A Federated Approach to Predict Emojis in Hindi Tweets

Anonymous ACL submission

Abstract

The use of emojis provide for adding a visual modality to textual communication. The task of predicting emojis however provides a challenge for computational approaches as emoji use tends to cluster into the frequently used and the rarely used emojis. Much of the research on emoji use has focused on high resource languages and conceptualised the task of predicting emojis around traditional servers-side machine learning approaches, which can introduce privacy concerns, as user data is transmitted to a central storage. In this paper, we provide a benchmark dataset of 118k tweets for emoji prediction in Hindi.[1] Specifically, we show that a privacy preserving approach, Federated Learning exhibits comparable performance to traditional servers-side transformer models.

1 Introduction

Since the creation of emojis around the turn of the millennium (Stark and Crawford, 2015; Alshenqeeti, 2016), they have become of a staple of informal textual communication, expressing emotion and intent in written text (Barbieri et al., 2018b). This development in communication style has prompted research into emoji analysis and prediction for English (e.g. Barbieri et al., 2018a,b; Felbo et al., 2017; Tomihira et al., 2020; Zhang et al., 2020) while comparatively little attention has been given to the low resource languages.

Emoji-prediction has posed a challenge for the research community because emojis express multiple modalities, containing visual semantics while simultaneously standing in place for words (Padilla López and Cap, 2017). The challenge is further compounded by the quantity of emojis sent and the imbalanced distribution of emoji use (Cappallo et al., 2018; Padilla López and Cap, 2017).

[1]The dataset will be made publicly available upon request.

Machine learning for emoji analysis and prediction has traditionally relied on traditional server-side architectures. However, training such models risks leaking highly sensitive information that may co-occur with emojis in texts. In contrast, federated learning (McMahan et al., 2017) approaches the task of training machine learning models by emphasising privacy of data. Such privacy is ensured by training models locally and sharing updates, rather than the data, with a central server (see Figure 1).

Motivated by prior work in privacy preserving machine learning (e.g. Ramaswamy et al., 2019; Yang et al., 2018) and emoji prediction for low resource languages (e.g. Choudhary et al., 2018), we consider the application of federated learning to the task of emoji prediction for Hindi. Specifically, we collect an imbalanced dataset of $118,030$ tweets in Hindi for emoji prediction. The dataset contains 700 unique emojis, that we classify into 10 pre-defined categories of emojis.[2] We further examine how balancing the data, using resampling and cost-sensitive re-weighting, influence a federated LSTM model and 6 server-side, centralised models: bi-directional LSTM (Hochreiter and Schmidhuber, 1997), IndicBert (Kakwani et al., 2020), HindiBERT,[3] Hindi-Electra,[4], mBERT (Devlin et al., 2019), and XLM-R (Conneau et al., 2020).

We find that the federated learning framework using simple machine learning models can provide results that are competitively with more complex models such as fine-tuned large language models. Moreover, we find that models that take into account a higher number of client updates provide for the best performing federated models, in spite of disregarding up to 50% of all available training data.

[2]These categories are obtained from the Emojis library, available at https://github.com/alexandrevicenzi/emojis.

[3]https://huggingface.co/monsoon-nlp/hindi-bert

[4]https://huggingface.co/monsoon-nlp/hindi-tpu-electra

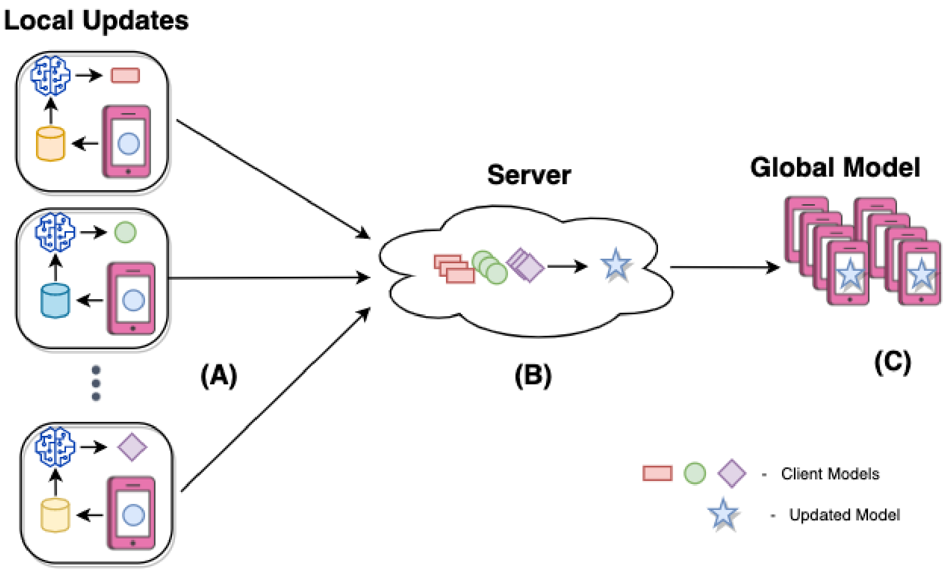


Figure 1: The Federated Learning process:(A) client devices compute updates based on the data stored locally, (B) client updates are aggregated by the server and a new global model is formed, (C) the resulting model is sent back to all the clients, and the process is repeated.

2 Prior work

Federated learning Federated Learning (McMahan et al., 2017) is a training procedure which distributes training of models onto a number of client devices. In this way, federated learning can help prevent computational bottlenecks when training models on a large corpus while simultaneously preserving privacy by not transmitting raw data. This training approach has previously been applied for on-device token prediction on mobile phones for English. In a study of the quality of mobile keyboard suggestions, Yang et al. (2018) show that federated learning improves the quality of suggested words. Addressing emoji-prediction, Ramaswamy et al. (2019) use federated learning, to improve on traditional server-based models on user devices.

Centralised training In efforts to extend emoji prediction, Ma et al. (2020) experiment with a BERT-based model on a new dataset that includes a large set of emojis for multi label prediction. Addressing the issue of low resource languages, Choudhary et al. (2018) train a bi-directional LSTM-based siamese network, jointly training their model with high resource and low resource languages.

3 Data

We collect our dataset for emoji prediction by scraping 1M tweets using the Twitter API, keeping only the 24, 794 tweets that contain at least one

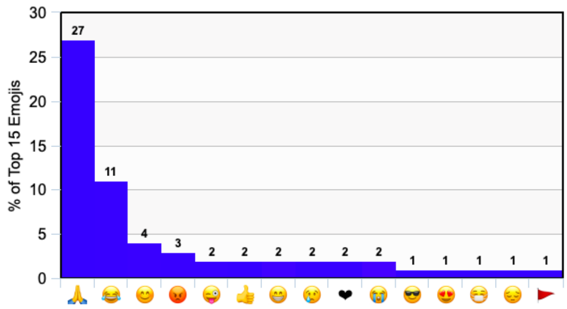


Figure 2: Distribution of 15 most frequently appearing emojis in Hindi.

emoji and are written in Hindi. For tweets that contain multiple emojis, we duplicate the tweet by the number of emojis they contain and assign a single emoji to each copy, resulting in a dataset of 118, 030 tweets with 700 unique emojis. Due to the highly imbalanced nature of emoji use in our dataset (see Figure 2), we categorise into a coarse-grained set of 10 emoji categories.

3.1 Balancing data

This dataset exhibits a long-tail in the distribution of emoji categories (see Figure 3), with the vast majority of tweets belonging to the “Smileys & Emotions” and “People & Body” categories. To address this issue, we use two different data balancing methods: re-sampling (He and Garcia, 2009) and cost-sensitive reweighting (Khan et al., 2017).

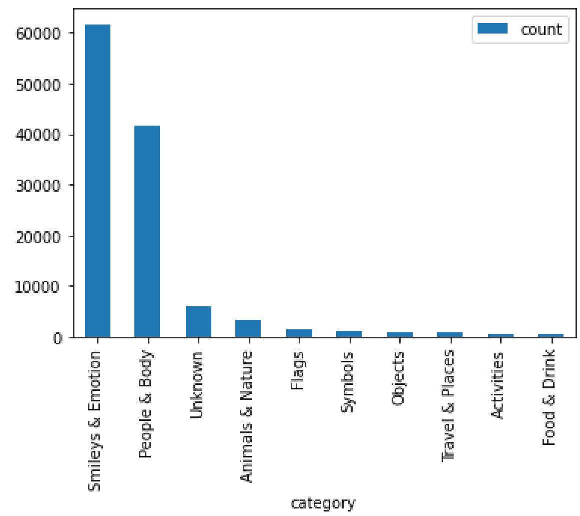


Figure 3: Category distribution of complete dataset

Re-Sampling Re-sampling has been used widely to address issues of class imbalances (e.g. Buda et al., 2018; Zou et al., 2018; Geifman and El-Yaniv, 2017; Shen et al., 2016). By balancing our data, through over-sampling the minority classes (Drumnond, 2003) and under-sampling the majority classes (Chawla et al., 2002), we obtain a dataset of $58,000$ tweets.

Cost-Sensitive Learning Similarly to re-sampling data, cost-sensitive weighting has been applied to the issue of dealing with class imbalances in data (e.g. Zhou and Liu, 2005; Huang et al., 2016; Ting, 2000; Sarafianos et al., 2018). Rather than over or under-sampling, each class is assigned a weight, that can be used to weight the loss function (Lin et al., 2017). For our models, we assign each class the inverse class frequency as its' weight:

$$W_{c \in C} = \frac{|S|}{|C| \cdot \hat{y}} \tag{1}$$

where $|S|$ is the number of samples and $|C|$ is the number of classes.

3.2 Pre-processing

We pre-process all tweets by lower-casing all text and removing numbers, punctuation, and URLS. We also remove Twitter specific such as hashtags, @-mentions, and the retweet marker: "RT:".

4 Experiments

We conduct our experiments using PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020) on Google Colab using a Nvidia Tesla V100 GPU with 26GB of RAM. The datasets are split into train (80%), validation (10%), and test sets (10%). We measure our performance with the accuracy, AUC, and weighted F1 metrics. Each model is trained and evaluated on the original imbalanced data and the two balancing approaches described in Section 3.1. Finally, for the federated setting, we conduct experiments where data is independent and identically distributed (I.I.D.) across the different client nodes.

4.1 Baseline models

We use 6 centralised models as baselines to compare the federated approach against. Specifically, we use a bi-LSTM (Hochreiter and Schmidhuber, 1997) with 2 hidden layers and dropout,[5] two multilingual models: mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020). Finally we use IndicBERT (Kakwani et al., 2020), HindiBERT, and Hindi-Electra as these are pre-trained on indic languages.[6] All baselines are trained with batch size 8, learning rate $4e-5$, and seq. length 128.

4.2 Federated models

For our federated experiments, we use the FedProx algorithm (Li et al., 2018) which trains a federated model by considering dissimilarity among the local gradient updates while preventing divergence under non-I.I.D. settings through adding a proximal term to the loss function.[7] We reuse the bi-LSTM architecture from Section 4.1 as our experimental model to limit the computational power required from client devices. For our experiments, we set the number of clients to 100 and simulate I.I.D. and non-I.I.D. settings. We simulate an I.I.D. setting by ensuring that all client devices receive data that is representative of the entire dataset. For the non-I.I.D. setting, we create severely imbalanced data splits for clients by first grouping the data by label, then splitting the grouped data into 200 bins and randomly assigning 2 bins to each client. We experiment with three different settings, in which we randomly select 10%, 30%, and 50% of all clients whose updates are incorporated into the global model.

[5] We set the dropout to 0.5

[6] IndicBERT is pre-trained on 12 indic languages, HindiBERT and Hindi-Electra are both trained on Hindi Wikipedia and CommonCrawl.

[7] We follow Li et al. (2018) in setting the value of the proximal term to 0.01.

	Bi-LSTM			mBERT			XLM-R			IndicBERT			hindiBERT			Hindi-Electra		
	Acc.	AUC	F1	Acc.	AUC	F1	Acc.	AUC	F1	Acc.	AUC	F1	Acc.	AUC	F1	Acc.	AUC	F1
Imbalanced	62.38	77.53	63.83	66.90	75.96	64.50	**70.39**	**83.09**	**69.44**	68.21	82.13	67.60	66.53	81.62	65.90	52.30	49.64	35.91
Re-sampled	53.95	78.97	58.61	53.43	78.79	56.58	60.76	**82.45**	63.39	**62.44**	80.00	**64.58**	55.16	78.60	57.92	52.30	50.10	60.30
Cost-Sensitive	61.19	79.76	64.46	62.73	79.42	63.30	**68.33**	**83.05**	**68.87**	67.98	82.70	68.66	65.32	81.55	66.06	52.29	50.10	35.91

Table 1: Centralised model performances.

	c = 10%						c = 30%						c = 50%					
	IID			non-IID			IID			non-IID			IID			non-IID		
	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1	Acc	AUC	F1
Imbalanced	63.32	70.27	62.32	63.78	60.96	57.96	**66.56**	67.13	42.68	56.99	62.56	54.86	65.91	66.86	63.57	59.41	61.92	58.09
Re-sampled	45.19	78.60	51.12	32.92	74.10	34.28	42.68	80.23	49.19	35.27	75.47	41.36	47.10	**80.66**	52.14	40.22	**75.71**	45.76
Cost-Sensitive	63.24	76.42	61.99	**64.81**	60.96	**61.25**	64.16	76.71	**63.78**	55.70	60.37	54.36	60.22	74.91	59.57	61.29	59.44	59.36

Table 2: Results of experiments run using the Federated Learning setup. c is the percentage of clients whose updates are considered.

Approach	**Results**	
	Server trained	Federated
Imbalanced	83.09	67.13
Balanced	82.45	80.23
Cost-Sensitive	83.05	76.71

Table 3: Results comparing the AUC scores for server-based approach(XLM-R) and federated approach on IID partitioning with client fraction of 0.3

4.3 Analysis

Considering first the results for our baseline models (see Table 1), we find that XLM-R and IndicBERT obtain the best performances. Across all baselines, with the exception of Hindi Electra, we find that using a cost-sensitive weighting out-performs balancing through re-sampling the dataset. Moreover, we find that the cost-sensitive weighting performs comparably or out-performs all other settings. Curiously, we find that Hindi Electra under-performs compared to all other models, including HindiBERT which is a smaller model trained on the same data. This discrepancy in the performances of these two models may be due to the differences in complexity, and thus data required to achieve competitive performances. Finally, the bi-LSTM slightly under-performs in comparison to XLM-R, however it obtains competitive performances with all other well-performing models.

Turning to results of our experiments with federated models (see Table 2), we note that the federated models achieve slightly lower, but comparative performances with the centralised models, across data distribution setting for clients. Considering the performance in terms of AUC given the number of clients' update considered, we find that the best performing setting is 50%, for both I.I.D. and non-I.I.D. settings. However, if we are to consider accuracy or F1, the best performing client percentage varies depending on whether the data is I.I.D. or not. In terms of the data balancing schemes, we find that the imbalanced label distribution tends to achieve low performances in terms of AUC and high performances in terms of accuracy, while the cost-sensitive strategy often offers a more balanced performance. In contrast, the re-sampled label balancing technique provides highly variable results. For the I.I.D. setting, the scores posted are comparable with the other label balancing schemes while the non-I.I.D. setting produces very high scores in terms of AUC but poor performances in terms of accuracy and F1.

5 Conclusion

Emoji prediction in user-generated text is a task which entails potentially highly private data, for which reason it is important to consider privacy-preserving methods for the task. Moreover, while emojis are used by people around the world, the majority of academic research has been focused on English. Here, we presented a new dataset for emoji for Hindi and compare a privacy preserving machine learning approach, Federated Learning, with traditional, centralised machine learning methods. Experimenting with the different data balancing methods and simulating settings where data is I.I.D. and where it is not, we find that using federated learning can afford competitive performances with more complex centralised machine learning methods, such as fine-tuned language models, while ensuring that user data is kept secure by only sharing model weight updates rather than the raw data.

References

Hamza Alshenqeeti. 2016. Are emojis creating a new or old visual language for new generations? a socio-semiotic study. *Advances in Language and Literary Studies*, 7(6).

Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018a. SemEval 2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 24–33, New Orleans, Louisiana. Association for Computational Linguistics.

Francesco Barbieri, Luis Espinosa-Anke, Jose Camacho-Collados, Steven Schockaert, and Horacio Saggion. 2018b. Interpretable emoji prediction via label-wise attention LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4766–4771, Brussels, Belgium. Association for Computational Linguistics.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

Spencer Cappallo, Stacey Svetlichnaya, Pierre Garrigues, Thomas Mensink, and Cees GM Snoek. 2018. New modality: Emoji challenges in prediction, anticipation, and retrieval. *IEEE Transactions on Multimedia*, 21(2):402–415.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Nurendra Choudhary, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. 2018. Contrastive learning of emoji-based representations for resource-poor languages. *arXiv preprint arXiv:1804.01855*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Drumnond. 2003. Class imbalance and cost sensitivity: Why undersampling beats oversampling. In *ICML-KDD 2003 Workshop: Learning from Imbalanced Datasets*, volume 3.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

Yonatan Geifman and Ran El-Yaniv. 2017. Deep active learning over the long tail. *arXiv preprint arXiv:1711.00941*.

Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. 2017. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Weicheng Ma, Ruibo Liu, Lili Wang, and Soroush Vosoughi. 2020. Emoji prediction: Extensions and benchmarking. *arXiv preprint arXiv:2007.07389*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Rebeca Padilla López and Fabienne Cap. 2017. Did you ever read about frogs drinking coffee? investigating the compositionality of multi-emoji expressions. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 113–117, Copenhagen, Denmark. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and et al. 2019. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, page 8024–8035. Curran Associates, Inc.

Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. 2019. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*.

Nikolaos Sarafianos, Xiang Xu, and Ioannis A Kakadiaris. 2018. Deep imbalanced attribute classification using visual attention aggregation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 680–697.

Li Shen, Zhouchen Lin, and Qingming Huang. 2016. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer.

Luke Stark and Kate Crawford. 2015. The conservatism of emoji: Work, affect, and communication. *Social Media+ Society*, 1(2):2056305115604853.

Kai Ming Ting. 2000. A comparative study of cost-sensitive boosting algorithms. In *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer.

Toshiki Tomihira, Atsushi Otsuka, Akihiro Yamashita, and Tetsuji Satoh. 2020. Multilingual emoji prediction using bert for sentiment analysis. *International Journal of Web Information Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*.

Linrui Zhang, Yisheng Zhou, Tatiana Erekhinskaya, and Dan Moldovan. 2020. Emoji prediction: A transfer learning approach. In *Future of Information and Communication Conference*, pages 864–872. Springer.

Zhi-Hua Zhou and Xu-Ying Liu. 2005. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77.

Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305.

A Appendix

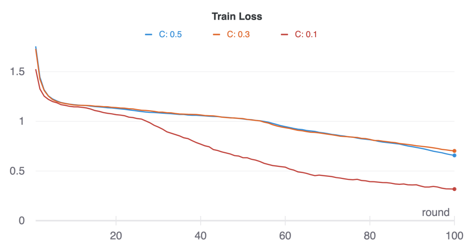
A.1 Server-Based Models

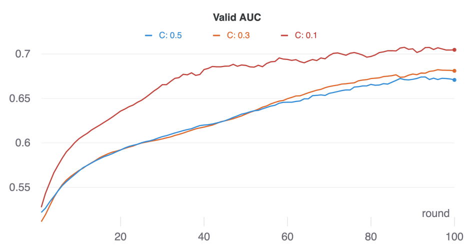
For traditional server-side transformer models, we use the simpletransformers[8] library. We use the default configuration options. We train all the transformer models for 25 epochs with a learning rate of 4e-5 and no weight decay or momentum.

A.2 Federated Learning Plots

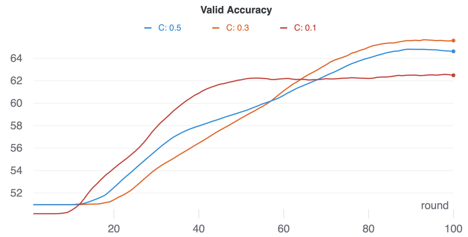
This section provides detailed graphs comparing the training loss, validation AUC, validation F1 score and validation accuracy for every dataset variation. All of these graphs were made using Weights and Biases (Biewald, 2020).

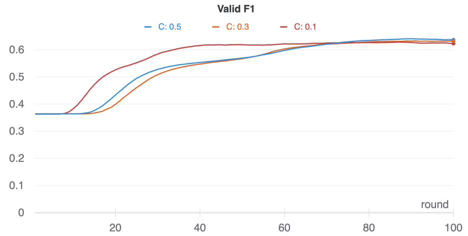
A.2.1 Imbalanced Dataset (IID)



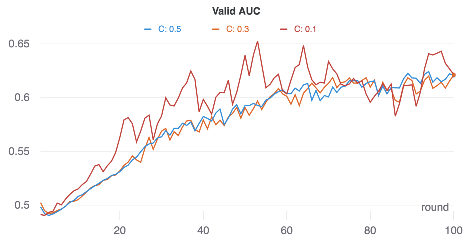


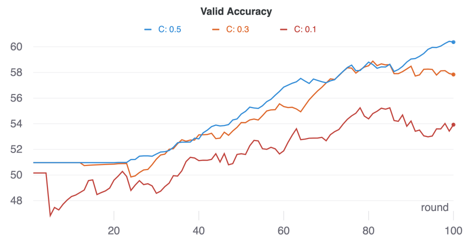
[8] https://simpletransformers.ai/

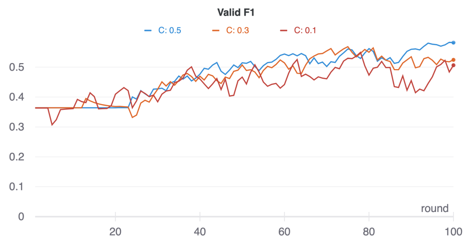




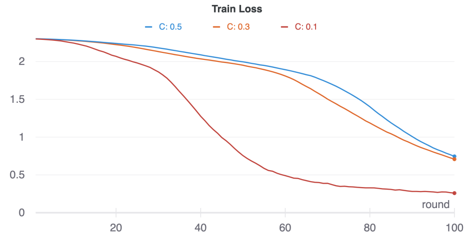
A.2.2 Imbalanced Dataset (non-IID)

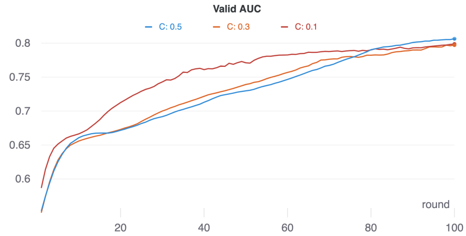


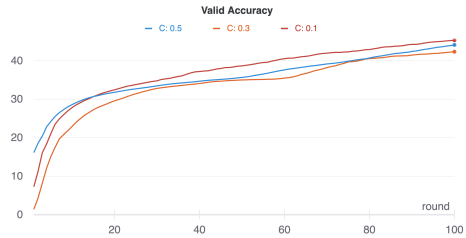


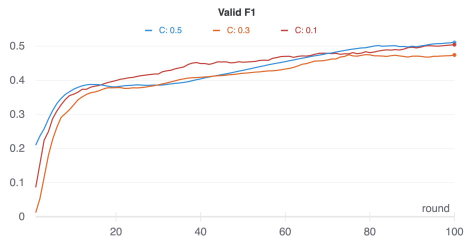


A.2.3 Balanced Dataset (IID)

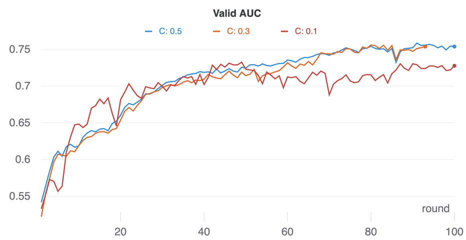


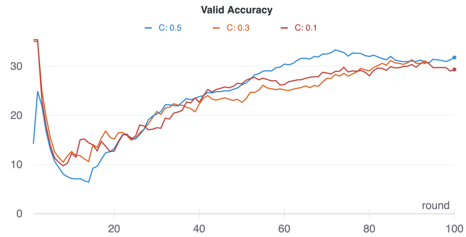


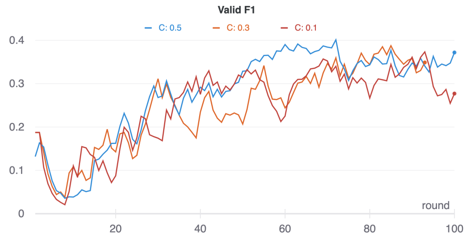




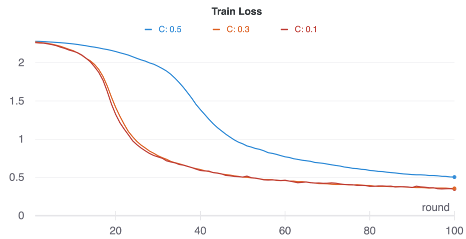
A.2.4 Balanced Dataset (non-IID)

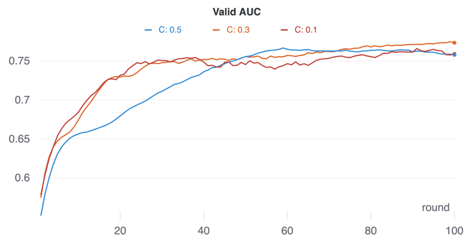


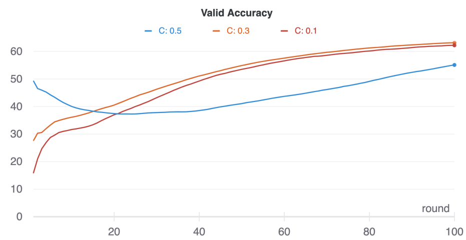


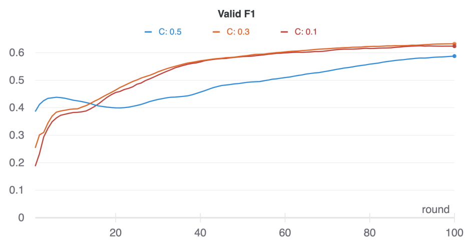


A.2.5 Cost Sensitive Approach (IID)

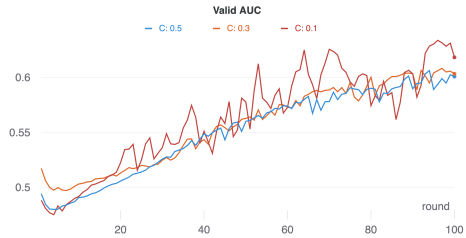


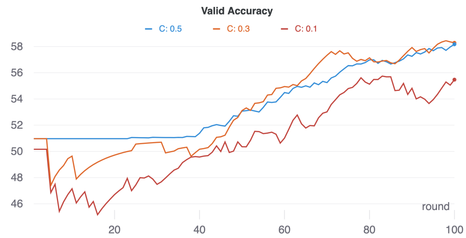


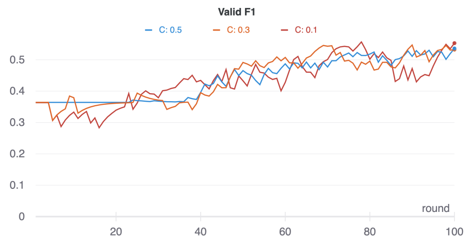




A.2.6 Cost Sensitive Approach (non-IID)



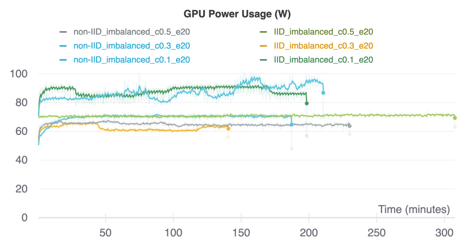




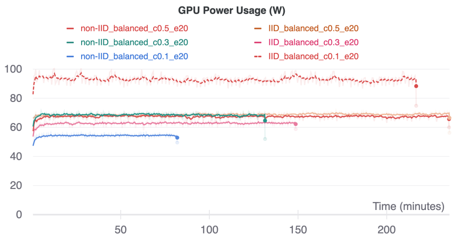
A.3 Time vs GPU Usage

This section provides detailed graphs for GPU usage in Watts for every variation of experiments run.

A.3.1 Imbalanced Dataset



A.3.2 Balanced Dataset



A.3.3 Cost Sensitive Approach