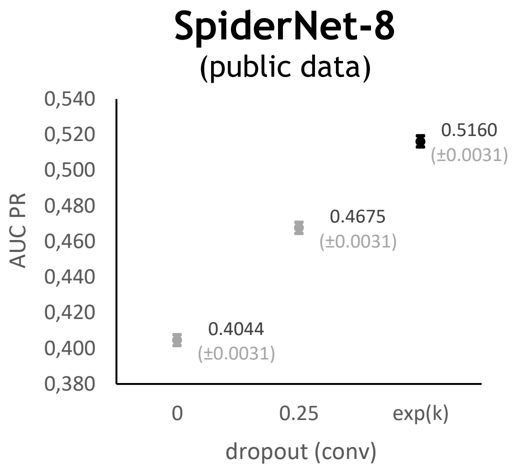


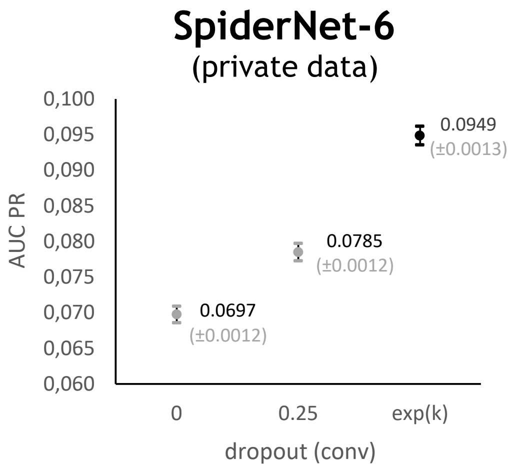
SpiderNet quality (AUC PR), depending on the dropout techniques

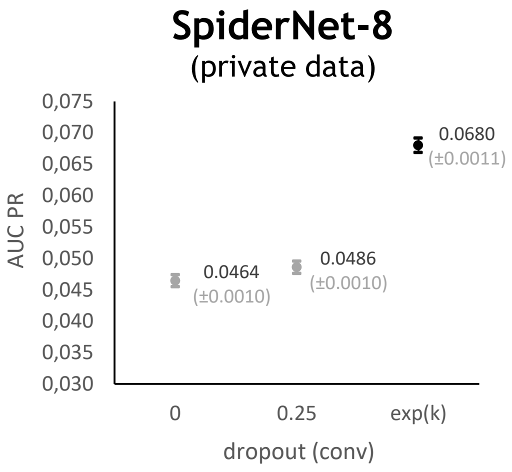
1) The value of zero corresponds to SpiderNet quality without a dropout;
2) The value of 0.25 corresponds to constant dropout=0.25 used in all Spider-blocks (vanilla technique);
3) The value of exp(k) corresponds to an exponential increase in the dropout value as the Spider-block number increases (our technique).

Public Dataset	**Private Dataset**









Conclusions

- ✔ In this work we have proposed a SpiderNet – a novel neural network architecture for fraud detection, which is an inductive bias network for tabular data. Using convolutional layers, our SpiderNet creates hierarchical anti-fraud rules, and skip-connections between blocks allows strong rules to be forwarded to the network output. Also, SpiderNet can select strong rules early on through the use of a multi-layered structure of pooling layers in Spider-blocks.
- ✔ SpiderNet should work well for heterogeneous input data, when there are clear leaders among the rules supplied to the network input and they must be forwarded to the output without additional transformation (scores of other models, strong rules, etc.).
- ✔ We proposed new methods for developing antifraud rules – B-tests and W-tests, which significantly affect the quality of the models.
- ✔ We also developed the Prevented Losses metric, which can be used to evaluate the cost-effectiveness of anti-fraud models.

- ☐ SpiderNet does not solve all the anti-fraud modeling problems identified by Bolton, Hand, Provost, and Breiman [see appendix]. In particular, we still use expert rules designed for specific fraud types to train models. Our B-tests and W-tests partially solve this problem, but there are other strong methods for fraud feature engineering, such as graph methods [47, 55, 12, 56], entropy changing methods [14], and variance anomaly detection methods (V-tests, similar to B-tests).
- ☐ An important component of SpiderNet is skip-connection, which helps to forward strong features directly to the output layers of the network, partially solving the problem of locality in convolutions, when the order of features in the input vector is important, and their rearrangement leads to a change in the quality of the model. However, the current implementation of SpiderNet does not completely solve the locality problem. Our future work will focus on this problem.

The SpiderNet code is available at: https://github.com/AfanasevSergey/HSE_Diploma_2022_SpiderNet

Thank you for your attention