A Federated Approach to Predicting Emojis in Hindi Tweets

Deep Gandhi*[1], Jash Mehta*[2], Nirali Parekh[3], Karan Waghela[4],
Lynette D'Mello[5], Zeerak Talat[6]

[1]University of Alberta, [2]Georgia Institute of Technology [3]Stanford University
[4]Santa Clara University, [5]DJ Sanghvi College of Engineering, [6]Simon Fraser University
[1]`drgandhi@ualberta.ca`, [2]`jmehta73@gatech.edu`, [3]`nirali25@stanford.edu`
[4]`kwaghela@scu.edu`, [5]`lynette.dmello@djsce.ac.in`, [6]`zeerak_talat@sfu.ca`

Abstract

The use of emojis affords a visual modality to, often private, textual communication. The task of predicting emojis however provides a challenge for machine learning as emoji use tends to cluster into the frequently used and the rarely used emojis. Much of the machine learning research on emoji use has focused on high resource languages and has conceptualised the task of predicting emojis around traditional server-side machine learning approaches. However, traditional machine learning approaches for private communication can introduce privacy concerns, as these approaches require all data to be transmitted to a central storage. In this paper, we seek to address the dual concerns of emphasising high resource languages for emoji prediction and risking the privacy of people's data. We introduce a new dataset of 118k tweets (augmented from 25k unique tweets) for emoji prediction in Hindi,[1] and propose a modification to the federated learning algorithm, CausalFedGSD, which aims to strike a balance between model performance and user privacy. We show that our approach obtains comparative scores with more complex centralised models while reducing the amount of data required to optimise the models and minimising risks to user privacy.

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). Comparatively little research attention has been given to the low resource languages.

Emoji-prediction has posed a challenge for the research community because emojis express multiple modalities, contain visual semantics and the ability to stand 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). Machine learning (ML) for emoji analysis and prediction has traditionally relied on traditional server-side architectures. However, training such models risks leaking sensitive information that may co-occur with emojis or be expressed through them. This can lead to potential breaches of data privacy regulation (e.g. the European General Data Protection Regulation and the California Consumer Privacy Act). In contrast, federated learning (FL) (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 weight updates, rather than the data, with a central server (see Figure 1). The FL approach assumes that some client-updates may be corrupted during transmission. FL therefore aims to retain predictive performance while emphasising user privacy in scenarios with potential data loss.

Motivated by prior work in privacy preserving ML (e.g. Ramaswamy et al., 2019; Yang et al., 2018) and emoji prediction for low resource languages (e.g. Choudhary et al., 2018b), we examine the application of FL to emoji topic prediction for Hindi. Specifically, we collect an imbalanced dataset of $118,030$ tweets in Hindi which contain 700 unique emojis that we classify into 10 pre-defined categories of emojis. The dataset contains 700 unique emojis, that we classify into 10 pre-defined categories of emojis.[2] We further

---

*Equal contribution.
[1]The dataset and code can be accessed at https://github.com/deep1401/fedmoji

[2]These categories are obtained from the Emojis library, available at https://github.com/alexandrevicenzi/emojis.
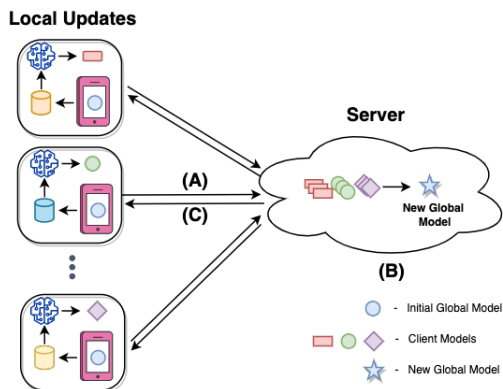
Figure 1: The Federated Learning process: (A) client devices compute weight updates on locally stored data, (B) client weight updates are transmitted to the server and used to update the global model, (C) the resulting global model is redistributed to all clients.

examine the impact of two different data balancing strategies on federated and server-side, centralised model performance. Specifically, we examine: re-sampling and cost-sensitive re-weighting. We consider 6 centralised models which form our baselines: 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); and LSTMs trained using two FL algorithms: FedProx (Li et al., 2018) and a modified version of CausalFedGSD (Francis et al., 2021).

We show that LSTMs trained using FL perform competitively with more complex, centralised models in spite of only using up to 50% of the data.

2 Prior work

Federated Learning Federated Learning (FL, McMahan et al., 2017) is a training procedure that distributes training of models onto a number of client devices. Each client device locally computes weight updates on the basis of local data, and transmits the updated weights to a central server. In this way, FL 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 FL improves the quality of suggested words. Addressing emoji-prediction

in English, Ramaswamy et al. (2019) use the FederatedAveraging algorithm, to improve on traditional server-based models on user devices. We diverge from Ramaswamy et al. (2019) by using the CausalFedGSD and FedProx algorithms on Hindi tweets. FedProx develops on the FederatedAveraging algorithm by introducing a regularization constant to it (Li et al., 2018). In related work, Choudhary et al. (2018b) seek to address the question of FL for emoji prediction for low resource languages. However, the dataset that they use, Choudhary et al. (2018a) relies on emojis that are frequently used in English text and therefore may not be representative of emoji use in other, low resource languages.

Centralised Training In efforts to extend emoji prediction, Ma et al. (2020) experiment with a BERT-based model on a new English dataset that includes a large set of emojis for multi label prediction. Addressing the issue of low resource languages, Choudhary et al. (2018b) train a bidirectional LSTM-based siamese network, jointly training their model with high resource and low resource languages. A number of studies on emoji prediction have been conducted in lower-resourced languages than English (e.g. Liebeskind and Liebeskind, 2019; Ronzano et al., 2018; Choudhary et al., 2018a; Barbieri et al., 2018a; Duarte et al., 2020; Tomihira et al., 2020). Common to these approaches is the use of centralised ML models, which increase risks of breaches of privacy. In our experiments, we study using FL for emoji topic classification in low resource settings.

3 Data

We collect our dataset for emoji topic prediction by scraping $\sim$1M tweets. We only keep the $24,794$ tweets that are written in Hindi and contain at least one emoji. We duplicate all tweets that contain multiple emojis by the number of emojis contained, assigning a single emoji to each copy, resulting in a dataset of $118,030$ tweets with 700 unique emojis. Due to the imbalanced distribution of emojis in our dataset (see Figure 2), we assign emojis into 10 coarse-grained categories. This reduction i.e., from multi-label to multi-class and unique emojis into categories, risks losing the semantic meaning of emojis. Our decision is motivated by how challenging emoji prediction is without such reductions (Choudhary et al., 2018b).

We pre-process our data to limit the risk of overfitting to rare tokens and platform specific tokens.

---

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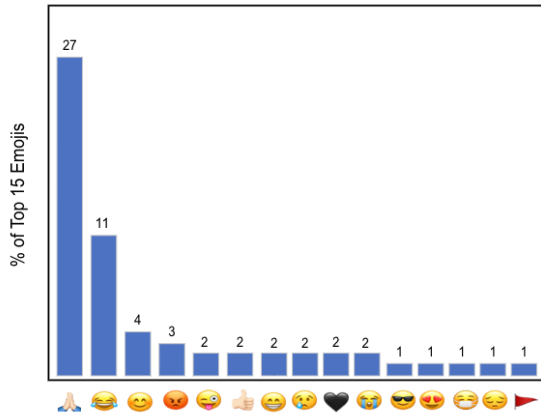[4]https://huggingface.co/monsoon-nlp/hindi-tpu-electra

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

For instance, we lower-case all text and remove numbers, punctuation, and retweet markers. We replace mentions, URLs, and hashtags with specific tokens to avoid issues of over-fitting to these.

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 re-weighting (Khan et al., 2017).

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). We balance the training data by up-sampling the minority class (Drumnond, 2003) and down-sampling the majority class (Chawla et al., 2002), resulting in a balanced dataset of $94,420$ tweets ($9,442$ documents per class). The validation and test sets are left unmodified to ensure a fair and realistic evaluation.

Cost-Sensitive learning Another method for addressing data imbalances is cost-sensitive learning (see Zhou and Liu, 2005; Huang et al., 2016; Ting, 2000; Sarafianos et al., 2018). In this method, each class is assigned a weight which is used to weigh the loss function (Lin et al., 2017). For our models, we set the class weights to the inverse class frequencies.

4 Experiments

We conduct experiments with PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020) on

Google Colab using a Nvidia Tesla V100 GPU with 26GB of RAM. We create train, validation, and test ($80/10/10$) splits of the dataset, and measure performances using precision, recall, and weighted F1. All models are trained and evaluated on the imbalanced data and the two data balancing methods (see §3.1). For the FL setting, we conduct experiments manipulating the independent and identically distributed (I.I.D.) data assumption on client nodes.

4.1 Baseline models

We use 6 centralised models as baselines for comparison with the federated approach. Specifically, we use a bi-LSTM (Hochreiter and Schmidhuber, 1997) with 2 hidden layers and dropout at 0.5; two multi-lingual models: mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020); and three models pre-trained on Indic languages: IndicBert (Kakwani et al., 2020), HindiBERT, and Hindi-Electra.

4.2 Federated models

For our federated learning experiments, we use the FedProx (Li et al., 2018) algorithm and a modification of the CausalFedGSD (Francis et al., 2021) algorithm. FedProx trains models by considering the dissimilarity between local gradients and adds a proximal term to the loss function to prevent divergence from non-I.I.D. data. CausalfedGSD reserves 30% of the training data for initializing client nodes. When a client node is created, it receives a random sample of the reserved data for initialization. In our modification, we similarly reserve 30% of the training data, however we diverge by using the full sample to initialize the global model, which is then transmitted to client nodes. This change means that (i) user data is transmitted fewer times; (ii) modellers retain control over the initialization of the model, e.g. to deal with class imbalances; and (iii) models converge faster, due to exposing client nodes to the distribution of all classes (see Appendix A.6).

We reuse the Bi-LSTM (see Section 4.1) as our experimental model on client devices due to its relative low compute requirements. 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

	Bi-LSTM			mBERT			XLM-R			IndicBERT			hindiBERT			Hindi-Electra		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Imbalanced	64.72	64.26	63.83	63.25	66.90	64.50	**68.74**	**70.39**	**69.44**	67.15	68.22	67.60	65.39	66.53	65.90	27.34	52.29	35.91
Re-sampled	64.42	55.41	58.61	62.18	53.43	56.58	67.92	60.76	63.39	68.04	62.44	64.58	62.95	55.16	57.92	64.42	57.93	60.30
Cost-Sensitive	68.41	62.27	64.46	63.99	62.73	63.30	69.79	68.33	68.87	69.54	67.98	68.66	66.97	65.32	66.06	27.34	52.29	35.91

Table 1: Centralised model performances.

	c = 10%						c = 30%						c = 50%					
	IID			non-IID			IID			non-IID			IID			non-IID		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Imbalanced	60.94	66.11	62.99	61.05	39.68	25.82	61.10	66.01	63.04	57.88	66.39	61.64	**61.11**	**66.91**	**63.35**	56.94	63.82	57.06
Re-sampled	60.89	46.01	50.89	60.83	22.52	23.04	60.58	46.58	51.22	57.38	35.85	37.37	60.78	47.14	51.63	53.16	36.83	41.95
Cost-Sensitive	60.45	59.50	59.50	60.47	41.24	28.92	60.91	60.99	60.47	55.76	56.77	52.69	61.56	60.39	60.48	56.51	63.41	56.80

Table 2: Results using the FedProx algorithm. c is the percentage of clients whose updates are considered.

	c = 10%						c = 30%						c = 50%					
	IID			non-IID			IID			non-IID			IID			non-IID		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Imbalanced	60.89	65.04	62.34	58.09	60.84	56.96	61.40	66.08	**63.22**	57.68	66.51	61.21	60.69	66.33	62.90	57.90	57.84	47.85
Re-sampled	62.14	44.95	50.27	53.31	34.01	39.69	**62.46**	45.39	50.67	58.21	27.61	27.61	61.86	46.00	51.15	59.84	22.57	28.83
Cost-Sensitive	61.62	61.15	60.92	56.72	65.20	60.51	62.17	61.97	61.60	56.86	65.54	60.59	60.50	61.28	60.44	59.33	60.56	54.22

Table 3: Results using the modified CausalFedGSD. c is the percentage of clients whose updates are considered.

	c = 10%			c = 30%			c = 50%		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Imbalanced	63.95	64.19	63.67	64.23	64.44	63.91	64.16	64.28	63.78
Re-sampled	63.07	51.14	55.08	62.84	52.04	55.71	62.84	51.72	55.50
Cost-Sensitive	66.72	64.96	65.38	66.66	64.84	65.27	**66.78**	**65.08**	**65.47**

Table 4: Results for the baseline CausalFedGSD. c is the client fraction per round.

Approach	Centralised	Federated	
	XLM-R	FedProx	Modified CausalFedGSD
Imbalanced	69.44	63.35	63.22
Re-sampled	63.39	51.63	51.15
Cost-Sensitive	68.87	60.48	61.60

Table 5: An approach-wise comparison of F1 scores for best performing models in centralized and federated settings.

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.

4.3 Analysis

Considering the results for our baseline models (see Table 1), we find that XLM-R and IndicBERT obtain the best performances. Further, using cost-sensitive weighting tends to out-perform re-sampling the dataset. In fact, the cost-sensitive weighting performs comparatively, or out-performs, other settings. Curiously, we see 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 models may be due to differences in complexity, and thus data required to achieve competitive performances.[5] Finally, the bi-LSTM slightly under-performs in comparison to XLM-R, however it performs competitively with all other well-performing models.

Turning to the performance of the federated baselines (see Table 2), we find an expected performance of the models.[6] Generally, we find that the federated models achieve comparative performances, that are slightly lower than the centralised systems. This is due to the privacy-performance trade-off, where the increased privacy offsets a small performance loss. Considering the F1-scores, we find that the optimal setting of the ratio of clients is subject to the data being I.I.D. In contrast, models trained on the re-sampled data tend to prefer data in an I.I.D. setting, but in general underperform in comparison with other weighting strategies, including the imbalanced sample. Using our modification of the CausalFedGSD algorithm, we show improvements over our FL baselines when the data is I.I.D. and variable performance for a non-I.I.D. setting (see Table 3). Comparing the results of the best performing settings, we find that the FL architectures perform comparably with the centralised models, in spite of being exposed to less data and preserving privacy of users (see Table 5). Table 4 refers to the results for the I.I.D. experiments of the baseline CausalFedGSD algorithm (Francis et al., 2021). We also observe a difference in optimization time for both models (see Appendix A.6). Models trained using our modification of CausalFedGSD converges faster than the original CausalFedGSD, which in turn converges much faster than FedProx. Moreover, we find indications that the original CausalFedGSD algorithm may be prone to over-fitting, as performance stagnates without fluctuations, while our modification shows fluctuations in performance that are similar to those of the FedProx models.

---

[5]The developers of Hindi Electra also note similar underperformance on other tasks.

[6]Please refer to the appendices for additional details on model performance and training.

5 Conclusion

Emoji topic prediction in user-generated text is a task which can contains highly private data. It is therefore important to consider privacy-preserving methods for the task. Here, we presented a new dataset for the task for Hindi and compared a privacy preserving approach, Federated Learning, with the centralised server-trained method. We present a modification to the CausalFedGSD algorithm, and find that it converges faster than our other experimetnal models. In our experiments with different data balancing methods and simulations of I.I.D. and non-I.I.D. settings, we find that using FL affords comparable performances to the more complex fine-tuned language models that are trained centrally, while ensuring privacy. In future work, we plan to extend this work to multi-label emoji topic prediction and investigate strategies for dealing with decay of the model vocabulary.

Ethical considerations

The primary reason for using federated learning is to ensure user-privacy. The approach can then stand in conflict with open and reproducible science, in terms of data sharing. We address this issue by making our dataset open to the public, given that researchers provide an Institutional Review Board (IRB) approval and a research statement that details the methods and goals of the research, where IRB processes are not implemented. For researchers who are at institutions without IRB processes, data will only be released given a research statement that also details potential harms to participants. The sharing of data will follow Twitter's developer agreement, which allows for 50k Tweet objects to be shared. We will further provide the code to our 24k tweets into the full dataset of 118k.

Our modification of the CausalFedGSD model introduces the concern of some data being used to initialise the model. Here a concern can be that some data will be available globally. While this concern is justified, the use of FL affords two things: First, FL can limit on the overall amount of raw data that is transmitted and risks exposure. Second, initialisation can occur using synthetic data, created for the express purposes of model initialisation. Moreover, pre-existing public, or privately owned, datasets can be used to initialise models, which can be further trained given weight updates provided by the client nodes. Federated learning, and our approach to FL thus reduce the risks of ex-posing sensitive information about users, although the method does not completely remove such risks.

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, Vijjini Anvesh Rao, and Manish Shrivastava. 2018a. Twitter corpus of resource-scarce languages for sentiment analysis and multilingual emoji prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1570–1577, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nurendra Choudhary, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. 2018b. 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 Drummond. 2003. Class imbalance and cost sensitivity: Why undersampling beats oversampling. In *ICML-KDD 2003 Workshop: Learning from Imbalanced Datasets*, volume 3.

Luis Duarte, Luís Macedo, and Hugo Gonçalo Oliveira. 2020. Emoji prediction for portuguese. In *International Conference on Computational Processing of the Portuguese Language*, pages 174–183. Springer.

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.

Sreya Francis, Irene Tenison, and Irina Rish. 2021. Towards causal federated learning for enhanced robustness and privacy. *arXiv preprint arXiv:2104.06557*.

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*.

Chaya Liebeskind and Shmuel Liebeskind. 2019. Emoji prediction for hebrew political domain. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 468–477, New York, NY, USA. Association for Computing Machinery.

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*.

Francesco Ronzano, Francesco Barbieri, Endang Wahyu Pamungkas, Viviana Patti, Francesca Chiusaroli, et al. 2018. Overview of the evalita 2018 italian emoji prediction (itamoji) task. In *6th Evaluation Campaign of Natural Language*

Processing and Speech Tools for Italian. Final Workshop, EVALITA 2018, volume 2263, pages 1–9. CEUR-WS.

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 Data

The tweets were collected using an "Elevated access" to the Twitter API v2. To collect tweets written in the Hindi language, we use "lang:hi" query. No other search criteria is used. The time-span of the tweets is from 19th April, 2021 to 8th May, 2021. Figure 4 shows a sample of tweets present in our Hindi dataset for the task of emoji prediction.
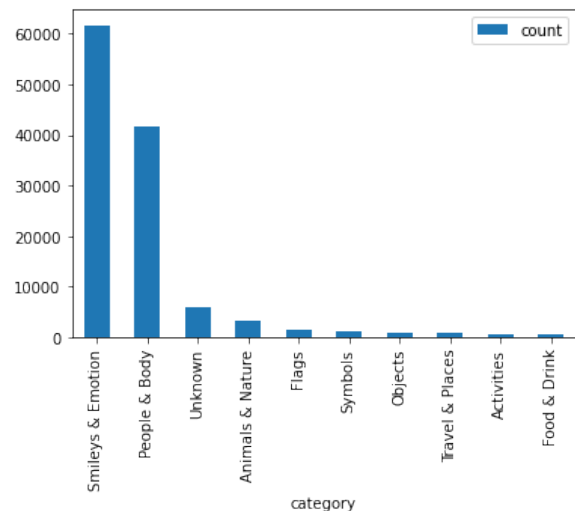

Figure 3: Category distribution of complete dataset

A.2 Server-Based Models

For traditional server-side transformer models, we use the simple transformers library.[7] We use the default configuration options and train all the transformer models for 25 epochs with a learning rate of 4e-5 and no weight decay or momentum.

All baseline models (transformer-based and others) are trained with batch size 8, learning rate $4e-5$, and seq. length 128.

All the models were trained using deterministic algorithms for randomness[8] in PyTorch and are easily reproducible using the same seeds.

A.3 Experiments considering only text

We run some additional experiments considering only the actually text without applying the token setup as described in §3 as reflected in Tables 6, 7, 8, and 9. Observing these results, we note that except for non-I.I.D. settings, we see negligible improvements from applying the extra tokens in the original dataset (see §3).

---

[7]https://simpletransformers.ai/
[8]https://pytorch.org/docs/stable/notes/randomness.html

Lang	Text	Emoji	Label
Hindi	बिलकुल सही कहा आपने भाई	👍	People & Body
English	You are absolutely right brother		
Hindi	याद रखना सिर्फ प्रेम अंधा होता है घरवाले और कॉलोनी वाले नहीं जनहित में जारी	🙈	Smileys & Emotion
English	Remember that only love is blind and not your family and the colony. Spreading the word in public interest		
Hindi	तुझे कितना चाहने लगे हम	🎧	Objects
English	How much we love you		
Hindi	एकदम जबरदस्त भावनात्मक मीठास शब्दों के सरसरी हवाओं में कुछ अजीब सी खुशबू।	🌹	Animals & Nature
English	Some strange fragrance in the whispering winds of very emotional sweet words		
Hindi	जन्मदिन की अनंत शुभकामनायें	🎂	Food & Drink
English	Best wishes for your birthday		

Figure 4: Example of our Hindi dataset

	c = 10%						c = 30%						c = 50%					
	IID			non-IID			IID			non-IID			IID			non-IID		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Imbalanced	61.33	64.66	62.32	57.70	64.10	57.96	61.55	**67.64**	63.60	58.01	58.42	54.86	61.65	66.83	63.57	58.30	61.59	58.09
Re-sampled	61.49	46.22	51.12	56.84	30.06	34.28	60.60	43.75	49.19	57.48	35.32	41.36	60.85	47.71	52.14	56.13	41.28	45.76
Cost-Sensitive	62.14	63.35	61.99	58.08	65.86	61.25	**63.72**	65.25	**63.78**	56.39	57.76	54.36	60.36	59.99	59.57	56.68	63.22	59.36

Table 6: Results using the FedProx algorithm on the dataset as explained in A.3.

	c = 10%						c = 30%						c = 50%					
	IID			non-IID			IID			non-IID			IID			non-IID		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Imbalanced	**61.83**	67.24	**63.87**	58.96	45.88	38.34	61.62	67.11	63.41	58.95	63.80	60.58	61.66	**67.38**	63.70	59.46	49.39	43.88
Re-sampled	59.44	37.53	43.68	53.10	49.91	41.50	59.53	41.06	46.54	58.61	26.68	32.45	60.97	39.02	45.48	57.70	32.98	39.71
Cost-Sensitive	60.88	59.38	59.49	54.82	57.42	46.17	60.45	60.71	59.96	59.05	66.52	62.09	60.44	61.41	60.38	58.69	63.60	60.11

Table 7: Results using the modified CausalFedGSD on the dataset as explained in A.3.

	c = 10%			c = 30%			c = 50%		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Imbalanced	62.93	63.37	62.68	63.01	63.48	62.73	63.21	63.60	62.90
Re-sampled	60.84	49.59	53.30	60.72	49.45	53.11	60.47	49.08	52.76
Cost-Sensitive	64.56	63.70	63.65	**64.61**	**63.88**	**63.70**	64.27	63.41	63.33

Table 8: Results for the baseline CausalFedGSD on the dataset as explained in A.3.

| Approach	Centralised	Federated	
	XLM-R	FedProx	Modified CausalFedGSD
Imbalanced	69.44	63.60	63.87
Re-sampled	63.39	52.14	46.54
Cost-Sensitive	68.87	63.78	62.09

Table 9: An approach-wise comparison of F1 scores for best performing models in centralized and federated settings trained on the dataset as explained in A.3.

A.4 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).
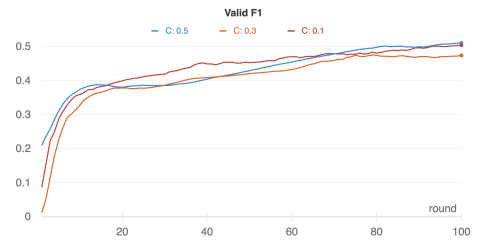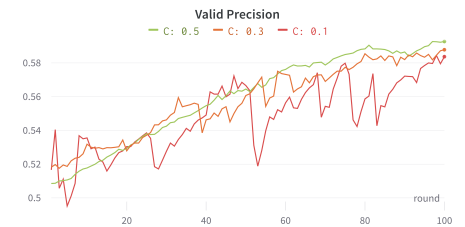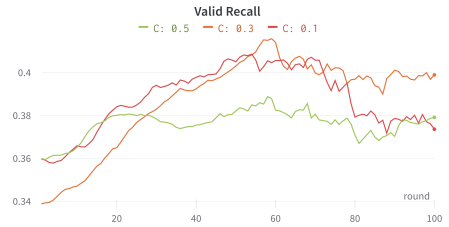
We set the value of the proximal term to 0.01 following Li et al. (2018). We set the learning rate as $1e - 02$ for Federated models based on hyperparameter sweeps.
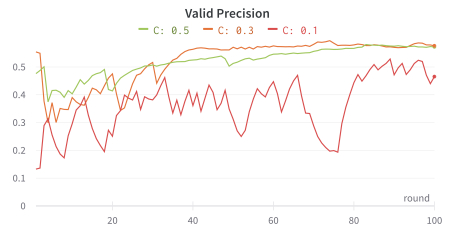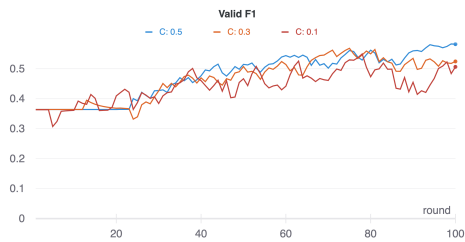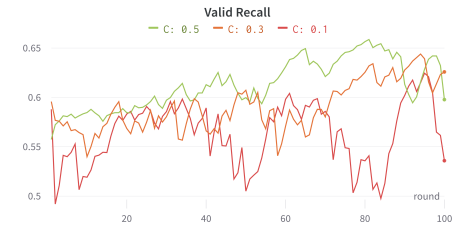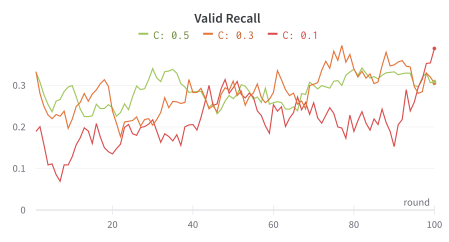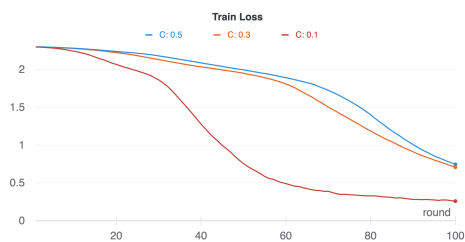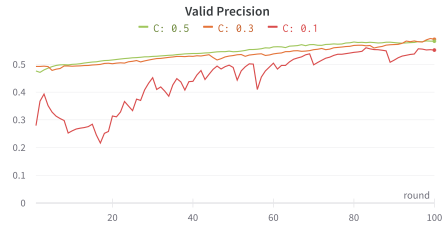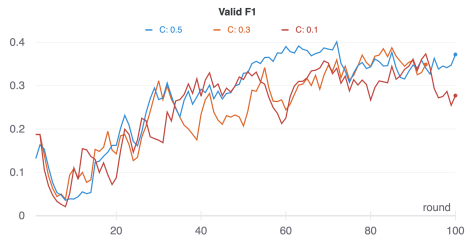
A.4.1 Imbalanced Dataset (IID)
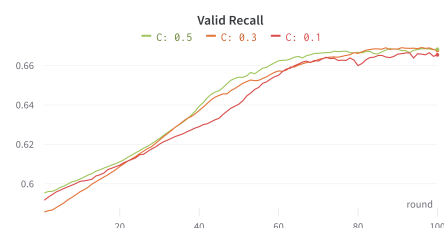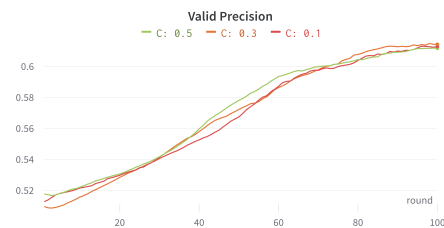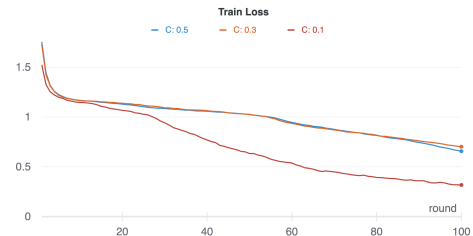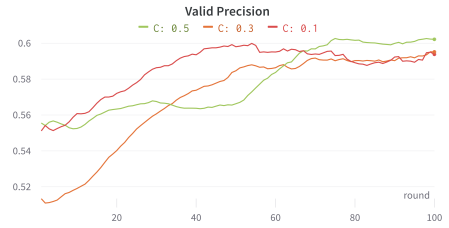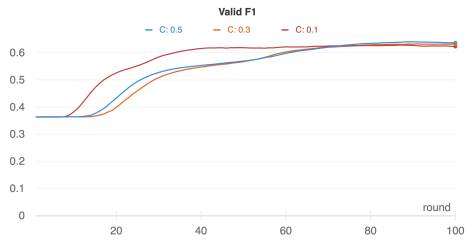
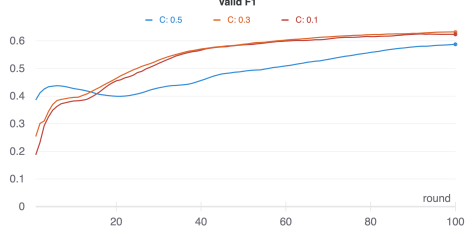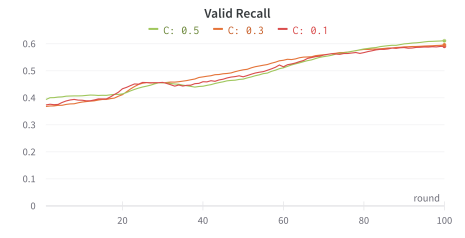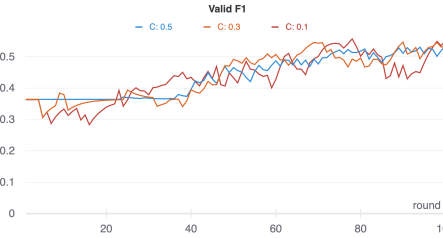## A.4.2 Imbalanced Dataset (non-IID)

A.4.4 Balanced Dataset (non-IID)

A.4.3 Balanced Dataset (IID)

Valid F1

Valid Precision

A.4.5 Cost Sensitive Approach (IID)

Train Loss

Valid Recall

Valid Precision

Valid F1

Valid Recall
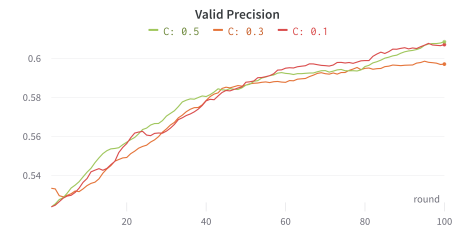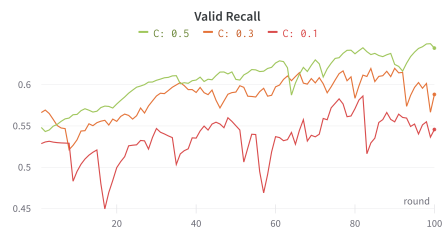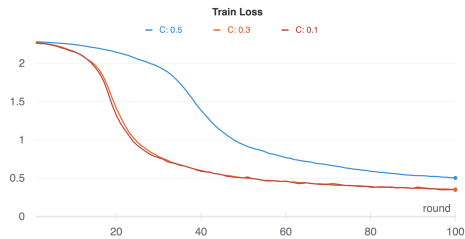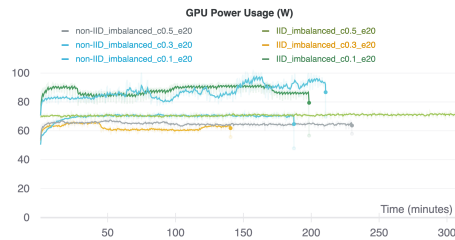
A.5 Time vs GPU Usage

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

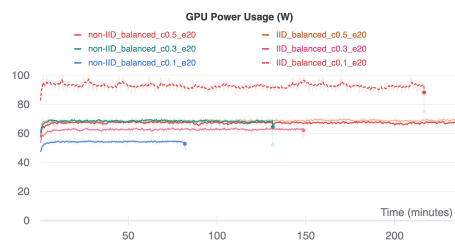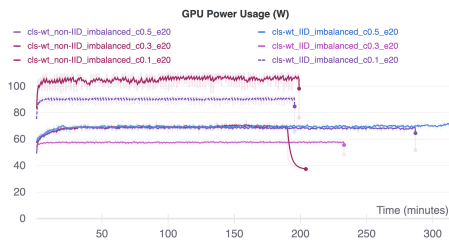### A.5.1 Imbalanced Dataset

GPU Power Usage (W)

Valid F1

A.4.6 Cost Sensitive Approach (non-IID)

A.5.2 Balanced Dataset

Train Loss

GPU Power Usage (W)

A.5.3 Cost Sensitive Approach

A.6 Performance Analysis of CausalFedGSD vs Modified CausalFedGSD

We observe that when we run the original CausalFedGSD and our modification on the same hardware settings with the same number of parameters, the modified version finishes training in just 3.75 hours as opposed to the original CausalFedGSD implementation which takes around 47 hours to finish training. Figure 5 is a run comparison based on the heaviest variant for both algorithms. The highest runtime recorded for both was for the class weight dataset $c = 0.5$ variant. Similar performances are recorded for all other variants on all datasets.
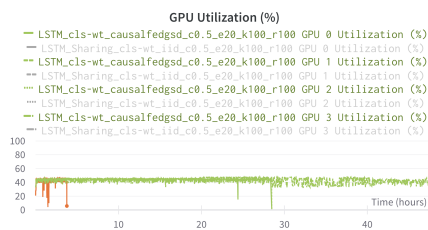

Figure 5: Comparison between optimization times for the baseline CausalFedGSD vs Modified CausalFedGSD