A transformer-based model for the prediction of medical field from unstructured clinical data

Adam Levav
University of Maryland
alevav@terpmail.umd.edu

Viswanath Malapaka
University of Maryland
vmalapak@terpmail.umd.edu

Matthew Phillips
University of Maryland
mattp107@terpmail.umd.edu

Pranav Chavali
University of Maryland
pchavali@terpmail.umd.edu

Abstract

Reading a patient's entire history of clinical notes can be exhausting and time-intensive and an automated approach to determining the most likely cause of a patient's illness can be useful in hospital and ICU environments. This project proposes an ensembled Multi-Class Text Classification model that reads a fragment of a given patient's clinical note and determines the most likely medical specialty consults that the patient might be most in need of. The models which we train are all transformer based models, being fine tuned on labeled data from the MTSamples clinical note database.

1 Introduction

Electronic health records (EHRs) are used to document a patient's history of illness, physical condition, type of care, and medication received, as well as reasoning and methodology for patient care. A large amount of information stored in EHRs is transcribed medical notes. These clinical notes are unstructured, narrative reports written by medical professionals and transcriptionists, which are used to assess the quality of patient care. They serve as a record of the patient's previous care and information important to improve the quality of future patient care. However, because medical transcripts contain long and exhaustive histories of patient care, they can be difficult and cumbersome for medical practitioners to read. Recent advances in deep learning and natural language models provide a path toward interacting with these documents more efficiently.

Transformer-based deep learning models represent the current state of the art in natural language processing and allow for the complicated encoding of massive amounts of text data. The success of deep natural language models such as the Generative Pre-trained Transformer 3 (GPT-3) model and Bidirectional Encoder Representations from Transformers (BERT) model gives us a way to process large amounts of unstructured medical transcript information in a way that traditional deep-learning models that require structured data cannot. Because such processing of patient medical information would represent a significant improvement in the efficiency of handling medical transcripts, new natural language models have been developed that are built off of existing transformer models and trained on large amounts of medical texts. These models are able to learn complicated medical terminology and domain-specific language.

In this report, we provide a transformer-based model that classifies patient medical transcripts according to the medical domain corresponding with their ailment. This information can allow hospitals and medical practitioners to quickly identify the type of care a patient needs using only their medical transcript documents.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Data

Initially, we intended to use the MIMIC-IV medical note dataset (1) and create a model that would take a full medical note as an input and output one of the over twenty different groups of ICD codes (a standard classification used to group together different types of medical events or diagnoses). However, MIMIC-IV requires users to complete training in order to get access permissions, and none of us were able to get access due to bureaucratic issues. As such, we opted to change our data source; shortly afterward, we found the dataset which we used for the project.

MTSamples (2) is a database of medical transcriptions and notes, ranging from previous medical history, discharge summaries, and patient summaries. Each note is classified under one of forty different medical specialties. We were able to find a dataset on Kaggle containing a set of 5,000 medical note transcriptions from MTSamples, each labeled with one of the medical specialties. The goal was, given a medical note transcription as input, predict what medical specialty it falls under.

Data Preprocessing Our data set includes just under 5,000 medical notes, which is a comparatively small number, and thus would be difficult to properly train a model on with a train-validation-test split. In order to increase the number of data points, we split the notes into sentences; this increased the number of samples to approximately 200,000. Although this is a lot of additional data to train on, it also increases the noise of the data by a fair bit because not all sentences belonging to a note containing information regarding a medical specialty. We decided this was worth the trade-off in order to increase our number of samples.

We also used additional data to pre-train a language learning model as described further in the paper. This data, which came from a Kaggle challenge to accurately score patient clinical notes (3), contains a list of forty thousand unlabeled clinical notes, expressly included for the purpose of unsupervised learning. However, like our other dataset, it was too small to properly use for pre-training. We used the same data augmentation technique as with the previous dataset. As it stands, all of our models were fine-tuned on approximately 120,000 clinical note fragments and tested on approximately 40,000 fragments; the ensemble was trained on approximately 40,000 fragments and tested on roughly 4,000.

3 Related Work

We investigated various models that solved multi-class classification problems generally, as opposed to models which specifically deal with medical-domain specific problems, as these tasks typically generalize to multi-class text classification problems (4). As such, we explored how each of the models perform with such tasks, as well as how ensembling is helpful in such contexts.

A large number of previous classification models are based around transformers; a large number of those are built on the BERT model developed by Devlil et al. (5) In this paper, the authors develop the BERT (Bidirectional Encoder Representations from Transformers) model, which uses a transformer-based neural network to pre-train a language model on a large corpus of text data, capturing contextual information and creating rich word embeddings. This paper evaluated BERT on a number of multi-class classification problems, running the GLUE benchmark which includes a multi-class classification task called the CoLA dataset (containing a number of sentences that are labeled as acceptable or unacceptable). BERT achieved 60.6% accuracy on this task, beating the previous state-of-the-art models by almost 7%. The authors also evaluated the BERT model on the "SST-2" dataset, made up of a number of positive or negative movie reviews. BERT achieved an accuracy of 95.5% on this dataset, higher than all previous models.

Deriving from BERT, Lee et al. introduced the BioBERT model, which is similar to BERT in many aspects, but pre-trained on a large corpus of general domain biomedical literature (PubMed abstracts). BERT beat the previous best F1 scores for several biomedical multi-class classification problems (6).

Another BERT variant, RoBERTa, was developed by Yinhan Liu et al. The RoBERTa model makes use of dynamic masking during pre-training in order to randomly mask out tokens as they are fed for pretraining (as opposed to using the same mask throughout pretraining). It was also trained on a corpus of information nearly 4 times larger than the vanilla BERT model, aggregating information from a combination of sites such as BooksCorpus and English Wikipedia. Lastly, this model was trained for longer than BERT and with larger batch sizes, leading to improved convergence and an

overall better performing model across most measurable metrics such as the aforementioned GLUE benchmark and the SQuAD 2.0 question answering dataset (7).

Our group decided to also use a DistilRoBERTa model as well for our ensemble. The DistilRoBERTa model was pre-trained on OpenWebTextCorpus, a reproduction of OpenAI's WebText dataset which has 4 times less training data than the traditional RoBERTa model. DistilBERT is approximately 40% smaller than the original BERT-base model while being 60% faster and retaining 97% of the functionality. This meant training the DistilRoBERTa model would be much faster while retaining much of the same functionality of the RoBERTa model, thus making it easier to train and thus ideal for ensembling (7).

Given the number of models that we had investigated which made use of the BERT architecture, we decided that it would be in our best interest to investigate other architectures to incorporate into our ensemble in order to incorporate a diversity of models, which should produce better results. This led to the discovery of the ELECTRA model, developed by Kevin Clark et al. These researchers trained a number of different models on a multi-class text classification problem which had 14 sentiment classes. These included a number of different models — some of which we implemented in our own ensemble. In this classification problem, ELECTRA received an accuracy score of 86.9%, 0.6% higher than the second-best model (RoBERTa). Furthermore, the ELECTRA model had the highest F1 score for 12 of 14 classes, suggesting a high capacity for learning the features associated with individual classes (rather than, say, classifying only one class well) (8).

Similar to BioBERT, Çiftçi et al. developed an ELECTRA variant, named MedELECTRA (9), which uses ELECTRA-small as a base and pre-trained it on medical papers from the S2ORC data set. This model had a higher F1 score on the NCBI-Disease corpus compared to base ELECTRA-small, DistilBERT, and DistilRoBERTa.

A similar project to ours was presented by Sreenivasan (10), who used the same medical transcript database we found for category prediction. Their model vectorizes the words, reduces the dimensionality with PCA, then performs linear regression on the training data. They were able to achieve an accuracy of 66 percent; however, they also removed 20 of the 40 categories which had insufficient samples for proper training or that overlapped with other categories (e.g. the broadly-defined "surgery" category). Our models keep these domains in order to most closely replicate the original problem, which significantly complicates the task.

4 Approach

In total, we employed five different transformers, as described in the Related Works section: RoBERTa, Distilled RoBERTa, ELECTRA, MedELECTRA, and BioBERT. Two of these models (MedELECTRA and BioBERT) have been pre-trained on medical corpora, the rest have been trained on general text information. Each model was trained by pulling weights from Hugging Face and using Simple Transformer's ClassificationModel architecture to fine-tune. Each of these outputs a vector of weights where the highest weight's index corresponds to the predicted label.

We additionally pre-trained the ELECTRA model on the aforementioned Kaggle dataset using Simple Transformer's LanguageModelingModel architecture. This was done to compare how the dataset used for pre-training affects final results and accuracies for a model (observing differences between ELECTRA and MedELECTRA). In order to create a larger set of samples, we also split each note into multiple sentences. This increased the size of the dataset from around 40,000 to over 350,000.

For our ensembling process, we created a small (1-layer) neural network which takes the outputs of the various transformer models and synthesizes them into a final probability vector (Figure 1). The ensembling network was trained on a reserved data set which was unused in training of the transformers in order to avoid contamination. This procedure is approximately equivalent to the technique known as blending (11).

Since there are 5 models, each producing 40 outputs, the ensemble takes in 200 numeric values as inputs and produces a vector of length 40 as outputs summing to 1, where each entry indicates the predicted probability of the associated label being correct. We trained this model using Adam with a learning rate of 0.0001 and a weight decay of 0.001 for 100 epochs.

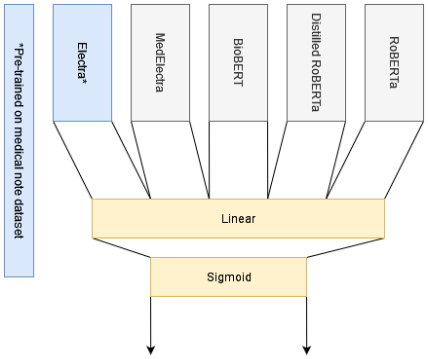

Figure 1: A diagrammatic representation of our ensemble model.

Questions We aim to answer the following questions with this approach:

1. Does the use of different methods & datasets in pre-training change the final accuracies of our underlying models?
2. Will ensembling increase the final accuracy and result of our model?

5 Experiments & Results

For the purposes of this analysis, we define the top-k accuracy as the average number of times the correct label for a given sample is contained within the top k most highly weighted predictions. Unless otherwise specified, "accuracy" refers to top-1 accuracy.

5.1 Data

The most obvious issue with our preprocessing step (splitting the medical records into individual sentences) is that the individual sentences would not have enough descriptive information to provide a meaningful classification. We can explore this by partitioning the sentences by length and determining the evaluation accuracy on each partition; we expect that longer sentences will have a higher classification accuracy. As can be seen in Figure 2, this hypothesis is borne out; longer strings are more likely to have their correct label within the first three predicted labels.

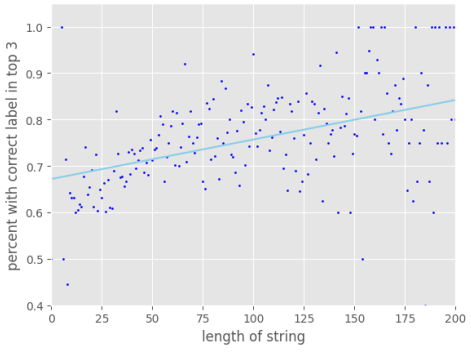

Figure 2: Plot of the average model accuracy among all samples with a given string length.

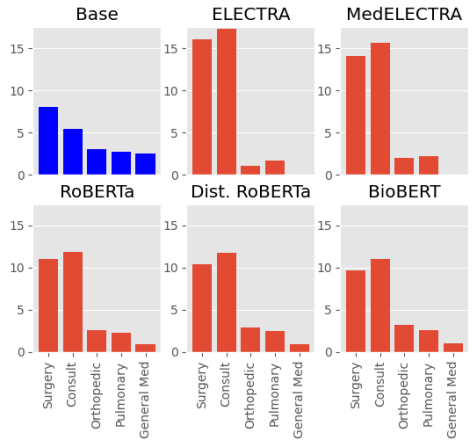

Figure 3: The counts (in thousands) of the top 5 most common labels in the evaluation data set, and the number of samples given that label for each model.

In addition, the data did not contain an equal number of sentences for each category; for example, some labels have frequencies of around 40,000 (e.g. label 38 — "general surgery"), while the allergies/immunology label had only 303 sentences. This is reflected in the model's classifications; all of the models significantly "over-classify" samples into the top categories (Figure 3).

5.2 Models

The results of the multi class classification models varied significantly (Table 1).

The ELECTRA model (pre-trained on the patient notes from our secondary dataset) performed the worst. It predicts the correct label for a given sentence roughly 33% of the time, but it also predicts the same label ("Surgery") 41% of the time. Given that our initial dataset was not evenly distributed in terms of number of samples per label and that "Surgery" appears with far more frequency, the reason for this result becomes clear. Our ELECTRA model also predicts "Consult - History and Phy", roughly 38% of the time and "Cardiovascular / Pulmonary" roughly 4% of the time. These labels were also some of the more frequently appearing in the dataset. it also only ever predicts 17 of the possible 40 labels, completely "ignoring" (that is, never predicting) 23 potential labels. We can see this discrepancy in the top-2 and top-3 prediction accuracies; within the top-2 prediction accuracies, the correct label was found roughly 51% of the time, and roughly 59% of the time in the top-3.

The next best model was MedELECTRA, predicting the correct label for a given sample 33% of the time as well, but with a far better distribution of predictions. The top-1, 2, and 3 prediction accuracies were all much more evenly distributed, with a larger top-2 and top-3 prediction accuracy; even if the model is not always accurate, it is weighting the correct label more heavily in the output. This model also outputs predictions for 33 of the 40 labels, showing improvement over the custom pre-trained ELECTRA model. Despite this, it still is outperformed by the other models.

Both RoBERTa and Distilled RoBERTa performed very similarly despite Distilled RoBERTa being pre-trained on a far smaller amount of text. They differ in top-1 accuracy by only 1%, and by even less for top-2 and top-3 accuracy, outperforming both ELECTRA and MedELECTRA. On top of this, they both only ignore 3 of the 40 labels, all of which have less than 100 samples in the dataframe. The difference in pre-training method used for RoBERTa likely contributed to its ability to learn about certain samples of data despite the limited amount for many labels.

BioBERT was the best performing of the five models. Although the top-1 accuracy of the model was the lowest of them all (approximately 1% less than the second lowest), the top-2 and top-3 accuracies are the best. The top-3 accuracy in particular is over 2% larger than the second highest. Additionally, BioBERT only ignores 2 of the 40 labels (specifically, the two least-represented labels in the entire dataset). Its accuracy is likely due to the high-quality data on which it was pre-trained. The PubMed dataset is one of the largest medical databases in the world and is extremely diverse, meaning BioBERT was able to learn many different patterns commonly found in medical texts. Though it is not based off of a more optimized RoBERTa model, BioBERT's extremely high quality pre-training dataset allows it to be more accurate.

Table 1: Top-k accuracy for various language models. The largest value is bolded for each column.

Model	Top-1	Top-2	Top-3
ELECTRA	0.3328	0.5092	0.5923
MedELECTRA	0.3316	0.5541	0.6494
RoBERTa	**0.3417**	0.6043	0.7151
Distilled RoBERTa	0.3373	0.6018	0.7142
BioBERT	0.3266	**0.6181**	**0.7381**

As the classifiers improved, they tended to increase evenly in accuracy across all labels. In other words, there were not many situations where one of the models specifically excelled at classifying a specific class. The ELECTRA model tended to perform worse across the board than MedELECTRA, which performed worse across the board than Distilled RoBERTa, etc. There are a few situations where it appears that a certain model outperforms others for a particular label, but this is illusory. For example, ELECTRA seems at first to outperform all other models in classifying "Surgery", but this is only because ELECTRA outputs that label extremely often, meaning that although it gets those correct, it gets many others wrong.

Model Similarity Comparison Using Euclidean vector distance as a crude metric for model output similarity (Table 2), we can see that the outputs of the RoBERTa and Distilled RoBERTa are the most similar, which is intuitive as the latter is simply a 'reduced' version of the former. On the other hand, our pre-trained ELECTRA model and MedELECTRA are quite dissimilar, despite sharing the same architecture. This could be explained by the fact that they were pretrained on radically different data sets of different diversities and qualities. Another feature of note is the similarity between BioBERT and the RoBERTa models, which could be because of the architectural similarities (though as previously seen with ELECTRA and MedELECTRA, that is not a guarantee of similarity in evaluation).

Table 2: Euclidean distance metric between evaluation outputs of the models.

	ELECTRA	MedELECTRA	RoBERTa	Dist. RoBERTa	BioBERT
ELECTRA	0.0000	2023.5001	1844.5974	1883.2116	1961.7430
MedELECTRA	2023.5001	0.0000	2684.9884	2722.0170	2687.3225
RoBERTa	1844.5974	2684.9884	0.0000	818.7786	1077.1062
Dist. RoBERTa	1883.2116	2722.0170	818.7786	0.0000	1113.5809
BioBERT	1961.7430	2687.3225	1077.1062	1113.5809	0.0000

5.3 Ensemble

When comparing our ensemble to the constituent transformers (Table 3), we see that the ensemble has a higher accuracy than any of the individual models, indicating that ensembling is a useful procedure which extracts relevant information from the models.

Table 3: Comparison of ensemble and best transformer accuracy for various top-k predictions.

	Ensemble Acc	Best Transformer Acc	Best Transformer
Top-1	0.3559	0.3417	RoBERTa
Top-2	0.6218	0.6181	BioBERT
Top-3	0.7383	0.7381	BioBERT

Ensemble Hyperparameters We discovered the addition of more more layers to the ensembling model caused a *reduction* in accuracy, specifically among top-2, top-3, and top-4 accuracy (Figure 4). This indicates that the ensembling model is already extracting the maximum possible information out of the constituent models — in other words, it is not possible to get higher accuracy from the models as trained. It is possible that if given more epochs to train, the more complex models would find a better minimum; however, observing the loss over time (see Supplementary Materials), it is clear that a majority of the learning occurs in the first few epochs.

Tweaking other parameters also had negligible effects on the model; for example, changing the batch size between 16 and 1024 had no meaningful effect on top-k performance for any k (Figure 5). This reinforces the claim that the model converges to the best possible weights. (It is likely that the reduced accuracy on batch size 4096 is because the model cannot reach the optimum within the 100 epochs.)

6 Discussion & Conclusion

6.1 Issues

When attempting to pre-train the ELECTRA model on a clinical notes sample, we initially fed the individual notes into the model as their own samples. However, due to the small number of samples for pre-training as well as their long length, the model did not train effectively. After properly cleaning the pre-training data and removing extraneous information, as well as splitting every paragraph into individual sentences, this issue was resolved.

One of the biggest issues our group faced was training time due to difficulties during setting up a Google spot VM. After getting TA assistance we were able to use our cloud resources to train our

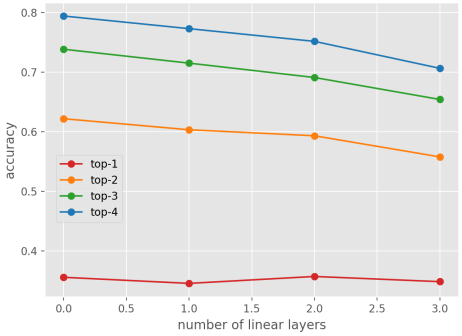

Figure 4: Accuracy over number of layers.

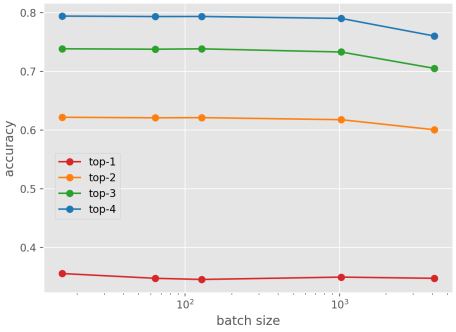

Figure 5: Accuracy over batch sizes.

models, but in order to remain within our allocated budget we had to use K80 GPUs, meaning the spot VM trained slower than Colab.

The main reason that we wanted to switch from Colab in the first place was because of Colab's time restrictions. Our classifiers each took different amounts of time to train, but they ranged from one and quarter hours with ELECTRA to over two hours with BioBERT. On top of that, we also needed to zip and download our model's once they were finished training so we could load them again in a separate notebook, which took additional time. This was a problem when we were attempting to fine-tune a BERT model, which took so long to train and download that Colab always ran out of GPU credits or timed out, forcing us to scrap that model and exclude it from our ensemble. Similar problems occurred with the XLM-RoBERTa-XL and RoBERTa-Large models.

6.2 Final Thoughts

Although we wished we could have worked more in certain areas for the project, we are rather satisfied with our conclusions. Both of our initial questions proved to be interesting to explore. After looking at our analysis from the previous section, we can say that different methods of pre-training and different pre-training datasets do change the results of a classification model. For our specific application, having both a large dataset and a large amount of medical text causes the models to be easier to fine-tune once pre-trained.

Ensembling also did increase the accuracy of our predictions, even if only slightly in the top-1 accuracy. Because we are ensembling 5 models, each of which outputs a 1 by 40 tensor of weights, the ensembling model only has (200)(40) + 40 = 8040 weights to tune while training, which is a very small number of weights. This is almost certainly why most of the decreases in loss and increases in accuracy occurred within the first few epochs, as the model had found a near-optimal solution rapidly.

7 Future Work

Given two additional weeks to work on the project, our group would first dedicate more time to training our model on Google's VMs, allowing us to train our models faster and thus spend time to tune hyperparameters, and test / tune additional models for our ensemble. In particular, two models that our group wished to train were the T5 transformer model (an encoder-decoder model that converts NLP problems into a text-to-text format and is trained using teacher-forcing) and the DeBERTa model (a transformer-based natural language model that aims to improve the BERT and RoBERTa models with the use of a disentangled attention mechanism and an enhanced mask decoder). Some of the research that we had conducted on transformers suggested that such models would be effective at completing our task compared to our self-trained models.

Furthermore, we would have also experimented with other ensemble methods. In our research, we came across 2 commonly-used methods that we did not have time to implement: boosting, which improves the accuracy of weak learners by iteratively training models on weighted versions of the data until the final model combines their predictions with high performance, and stacking, which involves training multiple models and using their predictions as features to train a meta-model that produces a final prediction.

Given an additional 2 months of time, we would have liked to also augment our training data, and find more data for model training. Ideally, our group would be able to access MIMIC-III/IV along with a number of similar clinical note datasets. This is important because, as previously mentioned, our dataset is highly unbalanced. All of the training and evaluation splits were based on a stratified sample of the initial dataset, thus meaning that they all still demonstrated the same underlying distribution as the original dataset. Therefore, labels with lower frequencies were often never predicted due to a lack of training data. A larger and more diverse sample would alleviate this issue.

Links: GitHub, Presentation, Slides, Supplementary Materials

References

[1] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and M. Roger, ""mimic-iv" (version 2.2)," *PhysioNet*, 2023.

[2] MTSamples, "Mtsamples - medical transcription samples." https://mtsamples.com/, accessed 2023-05-11.

[3] Kaggle, "Nbme score and clinical patient notes competition: Data." https://www.kaggle.com/competitions/nbme-score-clinical-patient-notes/data, accessed 2023-05-11.

[4] T. Tabosa de Oliveira, S. R. da Silva Neto, I. V. Teixeira, S. B. Aguiar de Oliveira, M. G. de Almeida Rodrigues, V. S. Sampaio, and P. T. Endo, "A comparative study of machine learning techniques for multi-class classification of arboviral diseases," *Frontiers in Tropical Diseases*, vol. 2, 2022.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[6] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, pp. 1234–1240, 09 2019.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[8] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," 2020.

[9] O. Köknar, A. Bilgin, A. S. G, and S. Yeşilyurt, "Medical-electra." https://github.com/okanvk/Medical-Electra, accessed 2023-05-11.

[10] R. Sreenivasan, "Clinical text classification." https://www.kaggle.com/code/ritheshsreenivasan/clinical-text-classification, accessed 2023-05-11.

[11] M. A. Ganaie, M. Hu, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *CoRR*, vol. abs/2104.02395, 2021.