Designing an attack-defense game: how to increase the robustness of financial transaction models via a competition

1st Alexey Zaytsev
Skoltech
Moscow, Russia
A.Zaytsev@skoltech.ru

2nd Maria Kovaleva
Skoltech
Moscow, Russia
Maria.Kovaleva@skoltech.ru

3rd Alex Natekin
Open Data Science
Moscow, Russia
natekin@ods.ai

4rd Evgeni Vorsin
Innotech
Moscow, Russia
vorsineo@gmail.com

5th Valerii Smirnov
Lomonosov Moscow State University
Moscow, Russia
smirnovvs@my.msu.ru

6th Georgii Smirnov
Lomonosov Moscow State University
Moscow, Russia
Georgii.S.Smirnov@gmail.com

7th Oleg Sidorshin
Kazan Federal University
Kazan, Russia
oasidorshin@gmail.com

8th Alexander Senin
Lomonosov Moscow State University
Moscow, Russia
aaasenin@gmail.com

9th Alexander Dudin
Kaspersky Lab
Moscow, Russia
alexander.dudin@outlook.com

10th Dmitry Berestnev
Zvuk
Moscow, Russia
Toberest@gmail.com

***Abstract*—**
Banks routinely use neural networks to make decisions. While these models offer higher accuracy, they are susceptible to adversarial attacks, a risk often overlooked in the context of event sequences, particularly sequences of financial transactions, as most works consider computer vision and NLP modalities.

We propose a thorough approach to studying these risks: a novel type of competition that allows a realistic and detailed investigation of problems in financial transaction data. The participants directly oppose each other, proposing attacks and defenses — so they are examined in close-to-real-life conditions.

The paper outlines our unique competition structure with direct opposition of participants, presents results for several different top submissions, and analyzes the competition results. We also introduce a new open dataset featuring financial transactions with credit default labels, enhancing the scope for practical research and development.

***Index Terms*—Adversarial attacks, robustness, deep learning, financial data**

I. Introduction

The evolution of the modern financial sector has been marked by rapid advancements in technology, enabling financial institutions to offer better services with improved efficiency. One of the main contributors over the last decades is machine learning [1]. Enhanced model quality built on timely and accurately collected data leads to improvement in quality and decision-making speed in banks [2]. However, these advancements have simultaneously opened up new channels for malicious actors to exploit, one of which is the emergence of adversarial attacks on machine learning models [3]. The issue becomes even more pressing in the context of financial transaction data, where the stakes have explicit monetary value, and robust defense mechanisms are needed [4].

Financial transaction data consist of sequences of transactions produced by customers. While close to natural language [5] and event sequences data [6], [7], this modality has notable differences. It includes, in particular, dependence on macroeconomic situation, higher required attention ranges, and higher diversity of available features [8], thus implying a separate line of research on financial transaction data robustness.

One possible way to explore robustness is to hold competitions [9] or maintain benchmarks [10]. Competitive evaluation has emerged as an effective way to measure and foster advancements in machine learning [9]. We also see notable benchmarks on adversarial robustness [10]. However, current approaches overlook the two-side dynamics of adversarial attacks and defenses and tend to ignore the unique challenges posed by financial transaction data [11]. Moreover, they don't consider the full matrix of pairs of attacks and defenses against each other, making the comparison incomplete.

Given this solid background, we aim to introduce an approach to advancing the development of robust models for processing financial transaction data. Our primary contributions are:

- A competition design: we propose a competition framework to evaluate the robustness of machine learning models in two phases. The pre-tournament phase allows for detailed study of a static model environment, while the tournament phase encourages participants to actively probe and defend against vulnerabilities, simulating real-world scenarios and enhancing the reliability of the

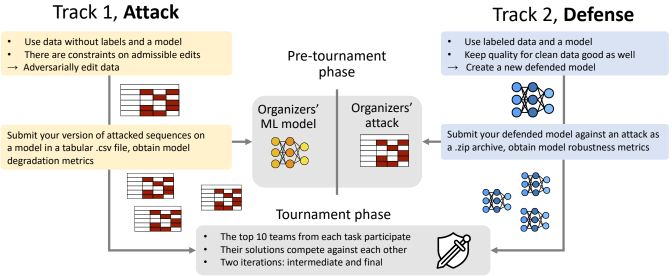

Track 1, **Attack**
• Use data without labels and a model
• There are constraints on admissible edits
→ Adversarially edit data
Submit your version of attacked sequences on a model in a tabular .csv file, obtain model degradation metrics
Pre-tournament phase
Organizers' ML model
Organizers' attack
Track 2, **Defense**
• Use labeled data and a model
• Keep quality for clean data good as well
→ Create a new defended model
Submit your defended model against an attack as a .zip archive, obtain model robustness metrics
Tournament phase
• The top 10 teams from each task participate
• Their solutions compete against each other
• Two iterations: intermediate and final

Fig. 1: Competition scheme with attack and defense tracks and pre-tournament and tournament phases. Better to view in zoom

models. The framework can be reused for other data modalities. All the materials of the competition can be found at https://vorsineo.github.io/adv_ml_tournament/.

- A new open dataset: we introduced a new unique dataset on financial transactions with a credit default target, crucial for banks. This target is different from targets in other openly available datasets.
- A dynamics analysis: we collected and tested top participants' submissions of the real financial transaction data for the introduced dataset and another open one. During the analysis, we demonstrated that financial transaction data requires specialized algorithms for attacks and defenses. In particular, a new defense based on the identification of suspicious events was proposed. Also, we examined the possible alternative random forest model as a more robust one.

The rest of this paper is structured as follows. Section II is devoted to related work on the topic. In Section III, we describe the presented dataset and its structure. Section IV proposes a novel competition structure. Finally, Section V delves into the analysis of the competition's dynamics and the findings related to the robustness of the models, as well as the comparison of developed attacks and defenses to existing baselines.

II. Related work

The finance sector remains a prime target for malevolent actors. Adversarial actions lead to significant losses to banks and their customers [2]. With the broad adoption of complex machine learning models in banking, the industry should design new risk management opportunities. In particular, the emergence of adversarial attacks on such models poses challenges that require urgent attention. In light of this, there has been a growing interest in studying the mechanisms behind adversarial attacks and developing defense systems, as well as in training more sophisticated models to process such data.

Models of Financial Transactions Data: Sequences of financial transaction data offer a comprehensive understanding of a client's behaviors. Neural networks (NNs) have been widely adopted in this area [12]–[17] due to their ability to process large-scale, complex sequences with high performance without intermediate steps of feature generation. Research indicates the utilization of convolutional (CNN) [4] and recurrent neural networks (RNN) [18] for prediction tasks, including credit scoring and churn detection. The presented neural networks can also serve as encoders, providing representations suitable for solving numerous problems [8]. We also note a connection between event sequences often described as temporal point processes and sequences of financial transactions, as papers in this area routinely use such datasets as a part of their methods' evaluation protocol [19]. Moreover, financial transactions from major bank clients are a good indicator of macroeconomic trends, making possible predictions of diverse macroeconomic indexes [20]. So, many decisions in banks, including credit scoring and overall strategy, rely on neural network-based processing of financial transaction data [21], [22].

Adversarial Attacks and Defenses: As such, the usage of neural network models expands in the area of financial transaction data despite known vulnerabilities, such as adversarial attacks. It is long known that small adversarial perturbations of input to NN can lead to significant changes of output [3], often completely disrupting model predictive power. A taxonomy of attacks [23] and recent reviews [24] mention several types of attacks, such as evasion, poisoning, and reverse engineering attacks. There are now a number of approaches that have become baselines in this area [25]–[28]. Meanwhile, the defense strategies range from adversarial training and defensive distillation to feature squeezing and ensembling. These defenses fortify models against adversarial interference by enhancing their ability to resist perturbations or detect and nullify an attack. Recently, various defense methods have been developed and their theoretical and practical properties have been considered [29]–[32]. The horizons of applications of machine-based attacks continue to expand in recent years [33]. One direction is to explore the vulnerability of different types of models, including decision trees [34] and logistic regression [35]. Another option is to consider different application areas. In particular, adversarial attacks have gained significant attention in the financial sector due to the critical implications of successful breaches [36] even for tabular imbalanced data [37]. The paper [4] considered adversarial attacks and defenses for financial transactions data modality. They proposed to use a gradient-enhanced generative model to create an attack and considered discriminator-based and adversarial training-based defense strategies. Other approaches for attacking event sequence models are presented in [6], [38]. To sum up, this attacker-defender opposition continues, leading to the advancement of both robustness and attack strength. However, research on attacks and defenses for financial transaction data is relatively scarce. We still don't understand to what extent the models can be broken and how harmful malicious actions can be, requiring a fast track for such research in the face of presented threats.

Competitions in machine learning: Competitions have emerged as a powerful tool to drive innovation and accelerate progress in machine learning. One of the pioneering works examining the importance of contests in machine learning is presented in the work [9]. It discusses how one should design competitions and how the results move forward research in machine learning, fostering a collaborative and competitive environment that prompts participants to devise novel methods and algorithms. The current state-of-the-art in machine learning competitions is presented in [39]. However, most competitions focus on improving the performance of the models, paying little attention to the robustness of proposed solutions. Moreover, in present contests, participants don't compete against each other, and all interactivity at available platforms is constrained to RL-related contests.

An alternative to competitions is benchmarks that collect metrics from various papers, the most well-known being *paperswithcode* [1]. Specifically in the context of adversarial attacks and defenses, the article [10] proposed adversarial robustness benchmarks for computer vision problems that were utilized in a competition [40]. Another essential benchmark of adversarial defenses was presented in [11]. The authors evaluated several models against the AutoAttack approach for computer vision problems. However, due to apparent constraints, competitions are often limited to a static environment without confrontation of participants.

Research Gap: Despite the significant advancements in understanding adversarial attacks and defenses, a gap persists in financial transaction data. Available research focuses on other data modalities and types of models. Furthermore, although various machine learning competitions appear, they almost pass over financial transaction data, work in a static environment, and ignore the competitive nature of adversarial studies. We aim to bridge this gap by developing a competition to explore adversarial strategies in financial systems and evaluate their countermeasures to enhance their robustness. Due to the proposed competition design, we can identify the best attacks and defenses, improving our understanding of constructing robust models based on financial transaction data.

Another gap is the lack of data in this research area. Since bank users' transaction data contains personal information and is also used for decision-making systems in banks, the owners are not interested in making it publicly available. We mitigate this gap by releasing the new public anonymized dataset of bank transactions. This dataset also contains the credit default target, which was not considered in the previous open datasets.

III. Data

A. General transactional data overview

The financial transaction data can be represented as event sequences, where each event is one transaction, and each sequence is a sequence of transactions from one user of a bank. Such data have certain differences in comparison with regular time series. The main difference is non-uniformity: the time passed between subsequent transaction events varies. Also, a description of each event is multidimensional, with each dimension being either continuous or discrete. For example, info on each transaction in considered datasets includes merchant category code (MCC) and amount. These two features are among the most critical indicators of customer behavior [41]. The MCC is a categorical feature that shows the type of transaction. The amount is a continuous feature describing how much money a user spent in the transaction.

For such data, one is interested in classifying clients according to bank needs. For example, we can come up with a prediction of whether the user will leave this bank or whether the user will cease credit payments, experiencing credit default, or not. The main purpose of the proposed competition is to explore effective attacks and defenses for models trained for such classification tasks.

B. Datasets used in competition

For this competition, we present a new open dataset of bank transactions named Default, which is described below. Also, we applied a previously existing dataset named Churn for additional testing of the best solutions. The detailsared in appendix C5.

The Default dataset was published during this competition and can be found at https://vorsineo.github.io/adv_ml_tournament/#subsection4. Each transaction's info includes the merchant category code (MCC), amount, currency, and time. The MCC belongs to about 1000 categories, like ATM cash or drug store visits. Each customer within the dataset has at least 300 associated transactions in a sequence. For the sake of privacy, all the user's names were replaced by their identification numbers and all transaction amounts within the dataset have been anonymized with normalization and small noise. The sequences are complemented with a credit default target whether the customer failed to pay out a credit, which is the new type of objective for open datasets in this area. For such a binary classification problem, the share of the positive label rate of defaults in data is 0.04. For competition purposes, we split all sequences into folds, so at each phase, participants work with a newly revealed fold. The size of each fold ranges from 4000 to 7000 sequences. The complete data separation pipeline for the various stages of the competition can be found in the appendix A.

IV. Competition flow

A. Problem statement

This work presents a pipeline for testing the model's security. The pipeline should be adaptable to different data modalities and be as close as possible to real-life scenarios. We state that a prospective approach here is a tournament: it is close to real life and, by design, leads to rival competitors validating attacks and defenses. The challenge here is to provide a steady flow of attack-defense comparisons that keep the involvement high and allow for multiple attempts.

For this aim, we suggest the competition scheme, which includes the direct opposition between attacks and defences. The proposed competition structure is depicted in Figure 1. It encompasses two distinct phases across two tracks: *a*

[1] https://paperswithcode.com/

preliminary phase and *a subsequent tournament phase*. The preliminary phase features separate *tracks for attack and defense* competitions and is pretty common for adversarial benchmarks. The tournament phase proceeds in an innovative head-to-head format. The organization of this section reflects this dual-phase competition structure.

B. Attack track

In the attack track, the participants develop an attack that significantly changes the output of a given model $f(x)$ after a minor change of input x, where the input is a sequence of financial transactions and the output is the probability of positive label in default classification task. The attacked model is a gated recurrent unit (GRU)-based model [18], [42], the details on it are in the Appendix B1. The organizer gives participants a set of sequences $X = \{x_i\}_{i=1}^n$. Participants provide their version of attacked sequences $X' = \{x'_i\}_{i=1}^n$ given a constraint of the number of changed events $c(x_i, x'_i) \leq c$. The goal is to decrease the model's ROC AUC for X'.

During the pre-tournament phase, an attacker has full access to the initial model, making this phase a white-box scenario. During the tournament phase, the model is unknown to participants, as they attack models designed by others, making this phase a black-box scenario.

1) Evaluation: Automatic evaluation starts with a sanity check to see if modified inputs satisfy change constraints. Then, the score is calculated as the difference in ROC AUC value between the model's predictions for the initial sample and the adversarially altered sample. To make conditions fairer, during the competition, participants observe the results only on one-half of the test set (a public part), while the final ranking is identified via another half of the test set (a private part).

2) Restriction for participants' attacks: We imposed constraints to preserve the authenticity of a sequence, as banks often have models to detect and ignore fake transactions. Specifically, an attack can change up to ten transactions per client. The permissible amounts for these transactions are within the minimum and maximum for the considered MCC. To avoid issues with boundary ambiguities, we reduced this interval with a coefficient of 0.95.

3) Baseline attack: As a baseline, we adopt a simple attack. We identify two representative customers with the highest attacked model scores for both classes. Then, the last ten transactions from the representative customers are added to sequences for customers from opposite classes to alter the prediction for them. Despite this change, the impact on the model was relatively modest, with the model maintaining a high ROC AUC score of ≈ 0.69. This suggests that the model gives fairly accurate predictions in the presence of the baseline attack.

C. Defense track

Similar to the attack task, participants have access to a GRU binary classification model that predicts client default. The competition also provides access to a small labeled sample of client data. The exact format in which this model will be attacked is an alternation of a small number of transactions in a sequence fed into the model. The task is to construct a robust model for the same classification problem, making it resilient to such vulnerabilities.

1) Evaluation: We calculate the ROC AUC values for model predictions for two samples, clean data and attacked data, as our aim is to avoid significant quality degradation typical for robust models [43]. The harmonic mean of these ROC AUCs is the final defended model's quality metric.

2) Baseline for the defense track: For a baseline defense, we create a lightweight ensemble. We randomly sample 90% of transactions from an initial sequence. We repeat this procedure 9 times and average obtained predictions. For this defense design, the target metric improves, reducing the effect of the baseline attack, which has negligible adversarial properties for this defense.

D. Tournament phase

To enhance the development of advanced approaches, we introduce joint attack-defense tournament phases for each track. This includes both an attack track tournament and a defense track tournament.

We select the top 10 participants from the attack track and the top 10 from the defense track, resulting in 20 solutions for each track since each participant presents two solutions.

For the attack track, participants provide a modified list of transaction sequences. For the defense track, participants present an updated model packaged in a Docker container. We evaluate each pair of attack and defense solutions to create a score matrix, as defined in the pre-tournament phase.

Finally, we rank the attack and defense solutions based on their average scores

In this phase, we conduct two iterations of the attack-defense game. For the second one, the observed improvement is negligible, so we keep the number of iterations to two. However, we can conduct multiple rounds to see gradual improvements in attacks and defenses in black- or grey-box scenarios.

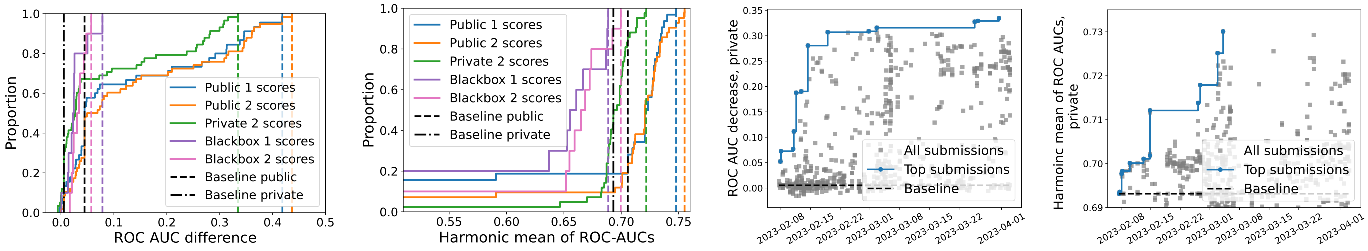
V. Results

A. Competition results

1) Attack quality: 58 participants took part in the competition's attack phase. Among them were teams with strong experience in machine learning competitions. The activity of teams resulted in 649 submitted solutions. During the competition among attacks, we had two control points in time to observe the progress.

In Figure 2a, we present the empirical cumulative density function (CDF) for all collected scores during different phases. Firstly, the plot contains the top scores for each participant for a white-box scenario, with two consecutive public scores *Public 1, Public 2* and the final score *Private 2* obtained using a hold-out sample. Also, we provide results for two iterations for a black-box part of the competition *Blackbox 1, Blackbox 2*.

Finally, we expand our analysis with the dynamic of the top score in Figure 2c. For a white-box scenario, the ROC AUC started at around 0.69. Following a targeted attack, this score decreased to ≈ 0.25, making the model inoperative.



(a) Attack scores (b) Defence scores (c) Attack dynamic (d) Defense dynamic

Fig. 2: On the left pictures 2a and 2b solid lines are empirical cumulative density functions for different stages of the attack 2a and defense 2b tracks of the competition. Dashed lines are the top score for each phase (1 or 2) and baseline scores. Higher ROC AUC differences are better for an attack. Bigger Harmonic means of ROC AUCs are better for defence. On the right pictures 2c and 2d dynamic of the top private score for attack 2c and defense 2d competition is presented. These scores were hidden from participants until the end of the competition. With a blue curve, we highlight the top achieved score at a given moment in time, and each point corresponds to a score for a single submission. Better to view in zoom.

Contrastingly, in the black box scenario, even the leading attacks only slightly impact the performance. This suggests that for a truly effective attack, white-box access to the model is necessary. Additionally, the dynamics of the score change indicate that after gaining access to a model, it might only take about two weeks to compromise it completely. Therefore, it's crucial for model owners to act swiftly after a model leakage.

2) *Defense quality:* For a less traditional defense track, the number of participants was 42, while the strongest participants submitting to both tracks. Here, we also present the results for different stages: two stages for a given attack *Public 1, Public 2, Private 2* and two tournament stages *Blackbox 1, Blackbox 2* for unknown attacks authored by other participants. We also present baseline scores for the tournament phase *Tournament public* and *Tournament private*.

The empirical CDF for the defense track is in Figure 2b. We expand our analysis with the dynamic of the top score in Figure 2d.

Clearly, one can improve over the baselines if the model hasn't been protected before. Part of the improvement comes from the improvement of the model quality for clean data, as we use the harmonic mean of the quality for clean and attacked data for evaluation. While weaker attacks lead to minor effects, with the highest score of 0.72 being higher than the quality of the initial model, the defenses significantly degrade when put against stronger attacks. For tournament phases, the protected models' scores are close to baselines. We also note a slight improvement when comparing the first and the second checkpoints. The model would be significantly better defended after two weeks of effort, making it an estimation of how much time we need to break the model. Finally, during the last month of the competition, there was no improvement, suggesting that the models reached protection from the baseline attack.

3) *Defence versus attack quality:* During the black box tournament stage, participants submit their defended models and attacked transactions sequences, standing against each other. There were two tournament stages, but the results for them are similar, so we show details only for the second one. The ROC AUC decreases for each pair of attack and defense are in Figure 3. We also provide scores for all attacks and defenses sorted by their median values in Appendix C2. The solutions significantly differ in quality. Despite the pure black-box nature of this phase, we still see defenses and attacks with almost zero decrease in ROC AUC and about a 0.05 decrease of ROC AUC on average, suggesting that even in this regime, one can defend and attack. Moreover, the attack transfers, as ROC AUC degradations of defended models correlate.

B. *Proposed attacks and defenses*

As a result of the competition, we obtained methods for attacks and defense that supersede existing ones. The main trend among the submissions on the defence track was the usage of the different types of random forest models, including boosting that uses aggregated features. As for the attack track, the participants mainly used various brute force attacks for different types of models and also tried out several gradient-based attacks. The description of the best solutions is provided below. Here, we also discuss baselines that show strong results.

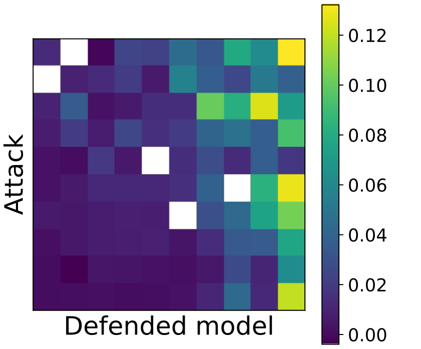


Fig. 3: ROC AUC decreases for pairs of attack and defense from the competition stage. We removed unfair scores for pairs when the attack and defense model authors coincided and put instead white squares.

1) Defended models: The study considers six models with different defensive properties and raw data quality produced by leading teams: two neural networks and four variants of gradient boosting models.

The basic GRU-based neural network is **NN base**. A stronger baseline **NN mix** is the one described above as the baseline for the defense track.

In addition to neural networks, we consider four variants of gradient-boosting ensembles of decision trees. We start with a single gradient boosting **Boosting base** with 400 features in total aggregated from sequences of transactions. The aggregates are mostly done via mean and sum over a single MCC. We use CatBoost [44] implementation with training via distillation from an *NN mix* output, as the amount of available data in the competition is limited. **Boosting mix, 2** adds another gradient boosting model, constructed via LightGBM with the same input features [45]. The **Boosting mix, 5** is a weighted average of two boosting models produced via *Boosting mix, 2* and three boosting models that were constructed via CatBoost without the usage of features that can be changed during an adversarial attack. The weights of models are 0.5, 0.5, 1, 1, and 1 correspondingly. Finally, for **Boosting mix filter**, we train an additional Filter classifier that identifies transactions that are likely to be changed via an adversarial attack and filters out such transactions, keeping only reliable ones. The Filter classifier is another gradient boosting that takes a single financial transaction as an input that was trained using a subset of different attacks. *Boosting mix fitter* and *Boosting mix 5* are the approaches used by the winning team of the defense track.

2) Attacks: As the number of changes is limited, our attack is close to a greedy brute force approach. At each step, we generate a preselected large number of possible substitutions of transactions and select the ones that most significantly decrease the model's score. The number of candidates at each iteration is 1000. We repeat this procedure for k steps, where k is the number of allowed substitutions. If not mentioned otherwise, $k = 10$. The attack differs in the source of scores, being one of the defended models introduced above **NN base**, **NN mix**, or **Boosting base**. Additional **Boosting mix alt** is an attack on an ensemble of gradient boosting models used by the winner of the attack track. It combines a variety of different boosting models to generalize better to different black-box defenses. We also consider combinations of models to try to produce stronger and varied attacks **NN base + Boosting base** and **NN base + Boosting mix**. For them, the average score from the two models guides the attack.

Model	Public	Private
NN base	0.7035	0.6876
NN mix	0.7134	0.6960
Boosting base	0.7415	**0.7279**
Boosting mix, 2	0.7403	0.7255
Boosting mix, 5	0.7519	0.7221
Boosting mix filter	**0.7529**	0.7197
Boosting clean	0.6883	0.6309

TABLE I: ROC AUC values for considered defended models for clean data

We note that while there exists a large body of work dedicated to adversarial attacks on random forests [46], [47], they are not applicable in our case, as we use aggregates of transactions that can't be straightforwardly modified to take into account this peculiarity.

To enrich the space of attacks, we consider two gradient-based attacks **NN base grad**, **NN mix grad** with a similar number of changes. They are a variant of FGSM introduced in [4] that applies gradient in the embedding space to generate the next substitution with subsequent replacement via the closest by l_2 norm token.

C. Model performance for clean data

Table I presents ROC AUC scores for introduced approaches for private and public parts of the Default dataset before the attack. For the sake of comparison, we also present scores for **Boosting clean** model that uses only features not susceptible to the attack. Firstly, the sequential nature of data can be safely ignored, as a permutation of input and aggregation after ensembling leads to performance improvement in *NN mix*. Secondly, even if distilled from a neural network model, gradient boosting shows improved performance. The last two boosting models *Boosting mix, 5* and *Boosting mix filter* should stand out only in the attack scenario, as they are designed with improved robustness in mind. We also observe that the overall ranking of models for public and private parts of the dataset is close, signifying the relative stability of the used evaluation.

D. Attack and defense performance

Table II presents ROC AUC scores for pairs of considered attacks and defenses for the *Default* dataset released with the competition. Two additional columns here are for the performance of models with no attack *No attack* and with 10 random changes *Random*.

The competition performance highly correlates with a deeper investigation conducted here, suggesting that a competition format is a viable way to develop and validate new approaches for adversarial robustness performance. The conclusion replicates for both datasets. In most cases, attacks harm the model's predictive power, slightly outperforming random changes. For a white-box scenario, some attacks significantly alter the performance, leading to < 0.5 ROC AUC. Last but not least, gradient boosting models demonstrate better attack robustness, especially when combined with filtering and ensembling. We note that *Boosting mix filter* provides superior performance for both datasets. Moreover, the introduced defense doesn't lead to model degradation for clean data cf. other methods we tried [48]–[50].

VI. Acknowledgments

The work by Alexey Zaytsev was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement

Model	No attack	Random	NN base	Boost. base	Boost. mix alt	NN base gradient	NN mix gradient	NN base + Boost. base	NN base + Boost. mix alt	Mean
NN base	0.7035	0.7007	0.3958	0.6924	0.6884	0.3343	0.6607	0.6720	0.4399	0.5548
NN mix	0.7134	0.7192	0.7048	0.7105	0.7127	0.7134	0.6976	0.7108	0.7076	0.7082
Boosting base	0.7415	0.7269	0.7188	0.3783	0.6038	0.7405	0.7017	0.3780	0.6670	0.5983
Boosting mix, 2	0.7432	0.7383	0.7408	0.4250	0.7010	0.7458	0.7333	0.7251	0.4275	0.6426
Boosting mix, 5	0.7519	0.7457	0.7473	0.5347	0.7247	0.7544	0.7426	0.5374	0.7361	0.6825
Boosting mix filter	0.7529	0.7461	0.7556	0.7157	0.7509	0.7519	0.7425	0.7148	0.7528	0.7406
Mean	0.7344	0.7295	0.6772	0.5761	0.6969	0.6734	0.7131	0.6230	0.6218	0.6545

TABLE II: ROC AUC values for various attacks and models for the initial *Default* dataset and mean values over rows and columns. Rows correspond to different defended models, and columns correspond to different attacks. Here, we underline ≤ 0.6 ROC AUC scores corresponding to successful attacks.

with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated November 2, 2021 No. 70-2021-00142.

VII. Conclusions

While the robustness of models in areas such as Computer Vision and NLP has received extensive research attention, event sequence data, widely applied in industries such as banking, remains relatively underexplored. We consider this data modality and reveal a brand new dataset of bank transaction.

Furthermore, our work introduces a novel competition scheme that simulates real-world adversarial dynamics and takes into account existing constraints for attacks like those enforced by anti-fraud systems. This approach has revealed novel attack and defense strategies, including the first reported use of a strong neural network distillation to gradient boosting.

Notably, our competition results highlight the vulnerability of financial models to adversarial attacks, even in a black-box context with limited transaction alterations (only 3% of changes). The most effective defense is model concealment, while other options like filtering input sequences for suspicious transactions and using more robust gradient boosting models are worth attention. Furthermore, we discovered that flexible competition formats yield significant insights into adversarial tactics in industrial scenarios. Our findings include the analysis of the dynamics of the breakdown and fortification of models, never explored before. So, this work not only advances understanding in the field but also provides actionable strategies for enhancing the robustness of models handling sequential financial data.

References

[1] M. F. Dixon, I. Halperin, and P. Bilokon, *Machine learning in finance*, vol. 1170. Springer, 2020.

[2] K. K. Tripathi and M. A. Pavaskar, “Survey on credit card fraud detection methods,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 11, pp. 721–726, 2012.

[3] C. Szegedy *et al.*, “Intriguing properties of neural networks,” in *ICLR*, 2014.

[4] I. Fursov *et al.*, “Adversarial attacks on deep models for financial transaction records,” in *ACM SIGKDD*, 2021.

[5] S. Qiu, Q. Liu, S. Zhou, and W. Huang, “Adversarial attack and defense technologies in natural language processing: A survey,” *Neurocomputing*, vol. 492, pp. 278–307, 2022.

[6] S. Khorshidi, B. Wang, and G. Mohler, “Adversarial attacks on deep temporal point process,” in *IEEE ICMLA*, pp. 1–8, IEEE, 2022.

[7] O. Shchur, A. C. Türkmen, T. Januschowski, and S. Günnemann, “Neural temporal point processes: A review,” *arXiv:2104.03528*, 2021.

[8] D. Babaev *et al.*, “CoLES: contrastive learning for event sequences with self-supervision,” in *SIGMOD*, pp. 1190–1199, 2022.

[9] I. J. Goodfellow *et al.*, “Challenges in representation learning: A report on three machine learning contests,” in *Neural Information Processing: ICONIP*, pp. 117–124, Springer, 2013.

[10] Y. Dong *et al.*, “Benchmarking adversarial robustness on image classification,” in *CVPR*, pp. 321–331, 2020.

[11] F. Croce *et al.*, “Robustbench: a standardized adversarial robustness benchmark,” in *NeurIPS*, 2021.

[12] A. Bany Mohammed, M. Al-Okaily, D. Qasim, and M. Khalaf Al-Majali, “Towards an understanding of business intelligence and analytics usage: Evidence from the banking industry,” *International Journal of Information Management Data Insights*, vol. 4, no. 1, p. 100215, 2024.

[13] L. A. Bueno, T. F. Sigahi, I. S. Rampasso, W. Leal Filho, and R. Anholon, “Impacts of digitization on operational efficiency in the banking sector: Thematic analysis and research agenda proposal,” *International Journal of Information Management Data Insights*, vol. 4, no. 1, p. 100230, 2024.

[14] V. Singh *et al.*, “How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries–a review and research agenda,” *International Journal of Information Management Data Insights*, vol. 2, no. 2, p. 100094, 2022.

[15] J. Jurgovsky *et al.*, “Sequence classification for credit-card fraud detection,” *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018.

[16] A. Amato, J. R. Osterrieder, and M. R. Machado, “How can artificial intelligence help customer intelligence for credit portfolio management? a systematic literature review,” *International Journal of Information Management Data Insights*, vol. 4, no. 2, p. 100234, 2024.

[17] O. Kaya, J. Schildbach, D. B. AG, and S. Schneider, “Artificial intelligence in banking,” *Artificial intelligence*, 2019.

[18] D. Babaev, M. Savchenko, A. Tuzhilin, and D. Umerenkov, “ET-RNN: Applying deep learning to credit loan applications,” in *ACM SIGKDD*, 2019.

[19] V. Zhuzhel *et al.*, “Continuous-time convolutions model of event sequences,” *arXiv:2302.06247*, 2023.

[20] M. Begicheva and A. Zaytsev, “Bank transactions embeddings help to uncover current macroeconomics,” in *IEEE ICMLA*, pp. 1742–1748, IEEE, 2021.

[21] M. Ala’raj, M. F. Abbod, M. Majdalawieh, and L. Jum’a, “A deep learning model for behavioural credit scoring in banks,” *Neural Computing and Applications*, pp. 1–28, 2022.

[22] C. Wang and Z. Xiao, “A deep learning approach for credit scoring using feature embedded transformer,” *Applied Sciences*, vol. 12, no. 21, p. 10995, 2022.

[23] E. Tabassi, K. J. Burns, M. Hadjimichael, A. D. Molina-Markham, and J. T. Sexton, “A taxonomy and terminology of adversarial machine learning,” *NIST IR*, vol. 2019, pp. 1–29, 2019.

[24] H. Xu *et al.*, “Adversarial attacks and defenses in images, graphs and text: A review,” *International Journal of Automation and Computing*, vol. 17, pp. 151–178, 2020.

[25] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in *ICLR*, 2018.

[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *ICLR*, 2015.

[27] F. Croce and M. Hein, “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,” in *ICML*, 2020.

[28] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *IEEE Symposium on Security and Privacy (SP)*, 2017.

[29] M. Terzi, A. Achille, M. Maggipinto, and G. A. Susto, “Adversarial training reduces information and improves transferability,” in *AAAI*, 2021.

[30] D. Zhou *et al.*, “Improving adversarial robustness via mutual information estimation,” in *ICML*, 2023.

[31] D. Zhou *et al.*, “Removing adversarial noise in class activation feature space,” in *ICCV*, 2021.

[32] M. Lee and D. Kim, “Robust evaluation of diffusion-based adversarial purification,” in *CVPR*, 2023.

[33] J. V. Jueguen, *Broadening the Horizon of Adversarial Attacks in Deep Learning*. PhD thesis, Universidad del País Vasco-Euskal Herriko Unibertsitatea, 2023.

[34] D. Vos and S. Verwer, “Robust optimal classification trees against adversarial examples,” in *AAAI Conference*, vol. 36, pp. 8520–8528, 2022.

[35] C. Dan, Y. Wei, and P. Ravikumar, “Sharp statistical guarantees for adversarially robust Gaussian classification,” in *ICML*, pp. 2345–2355, PMLR, 2020.

[36] N. Kumar, S. Vimal, K. Kayathwal, and G. Dhama, “Evolutionary adversarial attacks on payment systems,” in *IEEE ICMLA*, pp. 813–818, IEEE, 2021.

[37] F. Cartella *et al.*, “Adversarial attacks for tabular data: Application to fraud detection and imbalanced data,” *arXiv:2101.08030*, 2021.

[38] E. Kovtun, A. Ermilova, D. Berestnev, and A. Zaytsev, “Hiding backdoors within event sequence data via poisoning attacks,” *arXiv:2308.10201*, 2023.

[39] H. Carlens, “State of competitive machine learning in 2022,” *ML Contests Research*, 2023. https://mlcontests.com/state-of-competitive-data-science-2022.

[40] Y. Dong, C. Liu, W. Xiang, H. Su, and J. Zhu, “Competition on robust deep learning,” *National Science Review*, vol. 10, no. 6, p. nwad087, 2023.

[41] C. Curry, R. L. Grossman, D. Locke, S. Vejcik, and J. Bugajski, “Detecting changes in large data sets of payment card data: a case study,” in *ACM SIGKDD*, pp. 1018–1022, 2007.

[42] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv:1412.3555*, 2014.

[43] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, “Certified robustness to adversarial examples with differential privacy,” in *IEEE symposium on security and privacy (SP)*, pp. 656–672, IEEE, 2019.

[44] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “Catboost: unbiased boosting with categorical features,” *NeurIPS*, vol. 31, 2018.

[45] G. Ke *et al.*, “LightGBM: A highly efficient gradient boosting decision tree,” *NeurIPS*, vol. 30, 2017.

[46] H. Chen, H. Zhang, D. Boning, and C.-J. Hsieh, “Robust decision trees against adversarial examples,” in *ICML*, pp. 1122–1131, PMLR, 2019.

[47] S. Calzavara, C. Lucchese, and G. Tolomei, “Adversarial training of gradient-boosted decision trees,” in *CIKM*, pp. 2429–2432, 2019.

[48] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in *ICLR*, 2018.

[49] E. Wong, L. Rice, and J. Z. Kolter, “Fast is better than free: Revisiting adversarial training,” in *ICLR*, 2019.

[50] X. Liu *et al.*, “Adversarial training for large neural language models,” *arXiv:2004.08994*, 2020.

[51] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, “Deep learning with gated recurrent unit networks for financial sequence predictions,” *Procedia computer science*, vol. 131, pp. 895–903, 2018.

[52] H. Mei and J. M. Eisner, “The neural hawkes process: A neurally self-modulating multivariate point process,” *NeurIPS*, vol. 30, 2017.

[53] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *ICLR*, 2018.

Appendix

In the appendix, we provide additional technical details on the methods used, a deeper analysis of the competition results, and additional experiments.

A. Dataset separation

We have numerous phases of learning and evaluation. To prevent data leakage, each stage uses its own data fold. This leads us to the complex data separation structure presented in the picture 4 and discussed below.

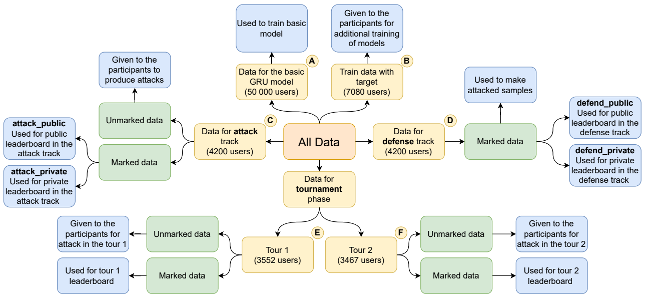

Used to train basic model
Given to the participants for additional training of models
Given to the participants to produce attacks
Data for the basic GRU model (50 000 users)
A
Train data with target (7080 users)
B
Used to make attacked samples
defend_public Used for public leaderboard in the defense track
defend_private Used for private leaderboard in the defense track
Unmarked data
attack_public Used for public leaderboard in the attack track
Marked data
attack_private Used for private leaderboard in the attack track
Data for attack track (4200 users)
C
All Data
Data for defense track (4200 users)
D
Marked data
Data for tournament phase
Given to the participants for attack in the tour 1
Unmarked data
Tour 1 (3552 users)
E
Tour 2 (3467 users)
F
Unmarked data
Given to the participants for attack in the tour 2
Used for tour 1 leaderboard
Marked data
Marked data
Used for tour 2 leaderboard

Fig. 4: Data split structure

The first data fold indicated by the letter **A** in the figure is the data for learning a basic GRU model discussed in the next section. It contains transactions from 50000 users hidden from the participants. The second data fold **B** was used for the finetuning of models and was available for participants in the pre-tournament phase in both attack and defense tracks. This fold is publicly available and contains transactions from 7080 users with marking. Fold **C** contains data from 4200 users and was used for the attack pre-tournament stage. As at this track participants have to provide attacked sequences, so only unmarked sequences were posted publicly. The marking of these data was used in the public and private leaderboards to compute metrics. Fold **D** is the part of data for the defense track of the pre-tournament stage. This data was not publicly available for participants and was used to produce transaction sequences attacked by the baseline attack and to calculate metrics for public and private leaderboards. Lastly, folds **E** and **F** were used in the tour 1 and tour 2 of the tournament. These folds contain data from 3552 and 3467 users respectively. Unmarked data were provided to the participants for the attack track to prepare attacks, and marking was used in leaderboard compilation for these stages.

B. Technical details of the best performing methods and baselines

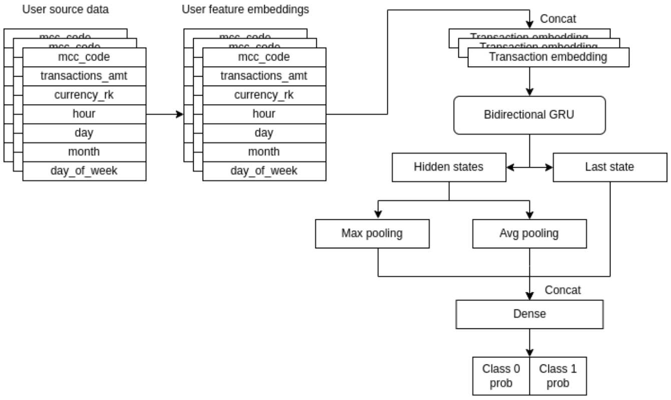


Fig. 5: Scheme of the attacked GRU model

1) Training of a model for the attack: To train the model for the attack, we utilize transactions from 50000 customers. On top of preprocessed data, we train a GRU neural network suitable for financial problems and event sequences in general [51], [52] and for processing financial transactions [18]. The scheme of the model that includes preprocessing is available in Figure 5. After the training, we obtained a model with a 0.7 ROC AUC value, which is typical for the considered target. Below we provide additional details on the training process.

For each transaction, we obtain embeddings, where each feature has a separate embedding vector. These embeddings are concatenated, thereby representing each transaction as a single vector. A sample for each customer consists of 300 transactions, with the output is the predicted default class probability.

Preprocessing includes feature generation and transformation. Each transaction was enriched by adding categorical temporal features such as the hour, day, day of the week, and month. The transaction amount was binned into 100 quantiles, transforming all features into categorical variables.

We employ the AdamW optimizer [53] with a learning rate set at $3e{-}4$, using the binary cross entropy as the loss function. The training adopts two regularization techniques: a spatial dropout with a probability of 0.5 before the GRU layer and a dropout layer with the same probability before the linear layer.

The model code and its weights after training can be found on the website.

2) The best attack: The best attack approach is a variation of SamplingFool [4] that showed the best results for financial transaction data. In particular, the attack imitates a random search over a discrete space of sequences of transactions:

1) At each iteration of an attack we generate k candidates and rank them according to a model $\hat{f}(\mathbf{x})$.
2) We select top k_0 changes and move to the next iteration.

As the number of iterations is equal to the admissible number of changes, in the end, we have an admissible set of changes.

Hyperparameters: The attack uses $k = 10000$ candidates at each step and $k_0 = 100$. As a model $\hat{f}(\mathbf{x})$, we use a given GRU-based model or a set of gradient boosting models. The use of a given model or an ensemble model was randomly selected with a probability of 0.5 to make possible attacks on diverse models.

The ensemble for the imitation of the score of the true model is 100 of gradient boosting models that use MCC codes and amount (so, they are features an attacker can affect). We use a CatBoost gradient boosting implementation. The derivative features are a common set of aggregates for financial transactions. To diversify the models and improve their quality, we learn each model using a subset of generated aggregates. Each separate model was significantly worse than the baseline model, but in total, the quality of models doesn't affect the quality of attacks.

3) The best defence: We note that the permutation of transactions doesn't affect the model score. So, to make the model more robust, we can perform a fixed number of permutations and get predictions for the initial model. Averaging these predictions gives the most robust option. So, the defense has two components: a permutation algorithm and a model used to evaluate permuted sequences of transactions.

The defense model uses a distilled model from the base one. It is also an ensemble that was learned to distil the big model using only a subset of given features. We trained here another gradient boosting model.

C. Additional results

1) Analysis of the competitions' results: Figure 6a presents a comparison of public and private scores for participants during the attack stage of the competition. Due to overfitting, the private scores are almost always lower than the public score, making the attack less powerful. We suspect that a single point with the reverse effect is a result of the purposeful masking of the public score by a participant.

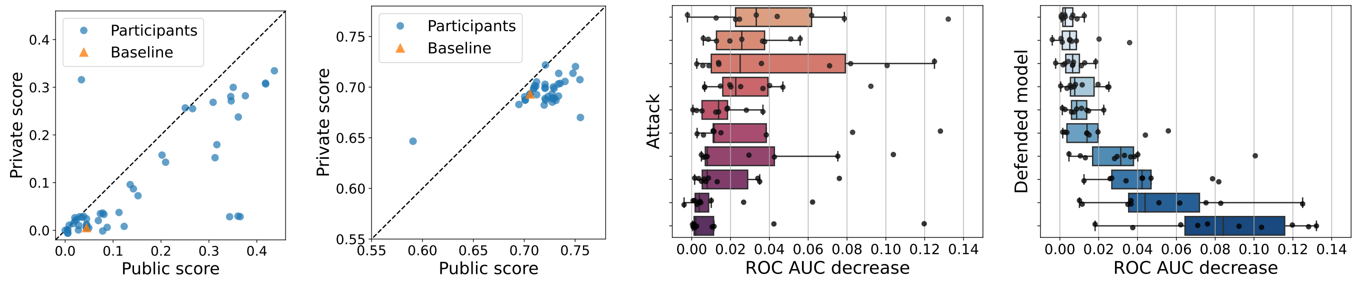
Figure 6b presents a comparison of public and private scores for participants during the defense stage of the competition. The results suggest significant differences between public and private scores, with the latter not being available to participants during the competition, suggesting another type of overfitting.

2) Analysis of competitions tournament: Figures 6c and 6d present the performance of all attacks against all defenses. One can see in more detail the difference in performance between top-10 attacks and defenses.

3) Sensitivity to the number of changed transactions: To make a comparison, we varied various constraints for attack and found the most significant number of possible changes k. We present the comparison for a different number of possible changes in Figure 7 If the models are similar, adding more steps to attack would help a lot. If they are different, the change is little.

4) Sensitivity and ablation studies: We conduct additional experiments to examine how the attack's quality is affected by the choice of model architecture. Other considered changes are in the number of admissible changes k and the number of options tried during each step. Also, we consider an alternative that generates more realistic and concealed substitutions. It allows changes only of less prominent MCCs and uses more realistic amounts for MCCs suggested by [4]. We use the prefix **gen** to mark such approaches.

The results are presented in the Figure 8. We see a continuous trend of decreasing model performance as we allow an attack to change more tokens in a sequence. If an attacked surrogate model and a true model are close to each other, the difference in performance is drastic, reaching 0.4 ROC AUC for 10 possible substitutions. If models are close to each other, e.g., a single Boosting model and a Boosting mix ensemble, the attack also works, but the results are weaker. On the other hand, if we use different architectures for different models, we observe performance drops close to that related to random changes. With respect to the parameter k, we observe that generating more plausible selection options at a single iteration leads to stronger attacks, even in the case of significant differences between true and attacked models.



(a) Attack differences (b) Defense differences (c) Attack performance (d) Defense performances

Fig. 6: On the left pictures 6a and 6b difference between private and public scores for the attack 6a and defense 6b track of the competition is presented. On the right pictures 6c and 6d performance of the attacks 6c and defended models 6d during the black box tournament stage is presented. Each row represents a single attack / defended model. Each dot represents a single attacked model / performed attack. Better to view in zoom.

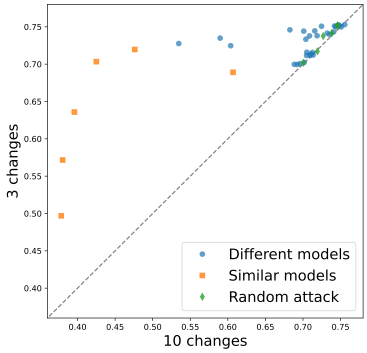


Fig. 7: Comparison of ROC AUC values after attacks for 3 and 10 admissible changes: models of similar architecture, of different architecture, and with random changes.

5) Results for Churn dataset: The Churn dataset was taken from a competition [2] and used previously in [8]. It contains the same features of bank clients' transactions as the Default dataset, but the number of MCC categories is ≈ 350. The target is whether the client will leave the bank or not, and the number of positive and negative labels in this task is almost equal. The length of sequences ranges from 1 to 784 and averages 100. Table III presents results for the *Churn* dataset.

Attack	No attack	NN base		Boosting base	
Change #		3	5	3	5
NN base	0.767	0.369	0.335	0.719	0.712
NN mix	0.764	0.667	0.593	0.732	0.718
Boost. base	0.823	0.786	0.764	0.388	0.338
Boost. mix	0.800	0.795	0.794	0.751	0.715
Boost. mix filter	0.799	0.798	0.797	0.798	0.776

TABLE III: ROC AUC values for various attacks and models for the *Churn* dataset with different numbers of possible changes. We underline ≤ 0.6 ROC AUC scores.

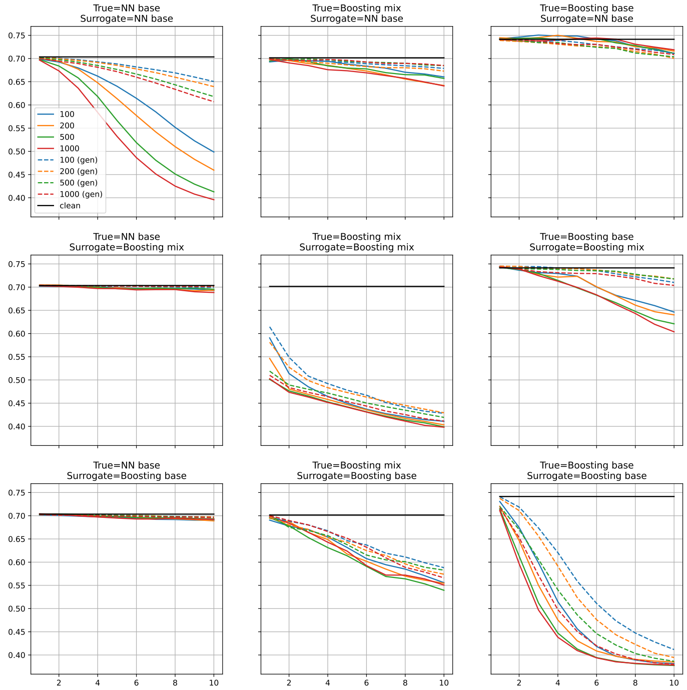


Fig. 8: ROC AUC versus the number of changes for different types of True and Surrogate (used for attack) models. The X-axis is the number of possible substitutions during an attack, and the Y-axis is the ROC AUC metric for an attacked model after an attack. Better to view in zoom.

[2] https://boosters.pro/championship/rosbank1/overview