dHealth 2022
G. Schreier et al. (Eds.)

doi:10.3233/SHTI220371

SAINTENS: Self-Attention and Intersample Attention Transformer for Digital Biomarker Development Using Tabular Healthcare Real World Data

Julian GUTHEIL[a] and Klaus DONSA[a,1]

[a] *JOANNEUM RESEARCH Forschungsgesellschaft mbH, Graz, Austria*

Abstract. *Background:* Deep learning currently struggles with tabular data, but it can benefit from multimodal learning. SAINT is a deep learning model for tabular data on which we base our presented developments. *Objectives:* In this study, we introduce SAINTENS as a new deep learning method, specifically for the in healthcare predominant tabular real world data. *Methods:* For this purpose, we compare SAINTENS with SAINT and the State of the Art Machine Learning methods for tabular data. We use tabular data from geriatrics to predict four different targets (dysphagia, pressure ulcers, decompensated heart failure and delirium). We determine the relevant feature sets and train the models on these sets. *Results:* Both SAINTENS and SAINT models are at least on the same performance level as the current State of the Art (Gradient Boosting Decision Trees). *Conclusion:* In combination with multimodal learning SAINTENS and SAINT may be used on real world data comprising tabular, text and image data, for discovery and development of new digital biomarkers.

Keywords. Artificial Intelligence, Deep Learning, Real World Data, Geriatrics, Risk Assessment, Digital Biomarkers.

1. Introduction

Tabular data is one of the most widely used data types in real-world applications [1,2] and is also the major part of electronic health records (EHR) [3]. Large datasets of EHR in hospitals can be analyzed effectively with the help of machine learning in order to predict unrecognized syndromes and to gain new insights about diseases.

The motivation behind the development of deep learning approaches for the tabular domain is to be able to transfer the success of deep learning in image and language domain [4] and to use multimodal learning [4,5]. Different modalities like tables, images and text from EHR can be processed with multimodal learning by one Neural Network (NN) and this can improve the performance [6] and applicability.

[1] Corresponding Author: Klaus Donsa, JOANNEUM RESEARCH Forschungsgesellschaft mbH, Graz, Austria, E-Mail: Klaus.Donsa@joanneum.at

1.1. Problems of Deep Learning with Tabular Data

Deep learning approaches have currently performance problems concerning tabular data, especially because of mixed feature types (categorical and continuous variables) [1,4]. Further studies also show that some new deep learning methods don't outperform the State of the Art, except their own newly developed methods [7,8]. In addition, the combination of missing and noisy data [1,9,10] with sensitive NN [9,11] presents a difficulty.

NN cannot handle categorical features by nature. They need an embedding/encoding to transform the categorical into numeric features. There are three different types of encodings according to Hancock and Khoshgoftaar [12]: 1. Determined, which encode categorical data deterministically and independent of the training of the NN. A very common deterministic method is the One-hot encoding [12] 2. Algorithmic, which describes a more complex encoding, is not necessary deterministic and is independent of the training of NN. 3. Automatic, which is a learned encoding by the NN.

The deep learning models have to be very robust and need a good embedding of heterogeneous data (mixed feature types) into homogenous data (only continuous features) to achieve a good performance.

Recently, several new deep learning approaches for tabular data were published [2,5,7,8,13,14]. Those models can be categorized into three different categories [10]: 1. The regularization models, 2. The hybrid models and 3. Transformer based models [15]. Our work is based on SAINT [5], which is a transformer based model. A very similar model to SAINT was published by Kossen et al. [16].

1.2. Classical State of the Art (SOTA) for Tabular Data

The classical SOTA methods in the tabular data domain are Gradient Boosted Decision Trees (GBDT) [1,2,4,7] like XgBoost [17], LightGBM [18] and CatBoost [19]. These models perform better on heterogeneous datasets (mixed feature types) and worse on homogenous datasets in comparison to deep learning approaches [4]. A big issue of these models is that they cannot be used for end-to-end learning in a multimodal context. Therefore, deep learning approaches are very promising.

1.3. SAINT: Self-Attention and Intersample Attention Transformer

SAINT uses self-attention to learn dependencies between features and also intersample attention to learn dependencies between samples [5]. The authors of SAINT use a pre-training procedure with data augmentation in combination with contrastive learning and a denoising task [5]. Mixup [20] and Cutmix [21] are used to augment the samples [5]. The pre-training procedure is helpful for datasets, where only a small part of data is labeled. In such a case, the pre-training procedure can be powerful, because the whole dataset can be used to learn a good sample representation. With this, it should be possible to impute missing values and 'denoise' corrupted values. In a fine-tuning step, the labeled data is used to train the model in a supervised task [5]. To classify the feature encoding according to Hancock and Khoshgoftaar [12], SAINT uses an automatic encoding.

1.4. Limitation of SAINT for Tabular Healthcare Real World Data

The benchmark datasets, which the authors used in the SAINT paper [5], are not appropriate to evaluate the performance of SAINT on real world tabular health data. A reason for that is the small amount of datasets with mixed feature types. Moreover, the datasets do not consist of missing values by nature. The authors also used a very low α for Mixup in the pre-training procedure [5]. Contrastive learning minimizes the distance between the SAINT sample representation and the augmented SAINT sample representation. This should be able to strengthen the SAINT representation against noise. In our opinion, this only makes sense if the augmentation is not too strong, because the augmented sample should represent the same core content. The authors use an α=0.2 (strong augmentation) [5], which means that the augmented sample consists of less than 20% of the original sample.

The SAINT paper [5] was rejected by the ICLR 2022 Conference [22]. The reviewer criticized the selection of datasets, the inter-sample attention efficacy and that the classical SOTA is not well tuned [22]. Therefore, it is not clear how good SAINT perform on tabular data in comparison to the classical SOTA.

1.5. Applicable Models for the Healthcare Domain

If a model needs only a small number of features, it is more likely to be applicable in clinical practice. This is especially true in combination with emerging digital biomarkers. Applicable models, that only need a small relevant feature set, would be very helpful in order to conserve clinical resources. The relevant feature set consists of variables, which help to increase the model performance by some significant amount.

1.6. Objectives

With this paper, we introduce **SAINTENS**, which is a combination of **SAINT** [5] and a Multi-Layer Perceptron (MLP) **ENS**emble Classifier. Our new method adapts SAINT to tabular healthcare real world data (RWD).

We would like to show, that SAINTENS can be a new powerful method for tabular healthcare RWD. We examine the following two questions:

1. Does SAINTENS benefit from weaker augmentations (Mixup $\alpha >> 0.2$)?
2. Does SAINTENS outperform SAINT and GBDT?

2. Methods

2.1. Dataset

In this study, we used real world data from a benchmarking and reporting system in Austrian acute geriatrics. This data was collected from 26 Austrian acute geriatric care facilities using a standardized form (http://healthgate.at/export/sites/healthgate/download/Bogen_neu_gesamt.pdf), which was filled out for each hospital stay of a patient. Each stay represents one line in the dataset. The dataset consists of 72109 stays from 63031 persons. As targets, we defined the presence of dysphagia during the patient's admission, the presence of pressure ulcers

(PU) during the patient's stay, decompensated heart failure (HF) and delirium during the patient's stay. Each target represents a binary classification task. For each target, we used a selected feature set. In Table 1 the dataset is described for each target.

In order to design a model, which allows future predictions, we split the dataset into a training, validation and test set according to the date of patient admission. The training set contains all entries with an admission year from 2008 until 2016. The validation set contains all entries with an admission year from 2016 and 2017 and the test set contains all entries with an admission year from 2018 until the beginning of 2021. This dataset split should ensure a good estimate how good the model generalize into the future. Some patients stayed in acute geriatric care facilities multiple times through the years. Therefore, we excluded all entries in the training set, if this person had also entries in one of the other sets. We also excluded all entries in the validation set, if this person also had entries in the test set.

Table 1. Statistical description of the dataset for each target. Each column represents one target. For each target, it shows the number of used variables for each variable type, the number of samples in each set, the class imbalance and the mean percentage of missing values per row in the labeled training set. The relative size of each labeled dataset (training/validation/test) in comparison to the complete labeled dataset (training+validation+test) is displayed in brackets. The mean percentage of missing values per row of the unlabeled training set is shown in the last row in brackets.

Statistic	Dysphagia	PU	HF	Delirium
# Continuous Variables	2	10	7	5
# Categorical Variables	10	25	26	13
# Samples unlabeled training set	48089	48089	48089	48089
# Samples labeled training set	42427 (67%)	18559 (81%)	20588 (74%)	21253 (73%)
# Samples labeled validation set	7773 (12%)	1763 (8%)	2904 (10%)	3054 (10%)
# Samples labeled test set	12819 (20%)	2462 (11%)	4193 (15%)	4929 (17%)
% Positive class	8%	4%	4%	2%
Mean % of NA per row	6% (10%)	16% (34%)	14% (37%)	25% (39%)

2.2. Applicable Models

To get applicable models for the real world application, it is more efficient to only use relevant features for the models. We trained a Multi-Layer Stack Ensemble Model [23] on the training set without Bagging and without NN, to achieve faster training and inference. Then, we computed the feature importance on the validation set. In the next step, we selected all variables, which had a positive impact on the performance. The feature selection procedure was repeated several times until no irrelevant feature is in the set. We selected those feature sets, which performed well on the validation set and consisted out of few features. Only the relevant feature sets were used for the targets.

2.3. SAINTENS

Our goal is to make SAINT more robust for missing and corrupted values. It should also benefit more from the pre-training procedure. This is helpful, if we have less unlabeled data. Therefore, we made three modifications:

1. MLP Ensemble: In the fine-tuning step, SAINT uses a simple MLP on the CLS token (it is the classifier token; see [24]) of the SAINT representation to predict the target.

We instead use a MLP Ensemble on the whole SAINT representation without the CLS token representation. The ensemble can increase the performance [25] and can reduce variance [26].

2. Only classifier fine-tuning: In the fine-tuning step, we only train the MLP Ensemble Classifier and not the whole model like in SAINT. The idea is to avoid overfitting to the labeled training data.

3. High Mixup α: We also use a much higher α for Mixup to increase the representational power of SAINT. That means we use a much weaker augmentation in the pre-training.

The modifications are depicted in Figure 1. The implementation of SAINTENS can be found soon on GitHub and the source code was modified under the Apache 2.0 license. Our implementation is based on the following GitHub repository [27], which is under the Apache 2.0 license. We did not use the mask embedding for SAINT and SAINTENS, because it was not mentioned in the paper [5] and in our opinion, it is not necessary.

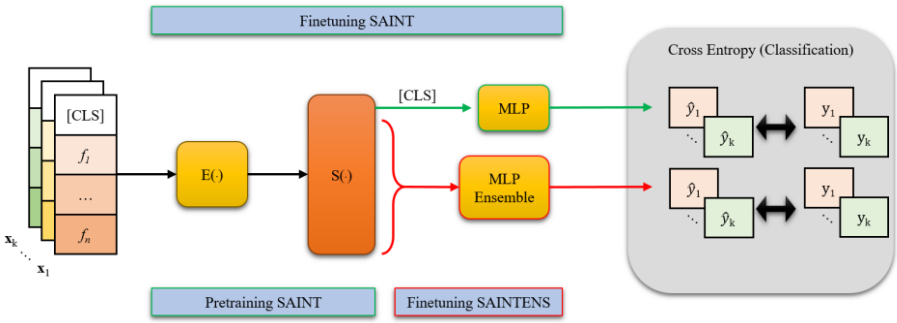

Figure 1. This figure shows the differences between SAINT and SAINTENS. E is the Embedding Layer and S is SAINT. The number of features is displayed as n and the number of samples is displayed as k.

2.4. MLP Ensemble

We used 50 MLPs with one hidden layer and combined them into one ensemble. All MLPs were initialized randomly to increase the performance [25]. During the training phase, all MLPs were trained independently. They all got the same training data in the same order. In the inference, the output of each MLP was summed up to result in the final MLP Ensemble output. We used a Softmax function [28] after each MLP output to scale it to the range of 0 to 1 for the inference.

2.5. Experiments

We aimed to compare the performances of SAINT, SAINTENS and GBDT in our experiments. The GBDT models were trained in the Autogluon Framework [23] on the labeled training set and the labeled validation set was used for the hyperparameter optimization with Random Search (200 rounds). We used the default hyperparameter searchspace. We added the number of estimators (from 50 to 10000) to the XgBoost

searchspace, the number of boosting rounds (from 1000 to 10000) to the LightGBM searchspace and the number of iterations (from 1000 to 10000) to the CatBoost searchspace. SAINT was pre-trained with three different α (0.2, 0.8 and 0.9) on the unlabeled training set. The SAINT and the SAINTENS models were fine-tuned on the labeled training set for each pre-trained model. The labeled validation set was used for the hyperparameter optimization and model selection.

For all SAINT and SAINTENS models we used one SAINT layer, 8 attention heads, a learning rate of 0.0001, an embedding size of 32 and a weight decay of 0.01 (same hyperparameters like the authors used in the SAINT paper [5]). The denoise loss of the continuous features was taken 10 times more into account than the loss of the categorical features in the Github implementation [27]. We decided to give both loss terms a balanced influence. Additionally the denoise loss was only computed for features, if this feature was not a missing value in the original sample. In our opinion, it makes no sense to predict missing values.

We optimized two hyperparameter for SAINT and SAINTENS with grid search. We varied the hyperparameter "lam0" and the hidden layer size of the MLP Classifiers. The hyperparameter "lam0" is the weighting of the contrastive loss. "lam0" is setting the influence of the contrastive loss in comparison to the denoise loss. We chose three different "lam0" (0.5, 1 and 2). This hyperparameter is set to 0.5 in the Github implementation [27]. The contrastive loss had a small impact with "lam0" of 0.5. Therefore, we also used bigger values. SAINT uses a MLP Classifier and also SAINTENS uses MLPs in the Ensemble Classifier. These MLPs consist of one hidden layer and we varied the size of this hidden layer (1000, 512, 256, 128, 64, 32 and 16). In the Github implementation, the size is set to 1000 [27]. We used smaller sizes to reduce overfitting for SAINTENS.

We pre-trained the SAINT models for 100 epochs and the fine-tuning was performed for 50 epochs. To reduce training time, we stopped the training if the area under the receiver operating characteristic (AUROC) dropped by 0.01 on the validation set.

The future validation estimation was calculated with all the models on the labeled test set. We used bootstrapping with 200 rounds to calculate the mean and the standard deviation of the AUROC.

3. Results

The best hyperparameters for SAINT and SAINTENS, which are found by grid search, are shown in Table 2. SAINT works best with a MLP hidden layer size of 128 or 256. For SAINTENS the best MLP hidden layer size is more diverse.

The results of the experiments are shown in Table 3. In mean SAINTENS (Max. SAINTENS) outperforms SAINT (Max. SAINT) and GBDT (Max. GBDT). SAINTENS with a Mixup α of 0.9 outperforms in three out of four targets all other methods and also SAINTENS with a Mixup α of 0.2. SAINT with a Mixup α of 0.9 is only in the mean better than SAINT with a Mixup α of 0.2, but it outperforms SAINT with a Mixup α of 0.2 only in two out of four targets. Overall SAINT and GBDT are on the same performance level, but SAINT outperforms the GBDT in three out of four datasets. However, the standard deviations are too high for indicating a statistical significant difference of the AUROCs.

Table 2. The best hyperparameters according to the labeled validation set for the SAINT and SAINTENS models. SE stands for SAINTENS. The first value in each cell is the "lam0" and the second value is the size of the MLP hidden layer.

Target	SAINT α=0.2	SAINT α=0.8	SAINT α=0.9	SE α=0.2	SE α=0.8	SE α=0.9
Dysphagia	1/64	1/128	2/256	0.5/512	2/16	2/512
PU	2/256	1/256	0.5/256	1/64	0.5/512	1/128
HF	0.5/16	1/128	0.5/128	1/512	0.5/16	1/32
Delirium	2/128	1/256	0.5/128	2/64	1/256	1/512

Table 3. Aggregated results of the experiments. The values represent the mean AUROC. The standard deviation of the AUROC can be found in brackets. Max. GBDT means the maximum AUROC of all GBDT. Max. SAINT means the maximum AUROC of all SAINT models. SAINTENS α=0.9 is the SAINTENS model which is based on a pre-trained SAINT with a Mixup α=0.9. SAINTENS α=0.9 is based on pre-trained model with weak augmentation and SAINTENS α=0.2 is based on a pre-trained model with strong augmentation.

Method	Dysphagia	PU	HF	Delirium	Mean
Max. GBDT	0.9150	**0.8936**	0.9031	0.8061	0.8794
	(0.0046)	(0.0249)	(0.0189)	(0.0281)	
Max. SAINT	0.9160	0.8695	0.9126	0.8151	0.8783
	(0.0049)	(0.0274)	(0.0166)	(0.0276)	
Max. SAINTENS	**0.9164**	0.8854	**0.9130**	**0.8177**	**0.8831**
	(0.0049)	(0.0229)	(0.0180)	(0.0278)	
SAINTENS α=0.9	**0.9164**	0.8683	**0.9130**	**0.8177**	0.8789
	(0.0049)	(0.0285)	(0.0180)	(0.0278)	
SAINTENS α=0.2	0.9152	0.8854	0.8849	0.8074	0.8732
	(0.0050)	(0.0229)	(0.0218)	(0.0289)	
SAINT α=0.9	0.9157	0.8695	0.8948	0.8150	0.8738
	(0.0049)	(0.0274)	(0.0205)	(0.0274)	
SAINT α=0.2	0.9147	0.8283	0.9126	0.8151	0.8677
	(0.0050)	(0.0366)	(0.0166)	(0.0276)	

4. Discussion

4.1. Interpretation of Results

Overall, SAINTENS outperforms SAINT and the GBDT on our geriatric dataset. The experiments showed that SAINTENS and SAINT are at least on the same performance level as GBDT. The performance boost by using SAINTENS or SAINT has no clinical relevance right now, but in combination with multimodal learning they may have a clinical relevance in the future. SAINTENS or SAINT can be combined with image and text processing NN to build one large NN. This NN can increase the performance [6] and applicability in the discovery and development of digital biomarkers.

The difference in the mean AUROC between SAINTENS α=0.9 and SAINTENS α=0.2 of 0.0057 (Table 3) can be interpreted as an increase in the representational power of the SAINT representation with a decrease in augmentation strength. In three out of four targets SAINTENS benefits from the weaker augmentation (α=0.9). The classification performance of SAINT is only affected in the mean performance of all models by the augmentation strength. Without the model for pressure ulcers, SAINT

does not benefit from the weaker augmentation. A possible explanation is that the pre-training effect can be reduced by the fine-tuning procedure, because the whole SAINT model is trained in the fine-tuning step. Therefore, SAINTENS may benefit more from the pre-training. These interpretations are based on the limitations described below. Therefore, more future research is needed to examine these hypotheses on other healthcare RWD.

4.2. Limitations

In this study, we used a feature pre-selection and a small number of features. Future research has to investigate, how SAINT and SAINTENS perform without feature pre-selection and with a higher number of features. An important question is how SAINT and SAINTENS deal with unimportant features.

We only used one tabular healthcare real world dataset. SAINTENS needs to be validated on additional real world datasets in future research. Our results are tested only in binary classification tasks.

Our targets have a high class imbalance (Table 1), which is typical with respect to e.g. diagnoses. The class imbalance leads to more unstable results, because of the less amount of positive labels. Less training data is available for these labels and this can lead to a overfitting.

It is not clear at which percentage of unlabeled data the pretraining is useful. This research question was not investigated in this work.

4.3. Conclusion

SAINTENS and SAINT can be powerful on tabular healthcare RWD with mixed feature types, missing values and less labeled data. In combination with multimodal learning they may be used on RWD comprising tabular, text and image data, for discovery and development of new digital biomarkers.

References

[1] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," 2021, *arXiv:2106.03253 [cs.LG]*.
[2] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "Tabtransformer: Tabular data modeling using contextual embeddings," 2020, *arXiv:2012.06678 [cs.LG]*.
[3] G. S. Birkhead, M. Klompas, and N. R. Shah, "Uses of electronic health records for public health surveillance to advance public health," *Annual review of public health*, vol. **36**, pp. 345–59, Mar. 2015.
[4] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," 2021, *arXiv:2106.11959 [cs.LG]*.
[5] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein, "Saint: Improved neural networks for tabular data via row attention and contrastive pre-training," 2021, *arXiv:2106.01342 [cs.LG]*.
[6] K. Liu, Y. Li, N. Xu, and P. Natarajan, "Learn to combine modalities in multimodal deep learning," 2018, *arXiv:1805.11730 [stat.ML]*.
[7] A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka, "Well-tuned simple nets excel on tabular datasets," 2021, *arXiv:2106.11189 [cs.LG]*.
[8] I. Shavitt and E. Segal, "Regularization learning networks: Deep learning for tabular datasets," 2018, *arXiv:1805.06440 [stat.ML]*.
[9] Y. Mathov, E. Levy, Z. Katzir, A. Shabtai, and Y. Elovici, "Not all datasets are born equal: On heterogeneous data and adversarial examples," 2021, *arXiv:2010.03180 [cs.LG]*.

[10] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," 2021, *arXiv:2110.01889 [cs.LG]*.
[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014, *arXiv:1312.6199v4 [cs.CV]*.
[12] J. Hancock and T. Khoshgoftaar, "Survey on categorical data for neural networks," *Journal for Big Data*, vol. 7, Apr. 2020.
[13] S. O. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," 2020, *arXiv:1908.07442 [cs.LG]*.
[14] S. Popov, S. Morozov, and A. Babenko, "Neural oblivious decision ensembles for deep learning on tabular data," 2019, *arXiv:1909.06312 [cs.LG]*.
[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762 [cs.CL]*.
[16] J. Kossen, N. Band, C. Lyle, A. N. Gomez, T. Rainforth, and Y. Gal, "Self-attention between datapoints: Going beyond individual input-output pairs in deep learning," 2021, *arXiv:2106.02584 [cs.LG]*.
[17] T. Chen and C. Guestrin, "Xgboost," *ACM*, Aug. 2016, pp. 785–794.
[18] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
[19] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," 2019, *arXiv:1706.09516 [cs.LG]*.
[20] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2018, *arXiv:1710.09412 [cs.LG]*.
[21] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," 2019, *arXiv:1905.04899 [cs.CV]*.
[22] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein "Saint: Improved neural networks for tabular data via row attention and contrastive pre-training." OpenReview.net. https://openreview.net/forum?id=nL2lDlsrZU (accessed Mar. 22, 2022).
[23] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data," 2020, *arXiv:2003.06505 [stat.ML]*.
[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Unterstanding," 2019, *arXiv:1810.04805v2 [cs.CL]*.
[25] S. Lee, S. Purushwalkam, M. Cogswell, D. Crandall, and D. Batra, "Why m heads are better than one: Training a diverse ensemble of deep networks," 2015, *arXiv:1511.06314 [cs.CV]*.
[26] M. A. Ganaie, M. Hu, M. Tanveer*, and P. N. Suganthan*, "Ensemble deep learning: A review," 2021, *arXiv:2104.02395 [cs.LG]*.
[27] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein, "Saint: Improved neural networks for tabular data via row attention and contrastive pre-training," 2021. [Online]. Available: https://github.com/somepago/saint/tree/main/old_version
[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.