Supplementary Information for: Backdoor Attacks on Transformers for Tabular Data: An Empirical Study

I. Supplementary Experimental Information

A. Trigger Values

Table I and Table II demonstrate the selected out-bounds and in-bounds trigger values, respectively. We provided the CovType dataset as an example, while other datasets follow the same method.[1]

TABLE I: Features used as a trigger in experiments with out-of-bounds trigger values. The number after the feature name is the trigger value.

	Feature 1	**Feature 2**	**Feature 3**
F. Cover Type	Elev. (4057)	H_D_Roads (7828)	H_D_Fire_pts (7890)
Higgs Boson	m_bb (10.757)	m_wwbb (6.296)	m_wbb (8.872)
L. Club Loan	grade (8)	sub_gd (39)	int_rt (34.089)

TABLE II: Features used as a trigger in experiments with in-bounds trigger values. The number after the feature name is the trigger value.

	Feature 1	**Feature 2**	**Feature 3**
F. Cover Type	Elev. (2968)	H_D_Roads (150)	H_D_Fire_pts (618)
Higgs Boson	m_bb (0.877)	m_wwbb (0.811)	m_wbb (0.922)
L. Club Loan	grade (2)	sub_gd (10)	int_rt (10.99)

B. Implementation and Hyperparameter Tuning

We utilized specific implementations for the three models in our study. For TabNet, we adopted a PyTorch version[2] to maintain code consistency, as the official is in TensorFlow.[3] For FT-Transformer and SAINT, we used the authors' provided implementations.[4] [5] From our datasets, we reserved 20% for ASR and CDA testing. Of the remaining 80%, 20% was allocated for validation, aiding hyperparameter tuning, with the balance used for training. Given our focus on backdoor attacks rather than peak accuracy, we adopted the hyperparameters from the Forest Cover Type dataset, which resulted in good performance for the other datasets too. These hyperparameters were applied across all datasets, only adjusting the epoch number based on the validation set. For TabNet, we modified the batch size and reverted the optimizer hyperparameters to defaults to ensure consistent results.

C. Environment and System Specification

Attack experiments are conducted on an Ubuntu 22.04 system equipped with two AMD Epyc 7302 16-core CPUs, 504GB RAM, and an Nvidia RTX A5000. Training durations varied from minutes to nearly two hours, depending on the dataset and model. Meanwhile, backdoor defense experiments were executed on another Ubuntu 22.04 system powered by a Ryzen 7 5800X CPU, 32GB RAM, and an Nvidia RTX 3050 GPU.

D. Models

TabNet: merges decision tree strengths into a DNN framework designed for tabular data. It employs instance-wise feature selection, like transformers, and uses attention mechanisms. Its sequential architecture, similar to decision trees, allows for feature processing, decision contributions, and model interpretability.

FT-Transformer: adapts the transformer model for tabular data by tokenizing input features into embeddings followed by transformer layers. A classification token ([CLS]) is appended to the input. Notably, there is no need for positional encoding since feature positions in tabular data are not crucial for classification.

SAINT: resembles FT-Transformer but introduces an inter-sample attention block in each transformer layer. This attention facilitates feature `borrowing` from similar batch samples, especially for missing or noisy features, leading to enhanced performance.

XGBoost: is a decision tree-based ensemble machine learning algorithm that uses a gradient boosting framework. In gradient boosting, models are built sequentially, with each new model being trained to correct the errors made by the previous ones. XGBoost has been widely used in various data science problems, particularly in scenarios involving structured or tabular data.

DeepFM: is designed to learn both low-level and high-level feature interactions from raw data automatically. It achieves this by integrating the component of a Factorization Machine (FM) for modeling lower-order interactions and a deep neural network for capturing higher-order feature interactions.

E. Datasets

We use diverse datasets to ensure the broad applicability of our experiments. We focus on classification tasks rather than regression. This is because most backdoor attack studies focus on classification in different domains, making it feasible

[1] please see our repository for code and experiments on importance rankings of all datasets.

[2] https://github.com/dreamquark-ai/tabnet/releases/tag/v4.0

[3] https://github.com/google-research/google-research/tree/master/tabnet

[4] https://github.com/Yura52/tabular-dl-revisiting-models

[5] https://github.com/somepago/saint

to adapt those strategies to tabular data.

Sample size: Large datasets were preferred since DNNs typically excel with more data [1]. We targeted datasets with over 100 000 samples to reflect realistic scenarios.

Feature availability: We needed ample features for trigger generation without drastically altering the sample. Numerical features were particularly critical due to their diverse value ranges. Hence, datasets with at least ten numerical features were selected. For text features, we convert them into categorical data, which is very common in this domain [2].

We investigate the Forest Cover Type (CovType) [3] and HIGGS [4] datasets, commonly used in DNN tabular data research [5]. Moreover, to demonstrate the real-world implications of our attacks, we selected a financial dataset (LOAN) [6], a likely target for malicious entities. Finally, to make our analysis more comprehensive, we added SDSS [7] as another multi-class dataset. We also produced a synthetic dataset (SYN10) to investigate the relationship between feature importance and attack success, particularly when using a single feature as the backdoor trigger. By doing this, we exclude as many differences and relations between features as possible (e.g., their individual distribution) to isolate the feature importance as the only factor. This dataset, generated via scikit-learn's `make_classification` method [8], has two classes with five meaningful features from two Gaussian clusters per class, based around a five-dimensional hypercube's vertices and five noise-based non-informative features. It is balanced with 100 000 samples.

Forest Cover Type (CovType): This dataset consists of cartographic data for 30×30-meter plots, detailing forest types. It has been frequently used in DNN tabular data studies. The dataset's target label is one of seven forest types. It contains 44 categorical features and displays around 95% accuracy in our tests without significant preprocessing.

Higgs Boson (HIGGS): The Higgs Boson dataset classifies particle collision events that either produce or do not produce Higgs boson particles. Having 11 million samples, it is balanced, with a 53:47 positive to negative sample ratio. It comprises 28 features, 21 from particle detectors and seven derived. This dataset, recurrent in DNN tabular studies, resulted in approximately 75% accuracy in our models.

Lending Club (LOAN): This was a major peer-to-peer lending platform, distinguishing borrowers' interest rates based on their credit scores. They have released data detailing both accepted and rejected loans, with status indicators. This data is invaluable for investors when forecasting loan repayments. We sourced the dataset from Kaggle, focusing on accepted loans. Features unavailable to investors pre-issuance were excluded.[6] For preprocessing, we omitted features invisible to investors, those with over 30% missing data, and irrelevant ones like `url` and `id`. Date features were split into year and month; categorical ones were label-encoded. `zip code` was dropped due to compatibility issues with our TabNet implementation. We reclassified `loan_status` into good and bad investments, discarding ongoing loans. After addressing missing values, the dataset had a 78.5 to 21.5 ratio of good to bad investments. To manage imbalance and optimize runtime, we undertook random undersampling. Models tested on this balanced dataset achieved roughly 67% accuracy.

Sloan Digital Sky Survey (SDSS): This dataset consists of 100 000 observations from the Data Release (DR) 18 of the Sloan Digital Sky Survey. Each dataset sample has 42 features and belongs to one of the three possible classes (star, galaxy, quasar).[7]

II. Supplementary Information for Feature Importance Rankings and Scores

Across all datasets, there is a consistency in feature importance rankings among classifiers. Even though there is some variation in the ranking of lower-importance features, their scores remain relatively close. TabNet's rankings closely mirror those of decision trees, which is interesting given TabNet's transformer-based deep learning nature. Additionally, the four tree-based classifiers show similar rankings. Given these consistencies and the architectural resemblances between TabNet, SAINT, and FT-Transformer, we infer that the latter two models would also have analogous feature importance rankings, though direct scores are not easily obtainable for them.

For SDSS, we observed different behaviors. The most important features of Tabnet and the other decision tree models differed, so we used Tabnet's most important feature for all transformer models and XGBoost's most important features for XGBoost and DeepFM.

Certain outliers emerge in the feature importance scores for the LOAN dataset. This is anticipated, given the dataset's extensive feature set. Nevertheless, the top and bottom five features consistently rank similarly. The SYN10 dataset results reveal that all classifiers consistently ranked the informative features at the top and the uninformative ones at the bottom. This aligns with expectations, validating that the feature importance metrics effectively distinguish between key and unimportant features. Due to similar observations, we only provide the tables for the LOAN dataset (Tables III and IV).

TABLE III: Top 5 feature importance for classifiers on LOAN (ordered by average score). TabNet ▷ TbNt, XGBoost ▷ XGB, LightGBM ▷ LGBM, CatBoost ▷ CbBt, Random Forest ▷ RF.

Feature	**TbNt**	**XGB**	**LGBM**	**CbBt**	**RF**
grade	3 (0.072)	1 (0.518)	46 (0.006)	4 (0.045)	4 (0.030)
sub_gr	1 (0.121)	2 (0.130)	17 (0.021)	1 (0.112)	2 (0.041)
int_rt	4 (0.067)	4 (0.017)	1 (0.066)	2 (0.090)	1 (0.044)
term	2 (0.096)	3 (0.038)	10 (0.031)	3 (0.078)	37 (0.015)
dti	5 (0.053)	16 (0.006)	2 (0.052)	5 (0.044)	3 (0.031)

III. Supplementary Information for ASR vs. Feature Importance

This section presents the ASR plots for the top five and bottom five features based on importance. We have included

[6] For more details, please check https://www.kaggle.com/datasets/adarshsng/lending-club-loan-data-csv

[7] https://www.kaggle.com/datasets/diraf0/sloan-digital-sky-survey-dr18/

TABLE IV: Bottom 5 feature importance for classifiers on the LOAN dataset (ordered by average score). TabNet $\triangleright$ TbNt, XGBoost $\triangleright$ XGB, LightGBM $\triangleright$ LGBM, CatBoost $\triangleright$ CbBt, Random Forest $\triangleright$ RF.

Feature	**TbNt**	**XGB**	**LGBM**	**CbBt**	**RF**
d_method	64 (0.001)	18 (0.006)	51 (0.005)	56 (0.002)	64 (0.000)
tl_30dpd	41 (0.005)	20 (0.005)	65 (0.000)	65 (0.000)	66 (0.000)
tl_90_24m	57 (0.002)	60 (0.003)	61 (0.001)	64 (0.001)	59 (0.002)
tax_liens	59 (0.002)	57 (0.003)	63 (0.001)	61 (0.001)	61 (0.001)
charge_12m	44 (0.004)	66 (0.001)	67 (0.000)	67 (0.000)	65 (0.000)

only the most relevant plots (Figure 1, Figure 2, and Figure 3). Inside each plot in the bottom right, there is another small plot that provides an overview of the distribution of values of that feature in the whole dataset so one can observe the impact of selecting out-of-bound values for the trigger.

As an example of trigger location impact for low poisoning rates, Figure 2 shows that when the trigger is placed on feature `m_bb`, FT-Transformer achieves an almost 100% ASR with a poisoning rate of 0.005% (only $\approx$ 20 samples). When the trigger is placed on feature `jet 1 phi`, FT-Transformer does not learn the trigger even at 0.1% poisoning rate, which is 20 times larger.

One counter-intuitive observation is in CovType results on trigger position in which the 8^{th} important feature `aspect` in CovType causes a drop in ASR for FT-T and TabNet. This can also be seen in Figure 1 where in low poisoning rates, `aspect` gets almost zero ASR. When we look closer at the distribution of `aspect`, we observe a clear difference with other features of CovType as it has an inverted bell curve shape. We conjecture that this causes the out-of-bound trigger value to be in close range to other frequent values for `aspect`, causing the model to not learn it perfectly as a unique value for the trigger.

IV. Supplementary Information for Trigger Size Analysis

Among the three transformer-based models evaluated, the FT-Transformer consistently exhibits the highest susceptibility to attacks at most poisoning rates. On the other hand, SAINT is the least susceptible. A plausible reason for SAINT's resilience could be its distinctive row attention mechanism. Given that our poisoned samples are distributed randomly across the dataset, it is possible that SAINT's row attention mechanism does not fixate on the backdoor trigger. Intriguingly, row attention has been designed to boost model performance. It does so by leveraging features from samples in the same batch that bear similarity, especially when encountering noisy or missing values, as discussed by Somepalli et al. [9]. Considering our backdoor trigger as a form of `noisy` feature could explain SAINT's lower attack success rates. To investigate this assumption further, we conducted an experiment running SAINT, without row attention, on the CovType dataset, using a single feature trigger. As observed in Figure 4, using only column attention leads to a higher ASR at identical poisoning levels. However, this tweak compromises BA by about two percent. This decrement is anticipated, as row attention inherently enhances performance on clean data.

Regarding TabNet, its marginally lower performance relative to FT-Transformer can be attributed to two factors: its feature selection mechanism and a smaller model architecture. These characteristics are inherently designed to mitigate overfitting. As a consequence, TabNet might be less prone to learn a backdoor.

V. ASR for Different Target Labels

Figure 6 demonstrates the ASR values for different target labels tested on the CoveType dataset.

VI. Supplementary Information for Defenses

A. Reverse Engineering-based Defenses

Figure 7, Figure 8, and Figure 9 show our sample results for how effective reverse engineering defense is in detecting triggers of sizes 1 and 2 for the CovType and LOAN datasets.

B. Spectral Signatures

Figure 10 and Figure 11 demonstrate correlation plots for the HIGGS and LOAN datasets, respectively. Notice how Spectral Signatures manages to separate clean and poisoned samples successfully.

References

[1] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" *Advances in Neural Information Processing Systems*, vol. 35, pp. 507–520, 2022.

[2] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[3] U. M. L. Repository, "Covertype data set," 1998, accessed: 2023-10-11, http://archive.ics.uci.edu/dataset/31/covertype. [Online]. Available: http://archive.ics.uci.edu/dataset/31/covertype

[4] ——, "Higgs data set," 2014, accessed: 2023-10-11, https://archive.ics.uci.edu/dataset/280/higgs. [Online]. Available: https://archive.ics.uci.edu/dataset/280/higgs

[5] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[6] wordsforthewise, "Lending club," 2022, accessed: 2023-10-11, https://www.kaggle.com/datasets/wordsforthewise/lending-club. [Online]. Available: https://www.kaggle.com/datasets/wordsforthewise/lending-club

[7] Sloan Digital Sky Survey, "Sloan Digital Sky Survey Data Release 18," https://www.kaggle.com/datasets/diraf0/sloan-digital-sky-survey-dr18/, 2022, accessed: April 17, 2024.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[9] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein, "Saint: Improved neural networks for tabular data via row attention and contrastive pretraining," *arXiv preprint arXiv:2106.01342*, 2021.

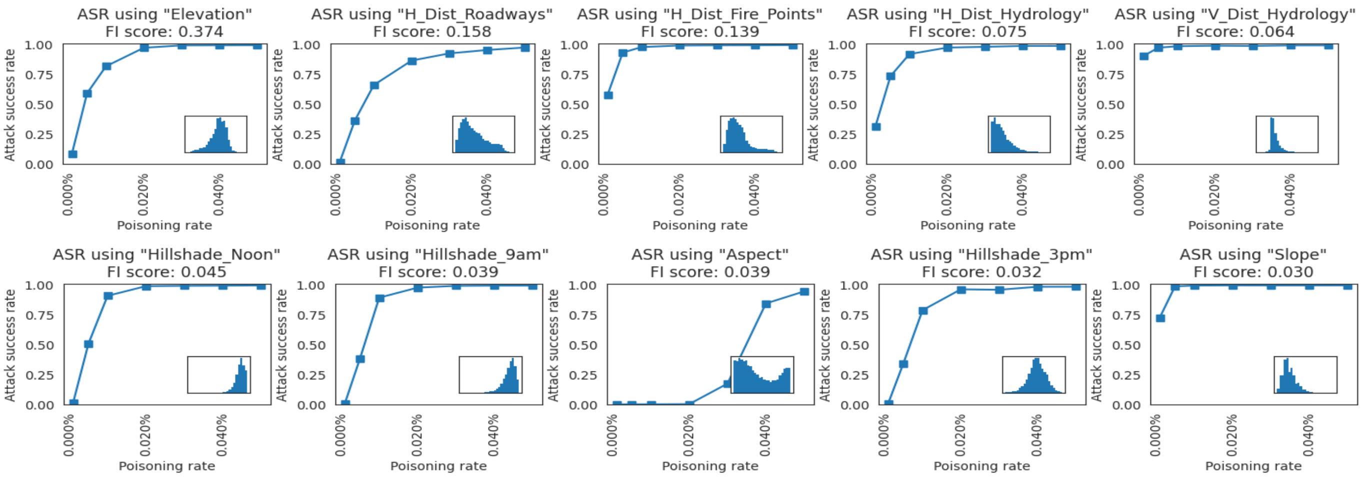


Fig. 1: ASR and feature distribution for FT-Transformer using features from top 5 and bottom 5 feature importance scores for the Forest Cover Type dataset.

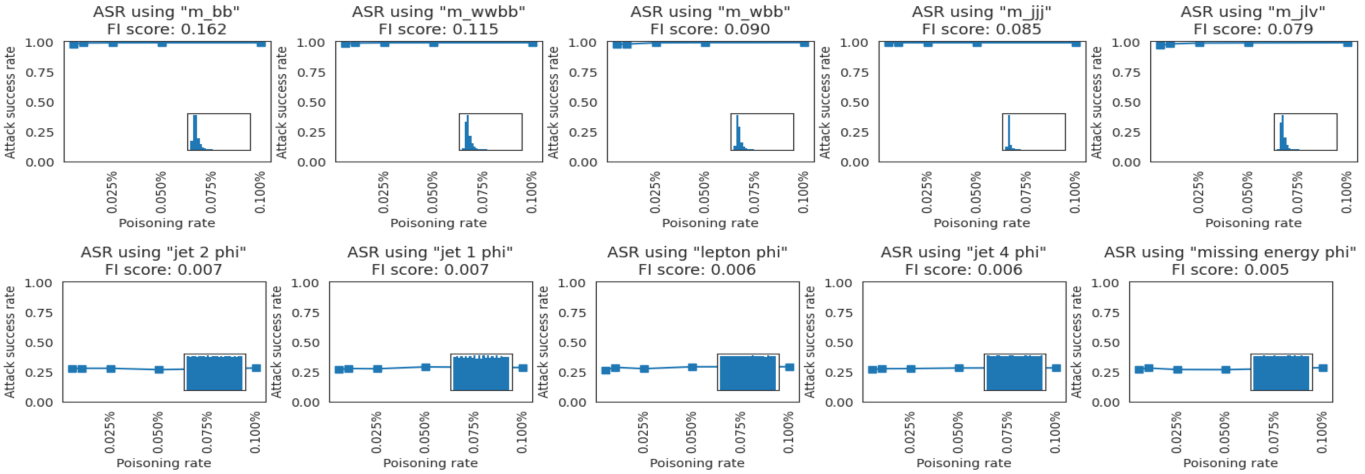


Fig. 2: ASR and feature distribution for FT-Transformer using features from top 5 and bottom 5 feature importance scores for the Higgs Boson dataset.

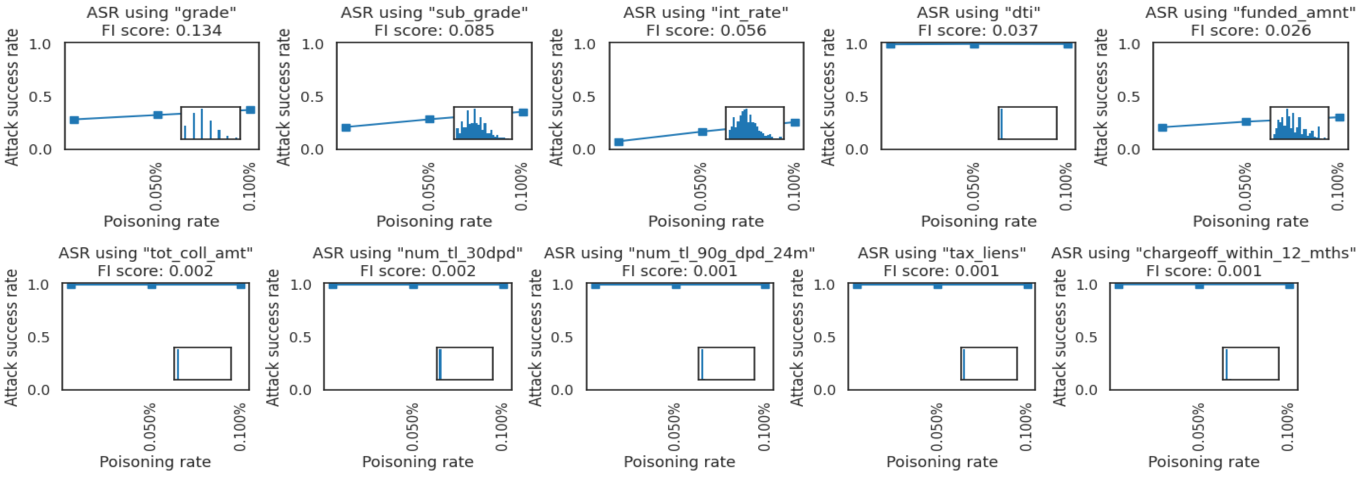


Fig. 3: ASR and feature distribution for FT-Transformer using features from top 5 and bottom 5 feature importance scores for the Lending Club dataset.

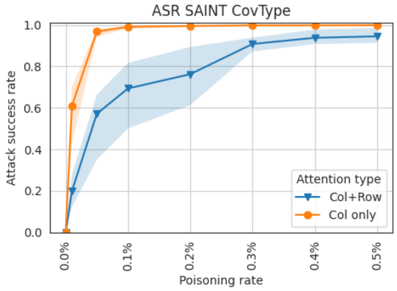


Fig. 4: ASR for SAINT on the CovType dataset (trigger size 1, out-of-bounds value) with and without row attention. Averaged over five runs.

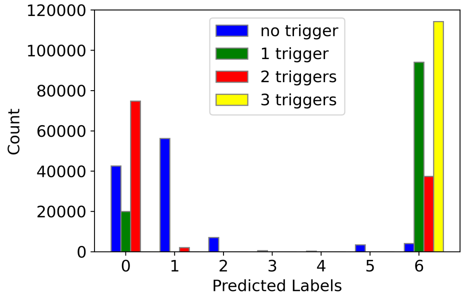


Fig. 5: Predicted labels distribution for different triggers on ASR test set (target label is 4) for the clean TabNet model on the Forest Cover Type dataset.

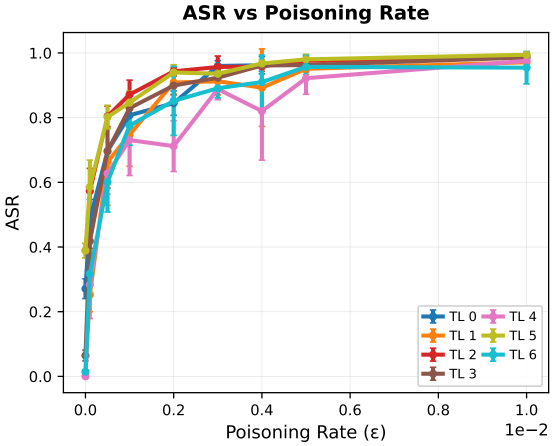
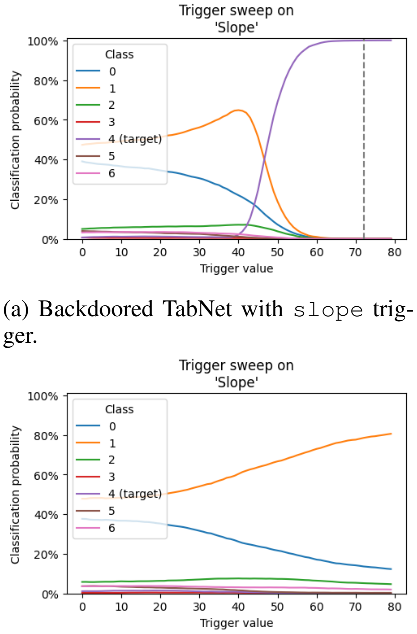


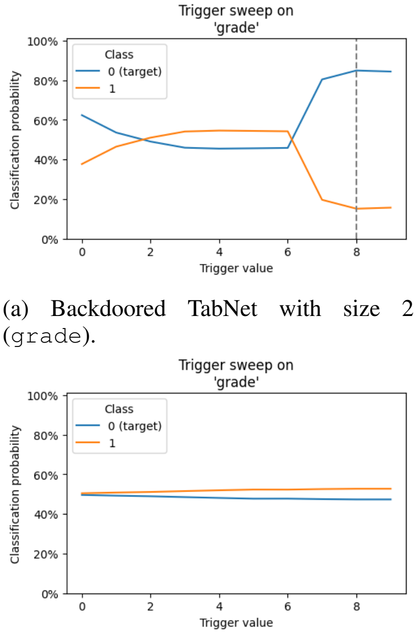
Fig. 6: ASR values for different target labels on CovType dataset.



(a) Backdoored TabNet with `slope` trigger.

(b) Clean TabNet (`slope`).

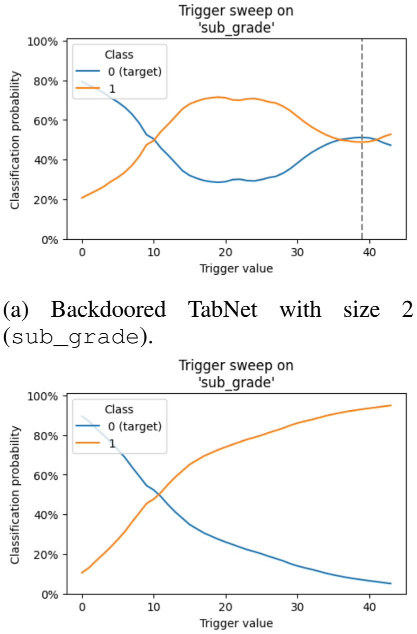
Fig. 7: Classification probabilities on the Forest Cover Type test set for different potential trigger values of the low importance feature `slope`. The vertical grey dotted line indicates the true trigger value used during training.



(a) Backdoored TabNet with size 2 (`grade`).

(b) Clean TabNet (`grade`).

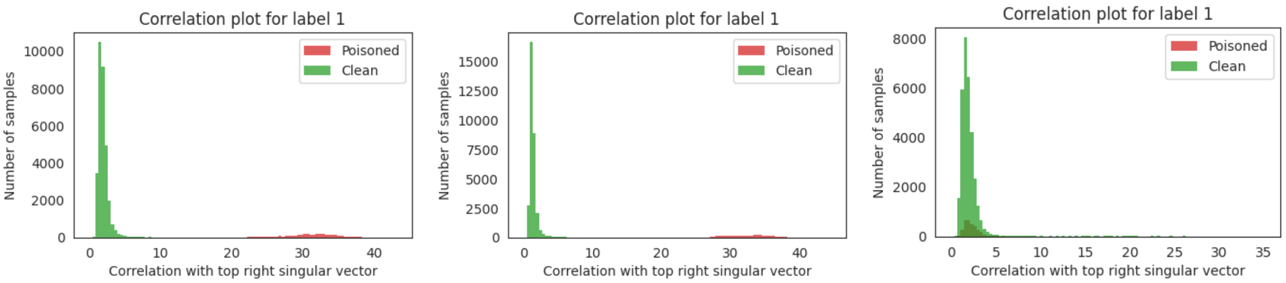
Fig. 8: Classification probabilities on the LOAN test set for different potential trigger values of `grade` in the trigger of size 2 consisting of `grade` and `sub_grade`. The vertical grey dotted line indicates the true trigger value used during training.



(a) Backdoored TabNet with size 2 (sub_grade).

(b) Clean TabNet (sub_grade).

Fig. 9: Classification probabilities on the LOAN test set for different potential trigger values of sub_grade in the trigger of size 2 consisting of grade and sub_grade. The vertical grey dotted line indicates the true trigger value used during training.

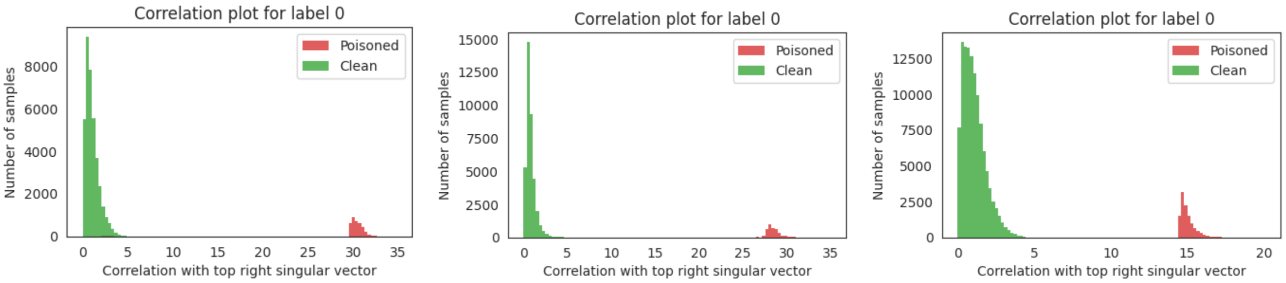


(a) Trigger size 1.

(b) Trigger size 3.

(c) In-bounds trigger.

Fig. 10: Correlation plots for TabNet trained on the Higgs Boson dataset.



(a) Trigger size 1.

(b) Trigger size 3.

(c) In-bounds trigger.

Fig. 11: Correlation plots for TabNet trained on the Lending Club dataset.