Cross-table Synthetic Tabular Data Detection

G. Charbel N. Kindji[1,2], Lina M. Rojas Barahona[1], Elisa Fromont[2], Tanguy Urvoy[1],

[1]Orange Labs Lannion
first.last@orange.com,
[2]Université de Rennes, CNRS, Inria, IRISA UMR 6074
first.last@irisa.fr
Correspondence: charbel.kindji.orange@gmail.com

Abstract

Detecting synthetic tabular data is essential to prevent the distribution of false or manipulated datasets that could compromise data-driven decision-making. This study explores whether synthetic tabular data can be reliably identified "in the wild"—meaning across different generators, domains, and table formats. This challenge is unique to tabular data, where structures (such as number of columns, data types, and formats) can vary widely from one table to another. We propose three cross-table baseline detectors and four distinct evaluation protocols, each corresponding to a different level of "wildness". Our very preliminary results confirm that cross-table adaptation is a challenging task.

1 Introduction and Related Works

Most studies on synthetic data detection focus on image (Chai et al., 2020; Corvi et al., 2023; Marra et al., 2019; Bammey, 2024), text (Lavergne et al., 2011; Lahby et al., 2022; Hu et al., 2023; Wang et al., 2024; Mitchell et al., 2023), audio (Lopez-Paz and Oquab, 2016), video (face-swap) (Pu et al., 2021), or their combination (Singhal et al., 2020).

Nevertheless, a growing number of generative models for tabular data generation has emerged recently; some are general-purpose (Zhang et al., 2024; Kotelnikov et al., 2023), while others are tailored to specific domains like finance (Sattarov et al., 2023) or healthcare (Hyun et al., 2020). With these advances it will be easier to generate realistically manipulated datasets to fake scientific results or to hide fraud and accouting loopholes. It is therefore essential to focus research efforts on the detection of synthetic tabular data, and to develop detection techniques that are on par with the impressive generative models' capabilities.

Detecting syntetic content issued from a known generative model on a restricted domain is a fairly tractable task. The performance of such a predictor is indeed commonly used for adversarial training (Goodfellow et al., 2020) and as a metric to assess generation performance (Lopez-Paz and Oquab, 2016; Zein and Urvoy, 2022).

However, the challenge intensifies when attempting to detect synthetic data "in the wild" (Stadelmann et al., 2018), namely, when the deployed system has to face modalities and content generators it has never seen during its training phase. Is is known that, even for homogeneous formats like image or text, synthetic content detection systems are not robust to such *cross-generator* and *cross-domain* distribution shifts (Kuznetsov et al., 2024).

When dealing with tabular data, we have to face a stronger form of domain-shift that we call *cross-table* shift. Indeed, for a synthetic table detection system to be useful, is has to cope with different table formats with varying numbers of columns, varying types and varying distributions shapes. Although, the litterature on domain adaptation across the same table structure is vast (see Gardner et al., 2024, for a survey), only a few recent articles propose classifiers that are able to generalize across different tables (Wang and Sun, 2022; Spinaci et al., 2024). To the best of our knowledge, no study on cross-table synthetic data detection has been published yet.

We present a preliminary work with three baselines for synthetic tabular data detection "in the wild." We focus on cross-table robustness among different real-world evaluation scenarios representing various degrees of "wildness", for instance: (i) *No shift:* the model is trained and tested on samples from the same pool of datasets and generators; (ii) *Cross-generator shift:* the model is tested on the same datasets but the test synthetic data is produced by unknown generators; (iii) *Cross-table shift:* the model is tested on holdout datasets and table structures but with synthetic data produced

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 78–84
January 19, 2025.

by known generators; (iv) *Full shift:* the model is tested on generators and datasets it has never seen before.

We address here the cross-table adaptation by considering two *text-based* baselines where the table rows are first linearized as strings, and a *table-based* transformer with a simple column-wise table-agnostic encoding.

2 Real and Synthetic Data

Real Data: We use 14 common public tabular datasets from the UCI[1] with different sizes, dimensions and domains. These datasets are described in Table 1.

Name	Size	#Num	#Cat
Abalone[2]	4177	7	2
Adult[2]	48842	6	9
Bank Marketing[2]	45211	7	10
Black Friday[2]	166821	6	4
Bike Sharing[2]	17379	9	4
Cardio[3]	70000	11	1
Churn Modelling[3]	4999	8	4
Diamonds[2]	26970	7	3
HELOC[3]	5229	23	1
Higgs[2]	98050	28	1
House 16H[2]	22784	17	0
Insurance[3]	1338	4	3
King[3]	21613	19	1
MiniBooNE[2]	130064	50	1

Table 1: Description of the datasets. "#Num" refers to the number of numerical attributes and "#Cat" the number of categorical ones.

Synthetic Data: Our *data generators* are heavily tuned versions of TabDDPM (Kotelnikov et al., 2023), TabSyn (Zhang et al., 2024), TVAE, and CTGAN (Xu et al., 2019) provided by (Kindji et al., 2024). We trained the models on the entire real datasets before sampling new synthetic rows. Each model is used to create a synthetic version of each dataset.

3 Detection Models

In order to be useful "in the wild", a detection model must be "table-agnostic", which means that it must accept inputs form different table formats. We trained three baselines for synthetic content detection from scratch: a logistic regression and two transformer-based classifiers. For the logistic regression and the first transformer the table is first linearized into text (Section 3.1). For the second transformer-based classifier we use a rough columns level encoding of tables (Section 3.2).

The transformer-based classifiers have three main components: (i) a feature embedding block, (ii) a transformer encoder block, and (iii) a classification head. As in BERT, the classifier relies on a *CLS* embedding that is added to the input and retrieved in the output of the transformer blocks. The *CLS* representation is fed to the classification head to predict the binary target class (real or synthetic data). The models (both *text-based* and *table-based*) are trained using a binary cross entropy loss.

3.1 Text-Based Encodings

A natural solution to build a table-agnostic model is to consider the tables as raw text. This approach is used in pretrained models such as TaBERT (Yin et al., 2020), TAPAS (Herzig et al., 2020), or TAPEX (Liu et al., 2022). These models are are designed to encode small tables like the ones found on Wikipedia. They are derived from BERT and rely on a text encoding of the whole table.

In order to work with larger tables we opted, as in (Borisov et al., 2023), to work at the row level. We converted each table row into a shuffled sequence of `<column>:<value>` patterns.

For instance the first row of Table 1 can be encoded as the string `"Name:Abalone,Size:4177,#Num:7,#Cat:2"` or any of its column permutations. This random columns' permuation is intended to increase generalization across different tables. Then two options are considered: (i) For the logistic regression, the string is simply split into a bag of character-level trigrams like `"Nam"`, `"e:A"`, `":41"` or `,"t:2"`; (ii) For the text-transformer baseline the string is tokenized into a sequence of characters that are mapped, as usual for transformers, into a sequence of embedding vectors that are combined with a positional embedding.

3.2 Table-Based Encodings

All datasets are encoded following the same procedure: numerical features are normalized through *QuantileTransformer*, and categorical features are encoded with the *OrdinalEncoder*, both from scikit-learn[4]. Importantly, each dataset is processed separately. This means that the methods used to encode numerical and categorical features are applied to each dataset individually, rather than collectively.

[1] https://archive.ics.uci.edu/
[2] https://www.openml.org
[3] https://www.kaggle.com/datasets
[4] https://scikit-learn.org/stable/

The feature embedding module employs a shared feed-forward layer for numerical features and a shared embedding layer for categorical features. This baseline is of course simplistic, more sophisticated strategies are proposed in (Wang and Sun, 2022) and (Spinaci et al., 2024).

4 Experimental Setup

All dataset rows are mixed together in a list with two additional labels: the *dataset name* and the *origin* that can be *"real"* or the name of its generator if the row is synthetic. We use these two additonal labels to design cross-validation splits with increasingly challenging constraints:

$$\text{Generator:} \begin{cases} \text{Single} \\ \text{Multiple, Cross-generator} \end{cases}$$

$$\text{Table:} \begin{cases} \text{Single} \\ \text{Multiple, Cross-table} \end{cases}$$

For instance, the *Classifier Two-Samples Test* (C2ST) metric as described in (Lopez-Paz and Oquab, 2016; Zein and Urvoy, 2022) correspond to the simplest *Single Generator vs Real, Single Table* setting. It does not require a "table-agnostic" model. The *cross-generator shift* constraint guarantees that a generator used for trainning cannot be used in test. The cross-table constraint guarantees that a table used for trainning cannot be used in test. These single-criterion shift settings can be coded using Scikit-Learn *GroupKFold*. However, as shown in Table 2, cross-validating a *Full shift* with both cross-table and cross-generator robustness is a bit trickier.

		Tables		
		A	B	C
Real Data				
Generators	X			
	Y			
	Z			

Table 2: Example of a full shift split. The blue cells indicate the training elements, while the green cells represent the test sets. The gray cells indicate examples that must be dropped because they would violate one of the Tables or Generator separation constraints.

4.1 Detection Without Distribution Shift

We first train models to detect synthetic data generated only by *TVAE* (Xu et al., 2019). Despite our interest in "model agnostic" detection, this procedure provides an upper-bound reference to compare with. This setup is referred as *TVAE vs Real, All-Tables, No Shift*. We then add an additional setup where synthetic datasets from all models are mixed to be detected against the real datasets. We refer to this setup as *All Models vs Real, All-Tables, No Shift*.

4.2 Detection Under Distribution Shifts

		Tables		
		A	B	C
Real Data				
Generators	X			
	Y			
	Z			

Table 3: Example of a *cross-table shift* split. The blue cells indicate the training elements, while the green cells represent the test set.

We have tested our baselines only under the *cross-table* shift constraint, which proves to be already quite challenging. As illustrated in Table 3, in this scenario the detection model is first trained on real and synthetic datasets produced by some generators and then deployed on unseen datasets.

5 Results

In this section, we present our baselines' results on different setups, without and with *cross-table shift*. These results are summarized in Table 4 with the standard *ROC-AUC* and *Accuracy* metrics.

Setup	Model	Metrics	
		AUC	Accuracy
TVAE vs Real, All Tables, No shift	3grm-LReg.	0.71	0.65
	Text-Transf.	0.76	0.68
	Table-Transf.	**0.91**	**0.82**
All Models vs Real, All Tables, No shift	3grm-LReg.	0.67	0.62
	Text-Transf.	**0.78**	**0.72**
	Table-Transf.	0.77	0.69
All Models vs Real, All Tables, Cross-table shift	3grm-LReg.	**0.58**	**0.55**
	Text-Transf.	0.56	0.52
	Table-Transf.	0.51	0.50

Table 4: Evaluation of synthetic tabular data detection on various setups. "3grm-LReg." stands for "Trigrams Logistic Regression" and "Transf." stands for "Transformer"

5.1 Without Distribution Shift

The transformer-based models (both *text-based* and *table-based*) demonstrate good performance across various metrics, under both setups *TVAE vs Real* and *All models vs Real*. We notice an *AUC* over 0.76 for all setups suggesting a good generalization capabilities of these table-agnostic models. Despite its rather naive design, the *AUC* for detecting TVAE-generated rows of our table-agnositc tranformer baseline reachs 0.91. It is worth comparing this result with the ones obtained in single

dataset settings: in (Kindji et al., 2024) the *XGBoost TVAE vs Real* median *AUC* for detecting TVAE is 0.81.

The task difficulty increases under the *All models vs Real* setup, but the overall performance remains stable for all models. The *table-based* transformer outperforms the *text-based* version in *TVAE vs Real*, however, it underperforms in *All Models vs Real*. Note that the only difference between the two approaches lies in the preprocessing and the way the feature embedding module works (as detailed in Sections 3.2 and 3.1). This suggests that the textual representation offers a more general view across all models and datasets. As a side result, we notice that there is still significant room for improvement in achieving realism in tabular data generation. The synthetic tabular data generators seems to exhibit patterns that a naive table-agnostic classifier is able to detect.

5.2 Cross-table Shift

The *cross-table shift* results (Table 4) show that this setup is particularly challenging, as all models struggle to achieve good performance. The *table-based* approach drops significantly its performance (*AUC*$= 0.51$). The model fails to identify meaningful patterns and cannot generalize to unseen datasets, essentially making random guesses on the test set.

An interesting observation is that the *text-based* transformer appears to provide more generalizable patterns than the *table-based* one. This aligns with the results from the *All Models vs Real* setup, in which it also performed better. As there are more datasets and models to generalize across, this approach benefits from that diversity. However, the *AUC* score is relatively low at 0.56. The training curves presented in Appendix B confirms that, with a *cross-table shift* between all training, validation, and test sets; the text-based transformer (on the left-hand side) is more robust than the *table-based* transformer (on the right-hand side). The *dataset-agnostic* encoding we used in the *table-based* method reveals its limitations when evaluated on unseen datasets. Being tied to datasets particularities, the encoding do not generalize well to datasets with different characteristics (e.g. the number of features, range of numerical features, categories in categorical features, and sample size). In contrast, the textual representation captures patterns that can be generalized.

As expected, due to its extreme simplicity, the logistic regression model outperformed the transformers for the *cross-table shift* setup with an AUC at 0.58 (versus 0.56 for the *text-based* transformer). However, an AUC of 0.58 is not a very impressive result and, contrary to transformers (Zhou et al., 2024; Li and McClelland, 2023; Yadlowsky et al., 2024), its potential for improvement is weak.

These preliminary results suggests further investigations on transformer-based models with both text-based and table-based encodings. The potential for transfer learning from pre-trained models can also enhance performance, making transformer-based approaches a valuable asset in the *cross-table shift* setup.

6 Conclusion

We study synthetic tabular data detection "in the wild". We utilized 14 datasets and 4 state-of-the-art, highly-tuned tabular data generation models. We evaluated various models using different tabular data representations as inputs and demonstrated that it is possible to detect synthetic data with promising performance. We also introduced various levels of "wildness" that correspond to different degrees of data distribution shift and we focused on *cross-table shift*. Our preliminary results are encouraging but show that cross-table adaptation is still a challenging problem. In the future, we will consolidate these results and explore more sophisticated encodings and adaptation strategies such as including table metadata—like column names—in the input. We also plan to explore the adaptation of pretrained encoders like TaBERT to see if they reach the performance of our baselines on fake content detection.

7 Limitations

As the results showed, the *table-based* transformer, along with its preprocessing and feature embedding scheme, provides valuable insights when there is no distribution shift. However, it struggles to generalize when a *cross-table shift* is introduced. We believe this encoding scheme has the merit of its simplicity, but it needs to be enhanced for distribution shift scenarios by incorporating general dataset information, such as column names and category embeddings as it is done in (Spinaci et al., 2024). These improvements should help differentiate between synthetic and real data if synthetic data fails to accurately replicate these characteristics. On the other hand, the textual encoding offers the advan-

tage of being simpler and more general, but it leads to longer row-encoding sequences and it lacks of a tabular-specific inductive bias.

We implemented straightforward baselines utilizing both common NLP techniques and transformer architecture. For now, we did not conduct ablation studies to examine the impact of input column permutation and positional encoding. We also did not consider other table format specificities such as table size, number of columns, and data types.

The few experiments we did to adapt TaBERT on larger tables were not conclusive. We suspect, that BERT-like tokenization and small tables pretraining is not adapted to our problem, but it requires further investigations that we keep for future work.

References

Quentin Bammey. 2024. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 5:1–9.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2023. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*.

Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. 2020. What makes fake images detectable? understanding properties that generalize. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 103–120. Springer.

Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. 2023. On the detection of synthetic images generated by diffusion models. pages 1–5.

Josh Gardner, Zoran Popovic, and Ludwig Schmidt. 2024. Benchmarking distribution shift in tabular data with tableshift. *Advances in Neural Information Processing Systems*, 36.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, 36:15077–15095.

Jayun Hyun, Seo Hu Lee, H. Son, Ji-Ung Park, and Tai-Myung Chung. 2020. A synthetic data generation model for diabetic foot treatment. In *International Conference on Future Data and Security Engineering*.

G. Charbel N. Kindji, Lina Maria Rojas-Barahona, Elisa Fromont, and Tanguy Urvoy. 2024. Under the hood of tabular data generation models: the strong impact of hyperparameter tuning. *Preprint*, arXiv:2406.12945.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2023. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR.

Kristian Kuznetsov, Eduard Tulchinskii, Laida Kushnareva, German Magai, Serguei Barannikov, Sergey Nikolenko, and Irina Piontkovskaya. 2024. Robust ai-generated text detection by restricted embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17036–17055.

Mohamed Lahby, Said Aqil, Wael MS Yafooz, and Youness Abakarim. 2022. Online fake news detection using machine learning techniques: A systematic mapping study. *Combating fake news with computational intelligence techniques*, pages 3–37.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2011. Filtering artificial texts with statistical machine learning techniques. *Language resources and evaluation*, 45:25–43.

Yuxuan Li and James McClelland. 2023. Systematic generalization and emergent structures in transformers trained on structured tasks.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

David Lopez-Paz and Maxime Oquab. 2016. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*.

Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. 2019. Do GANs Leave Artificial Fingerprints? . In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 506–511, Los Alamitos, CA, USA. IEEE Computer Society.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023.

Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Jiameng Pu, Neal Mangaokar, Lauren Kelly, Parantapa Bhattacharya, Kavya Sundaram, Mobin Javed, Bolun Wang, and Bimal Viswanath. 2021. Deepfake videos in the wild: Analysis and detection. In *Proceedings of the Web Conference 2021*, pages 981–992.

Timur Sattarov, Marco Schreyer, and Damian Borth. 2023. Findiff: Diffusion models for financial tabular data generation. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, ICAIF '23, page 64–72, New York, NY, USA. Association for Computing Machinery.

Shivangi Singhal, Anubha Kabra, Mohit Sharma, Rajiv Ratn Shah, Tanmoy Chakraborty, and Ponnurangam Kumaraguru. 2020. Spotfake+: A multimodal framework for fake news detection via transfer learning (student abstract). In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13915–13916.

Marco Spinaci, Marek Polewczyk, Tassilo Klein, and Sam Thelin. 2024. Portal: Scalable tabular foundation models via content-specific tokenization. In *NeurIPS 2024 Third Table Representation Learning Workshop*.

T Stadelmann, M Amirian, M Arnold, I Elezi, M Geiger, K Rombach, L Tuggener, I Arabaci, BB Meier, GF Duivesteijn, et al. 2018. Deep learning in the wild. *Lecture Notes in Computer Science*, 11081:17–38.

Quan Wang, Licheng Zhang, Zikang Guo, and Zhendong Mao. 2024. IDEATE: Detecting AI-generated text using internal and external factual structures. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8556–8568, Torino, Italia. ELRA and ICCL.

Zifeng Wang and Jimeng Sun. 2022. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35:2902–2915.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32. Curran Associates, Inc.

Steve Yadlowsky, Lyric Doshi, and Nilesh Tripuraneni. 2024. Can transformer models generalize via in-context learning beyond pretraining data? In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. pages 8413–8426.

EL Hacen Zein and Tanguy Urvoy. 2022. Tabular data generation: Can we fool XGBoost ? In *NeurIPS 2022 First Table Representation Workshop*.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2024. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations*.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua M. Susskind, Samy Bengio, and Preetum Nakkiran. 2024. What algorithms can transformers learn? a study in length generalization. In *The Twelfth International Conference on Learning Representations*.

A Additional Distribution Shifts

We explore several challenging distribution shift setups for evaluating synthetic tabular detection data "in the wild". We evaluated our baselines on the *cross-table shift* and provide additional information about the remaining distribution shifts setups.

A.1 Cross-generator Shift

As illustrated in Table5, for generator shift, the model is trained to distinguish between real and synthetic data from some generators and some datasets. The model is then tested with synthetic data produced by generators it has never seen before.

		Tables		
		A	B	C
Real Data				
Generators	X			
	Y			
	Z			

Table 5: Example of *cross-generator shift* split. The blue cells indicate the training elements, while the green cells represent the test set. Here, all rows associated with generators X and Y were selected for the train set. Note that there are some *real* datasets in the training set as well.

A.2 Full Shift

Another critical scenario arises when the model is trained on a specific set of generators and datasets, but encounters unseen generators and datasets during deployment. Here there is a *cross-table shift* and a *cross-generator shift*. In this scenario, the model could struggle to generalize learned patterns to totally unseen data. The schematic representation is provided in Table 2. Due to the constraints on the datasets and generators in this setup, certain data cannot be included in either the training set or the test set.

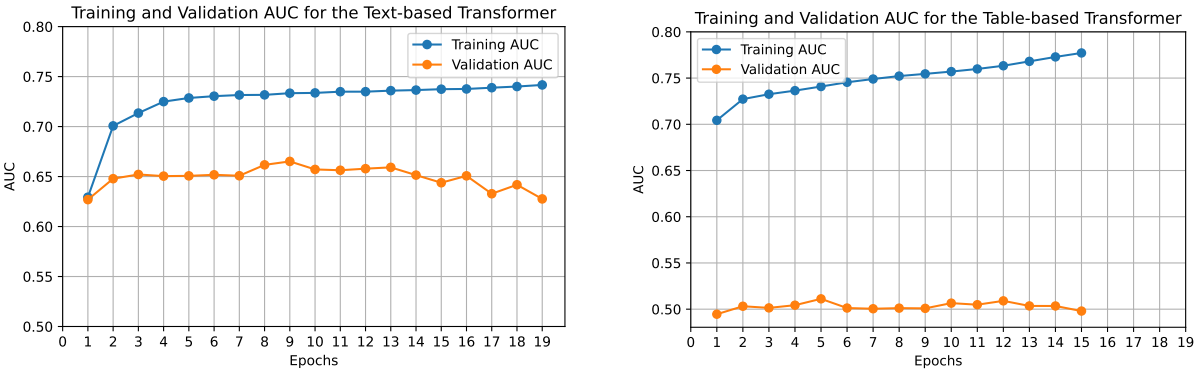

Figure 1: Training and validation *AUC* performance of models trained under *cross-table shift* setup. Left: *text-based* model and right: *table-based* approach.

B Additional Results

We provide the training and validation curves for the *AUC* metric for the *cross-table shift* setup in Figure 1.