SMPNet: An Algorithmic Framework for Loneliness Detection and Mitigation in Social Media

Venkatesh, Sumukh*
Belin-Blank Center
University of Iowa
Iowa City, United States
sumukh.mails@gmail.com

Yin, Jack*
Belin-Blank Center
University of Iowa
Iowa City, United States
jack.yin.2006@gmail.com

Sng, Grace*
Belin-Blank Center
University of Iowa
Iowa City, United States
gracie.s.sng@gmail.com

Aggarwal, Raghav*
Belin-Blank Center
University of Iowa
Iowa City, United States
raghavaggarwal926123@gmail.com

Fan, Weiguo
Department of Business Analytics
University of Iowa
Iowa City, United States
0000-0003-1272-5538

Huang, Chengyue
Department of Business Analytics
University of Iowa
Iowa City, United States
0000-0001-5256-0120

Tong, Ling
Department of Business Analytics
University of Iowa
Iowa City, United States
ling-tong@uiowa.edu

*These authors contributed equally to the paper

***Abstract*—Loneliness is a growing problem in today's digital age. This study aimed to use NLP models for loneliness detection and prevention on social media platforms. Out of seventy-two combinations of eight models and nine preprocessing methods, SMPNet, made using LSA and MLP, with TFIDF performed best with 85% accuracy. Reddit and Discord bots were then created using SMPNet, able to detect loneliness, offer remedial resources, and alert moderators. The model was retrained on incorrect predictions, continuously improving its accuracy. The success of the model and bots means loneliness detection and prevention are very real and implementable in social media environments.**

I. Introduction

Social media use has exploded over the past few years, with 72% of adults in the United States using some form of it. This is especially true among young people, with 84% of those aged between 18 to 29 using it in 2021 [1].

Social media can greatly benefit people looking to make friends and connect with others, but it can also cause users to feel unwanted and unpopular. Comparing one's lack of likes, comments, and messages to people they follow can affect their perception of their own self-worth. Seeing others happy with friends and family in their posts only amplifies feelings of inadequacy, resulting in loneliness.

Case in point, the growth in social media use has been accompanied by a similar growth in people experiencing loneliness, with consistent findings across multiple studies showing a correlation between social media use with not only loneliness but also further mental and emotional distress [2].

Recently, the growth in social media coupled with the consequences of the recent COVID-19 pandemic have resulted in peak loneliness levels; 36% of Americans suffered from loneliness continuously over a period of 4 weeks, with an astounding 61% of young people suffering [2]. The pandemic and subsequent lockdown resulted in a severe lack of in-person social interaction, and for isolated individuals, social media is the only form of social interaction they can have with others, consequences of which have already been gone over.

The increase in people suffering from loneliness can be seen through the growth in the r/lonely subreddit on social media platform Reddit, which grew from less than 50,000 members to 350,000 post-pandemic. It is a subreddit where users from across the world share their own experiences in regard to loneliness and offer guidance to others. The loneliness suffered by these users and people comes in a wide range, from feelings of not belonging to self-harm to lacking a significant other. The sheer number of posts being sent every day means many of these 'lonely posts' will go unnoticed, likely resulting in even greater feelings of loneliness.

This is where a natural language processing (NLP) model that can detect loneliness comes in. An NLP model would be capable of scanning through thousands of texts in only a few seconds, making them much more efficient at detecting loneliness than humans are. Currently, while there have been models created able to detect loneliness, there aren't any major loneliness prevention mechanisms, such as bots, on social media like Reddit or Discord, which can use the models to detect and help suffering users. Creating a bot with an NLP model could work to detect and alert server and community moderators before symptoms worsen, scanning through multiple servers and communities it is added to. As long as a bot is running, it can work continuously as new posts and messages are sent, making bots extremely effective in social media environments. The lack of these detection bots means the effects of these NLP models are yet to be utilized fully.

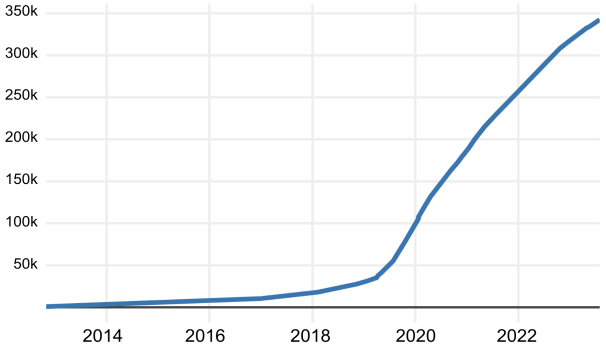

Fig. 1. Growth in subscribers for the subreddit r/lonely sampled from subredditstats.com

II. Background Literature

Using machine and deep learning models for sentiment analysis has been a growing topic of interest for researchers, identifying and preventing certain negative sentiments.

Deep learning models have been used to identify hate speech in tweets, for example. A model that combined LSTM, Random Embedding, and GBDT was able to classify annotated hate speech tweets with a 93.0% F-1 score [3].

Similarly, machine learning models were able to detect feelings of anxiety or depression during the COVID-19 pandemic via posts from Twitter, Facebook, Reddit, Weibo, Instagram, etc. The model was able to perform at an accuracy of over 70% with a precision of 82% [4].

Another study investigated various machine learning, deep learning, and transformer-based models' abilities to classify textual emotion in the Tamil language into various classes, including sadness and joy. The XLM-R model performed the best [5].

For learning models that have been used to predict loneliness, there have been several, including:

1) "Identifying Behavioral Phenotypes of Loneliness and Social Isolation with Passive Sensing: Statistical Analysis, Data Mining and Machine Learning of Smartphone and Fitbit Data" (Doryab et al.): Researchers used machine learning to detect loneliness in college students from UCLA with an accuracy of 80.2% [6].

2) "A Predictive Model for Automatic Detection of Loneliness and Social Isolation using Machine Learning" (Bello-Valle et al.): Researchers used kNN to achieve an accuracy of 80% in detecting loneliness in older adults [7].

3) "Privacy Preserving Loneliness Detection: A Federated Learning Approach" (Pulekar et al.) Researchers achieved an accuracy of 77.5% using XGBoost in a federated learning approach, which preserved privacy [8].

Published models developed so far range in accuracy in the 70s to even the 90s, but given the differing parameters used to define loneliness, any model above 80% is sufficiently accurate to perform detection with incorrectly labeled posts often residing in the gray area of loneliness or no loneliness.

III. Proposed Methodology

For this paper, data was scraped from the subreddit r/lonely. 8000 posts were collected from the Reddit API using Python, with links to the posts and the text in the posts. To annotate the data, we decided to split the dataset into two chunks of 4,000 posts and have teams of two annotate them. Each post was labeled with 0 (indicating a lack of loneliness) or 1 (indicating loneliness). Two different characteristics had to be present in a post to be labeled lonely: a lack of social interaction and resultant negative feelings. Simply being alone did not qualify as loneliness. To ensure inter-rater reliability, each pair also calculated their Kappa score (a metric on a scale from 0.0 to 1.0) and maintained a score of at least 0.7, indicating substantial agreement. After both members of each team annotated the posts, they combined them into a single dataset with posts where the partners had different labels dropped. In our combined dataset, we had 6726 posts, of which 2607 had 'feelLonely' as true, while 4119 had 'feelLonely' as false.

	text	feelLonely	cleaned
0	Just wanna talkIm here to talk if you want	0	wanna talkim talk want
1	Were all lonely people, right?Saw a post on he...	0	lonely people rightsaw post inspired im m22 im...
2	i hate my birthdaymy birthday is in two days a...	1	hate birthdaymy birthday two day nothing even ...
4	No escape no way to get away for awhileNo esca...	1	escape way get away awhileno escape way get aw...
5	A thanks to this community!I just wanted to sa...	0	thanks communityi wanted say thank everyone re...

Fig. 2. Example of text, before and after cleaning; however, uncleaned data was used in the final training of the model

Fig. 3. WordClouds of the most common words for posts that were labeled lonely and not lonely

For each machine learning model, we tested them in accompaniment to different vectorization and tokenization methods as summarized below:

A. TFIDF

TFIDF is a text vectorization technique that represents the importance of each word using TF (frequency of a word in a specific document) versus IDF (rarity across a corpus). Both scores are multiplied to determine the importance of terms [9].

B. CountVectorizer

CountVectorizer is a text vectorization technique that transforms a list of texts into a matrix of token counts, disregarding word order and context [9].

C. Word2Vec

Word2Vec is a word embedding technique that maps words from a vocabulary to dense vectors, capturing semantic relationships and enabling meaningful mathematical operations on word representations [10].

D. XLNet

XLNet is a transformer-based model that incorporates both bidirectional and autoregressive approaches, surpassing BERT's limitations by considering all possible permutations of words and enhancing its ability to understand context [10].

E. Bert

BERT processes text bi-directionally using a Transformer-based architecture, understanding context from both directions (left-to-right and right-to-left) to generate context-aware representations for each word [10].

F. SBERT

SBERT is a sentence embedding technique that transforms sentences into fixed-length vectors, allowing semantic similarity calculations and comparisons [11].

G. RoBERTa

RoBERTa is an optimized variant of BERT created by Facebook that employs a larger training corpus and revised hyperparameters, leading to better performance on various language understanding tasks [10].

H. DistilBERT

DistilBERT is a distilled version of BERT, offering similar performance but with fewer parameters, making it computationally lighter and more suitable for memory-constrained environments [10].

I. GPT-2

GPT-2 is a tokenizer created by OpenAI, which processes input text by breaking it down into smaller units, such as

979-8-3503-0965-2/23/$31.00 ©2023 IEEE

words or subwords, enabling the language model to comprehend and generate coherent sequences [10].

We combined these with the eight different models tested for a total of seventy-two different combinations. Other preprocessing steps, such as lowercasing, spell check, lemmatization, stemming, punctuation removal, and stop word removal, were tried; however, these resulted in lower accuracies for the machine learning models and were not used in our final methodology. This was likely because the steps resulted in a loss of context of certain words and terms, decreasing overall accuracy.

The models used are summarized below:

A. XGBoost

XGBoost is a scalable machine learning system. The model predicts the output using K additive functions, adding the function that most improves the model according to a regularized objective, a combination of a differentiable convex loss function and a term penalizing the model's complexity. The learning process uses gradient tree boosting to find the best split for the tree structure [12].

B. Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning algorithm. The key logic behind SVM involves finding an optimal hyperplane in the feature space for which the margin between the two closest data points (support vectors) from each class is maximized. The SVM algorithm uses the concept of the dual problem and the kernel trick to handle nonlinearly separable data [9].

C. Random Forest

The Random Forest algorithm builds numerous decision trees during training and merges their predictions during testing. It uses bootstrapping to randomly select samples to train individual trees, and at each node, a randomly chosen subset of features splits the data. It makes its final prediction by taking a majority vote for classification from all trees [9].

D. Long Short Term Memory

LSTMs are a type of recurrent neural network designed to avoid vanishing gradient issues. The core of the LSTM is composed of a cell plus an input, output, and forget gate. The three gates regulate information flowing through the memory cell, removing information, updating values, and deciding future hidden states, enabling LSTM to learn and remember long-term dependencies in the sequential data [13].

E. Logistic Regression

Logistic regression is a model that predicts a dichotomous outcome from one or more predictor variables. The central mathematical concept is the logit, which is the natural logarithm of an odds ratio. The logistic model is expressed as $\text{logit}(Y) = \ln(\pi/1\text{-}\pi) = \alpha + \beta X$, where π is the probability of the outcome of interest, α the intercept, β the regression coefficient, and X the predictor variable. The logits of Y and X are linearly related, while between the probability of Y and X is an S-shaped curve. The coefficients are typically estimated using the maximum likelihood method [9].

F. K-Nearest Neighbors

K-Nearest Neighbors (kNN) works by classifying objects based on their 'k' closest training examples. The 'k' in kNN is a user-defined constant with larger k's decreasing variance but also performance. The nearest neighbors are identified using a distance metric, i.e., Euclidean distance with n characteristics with which subjects p and q are compared. The object is then assigned to the class with the most neighbors [9].

G. Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm applying Bayes' theorem with the "naive" assumption that features are independent given class. Given a set of features, the algorithm calculates the probability of each class label, assigning the new instance to the class of the highest probability. Despite the information lost from the "naive" assumption, the classifier is still very effective in decision-making, even if the probabilities are incorrect [9].

H. SMPNet

Latent Semantic Analysis (LSA) helps to address the curse of dimensionality and removes noise from the data, making the representations more compact and interpretable [14].

In code, an MLP classifier is used to perform the final text classification task. An MLP is a type of feedforward neural network where information flows only from input to output through one or more hidden layers of neurons [9].

The MLP classifier is trained on the reduced-dimensional embeddings X_lsa obtained from LSA. The hyperparameters for the MLP, including the number of hidden layers, the number of neurons in each layer (hidden_layer_sizes), and the regularization strength (alpha), are determined through a grid search using cross-validation. The MLP model allows for non-linear mappings and can effectively learn complex patterns and relationships in the data.

The combination of LSA and MLP in SMPNet creates a powerful text classification model that leverages both the semantic meaning captured through LSA and the representation learning capabilities of the MLP neural network. By using the reduced representations from LSA as input to the MLP, the SMPNet is likely to achieve better generalization and improved performance on the text classification task compared to using either technique alone.

SMPNet incorporates data augmentation using SMOTE, a technique that creates synthetic samples by interpolating new instances between existing ones. SMOTE is employed to address class imbalance in the dataset, where one class has significantly fewer instances than others. By generating synthetic samples for the minority class, SMOTE balances the dataset and allows the model to learn from the minority class effectively [15]. This results in improved overall model performance, higher accuracy, and enhanced generalization on unseen data while promoting fairness and equal representation for all classes during training.

After comparing the accuracies of the different models, the best combination of model and vectorizer/tokenizer was used to create social media bots, which could test the model in real and live situations:

A. Bot Implementation on Discord

Combining Python with the Discord Developer, we were able to create a Discord bot. We saved the SMPNet model (the model with the highest accuracy, as we will get into later) using Pickle, which we then uploaded into a Python file. The bot is coded to be able to read and send messages, reading any message of sufficient length (len(message) > 8) and plugging

979-8-3503-0965-2/23/$31.00 ©2023 IEEE

it into the pickled SMPNet model for prediction. Any user message flagged for loneliness prompts a reply from the bot within less than a second. Using the bot, we were able to derive qualitative accuracy in addition to the quantitative accuracy given by the different models, as we could see exactly what types of messages were being flagged or not. The model might be highly accurate in number yet not be able to predict very simple messages like "I am lonely," which would be detrimental to its usage. To combat this, the bot is coded to use incorrectly predicted messages to retrain the model by adding incorrectly predicted statements into its training dataset. As a result, the model is constantly learning from messages inputted into it from Discord, which could be over a hundred in a minute in a hundred different servers.

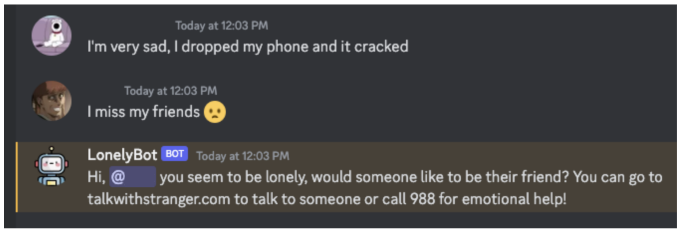

Fig. 4. Demonstration of prototype LonelyBot for Discord

B. Bot Implementation on Reddit

Using Reddit's app preferences combined with Python code using PRAW, the SMPNet model was likewise used to create a Reddit bot. It is very similar to the Discord bot but has a few differences due to the differing platforms. For one, it can access any public servers added to. Second, it can be used to detect loneliness in both posts and comments. However, due to Reddit clamping down on spamming-type bots, it is programmed only to report loneliness in posts every ten seconds, meaning its capabilities in comments are drastically lowered, and its main capabilities will be in detecting loneliness in posts. On a positive note, we were able to use an account already with 514k karma for the bot, which means that it will have access to all large subreddits and also lowers the chances of spam reports. The Reddit bot likewise can use incorrect labelings to retrain itself, as all posts it looks at can be printed for the developer to look over.

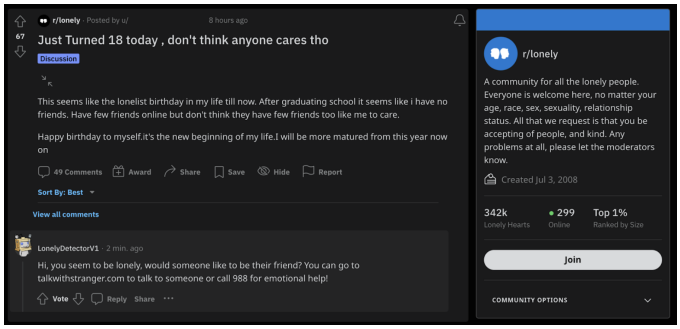

Fig. 5. Demonstration of prototype LonelyDetector for Reddit on a r/lonely subreddit post

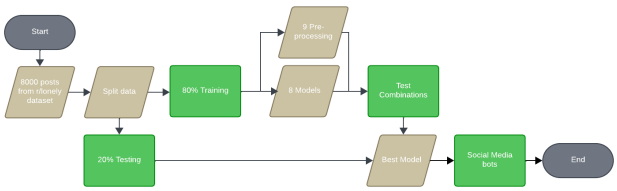

Fig. 6. Diagram of each step of the methodology

IV. Results

We used eight different machine-learning models and tested each model with nine different vectorizers and tokenizers. The results of each of the seventy-two different combinations are shown in the tables below:

TABLE I. Model Accuracies

Tokenizer/Vectorizer	Model							
	SVM	RF	LR	XGB	NB	KNN	LSTM	SMPNet
TFIDF[1]	0.804606	0.768202	0.798663	0.788262	0.754086	0.684993	0.787909	0.854369
Count[1]	0.789747	0.756315	0.774889	0.786033	0.720654	0.710996	0.780971	0.843447
W2V[1]	0.739227	0.744428	0.68425	0.72734	0.638187	0.667162	0.731417	0.827063
XLNet[2]	0.756315	0.775631	0.756315	0.786033	0.625557	0.693908	0.784546	0.790655
BERT[2]	0.78306	0.769688	0.769688	0.768202	0.668648	0.679049	0.796433	0.82585
SBERT[2]	0.81575	0.796433	0.79792	0.808321	0.769688	0.695394	0.772659	0.825849
GPT-2[2]	0.787519	0.763744	0.742942	0.759287	0.695394	0.658247	0.784546	0.811893
RoBERTa[2]	0.76523	0.768202	0.812778	0.786032	0.686478	0.70728	0.818722	0.836165
DistilBERT[2]	0.806835	0.750371	0.80535	0.763744	0.69094	0.690936	0.783061	0.824636

1. Tokenization
2. Vectorization

Fig. 7. Accuracies from our models

TABLE II. Model F-1 Scores

Tokenizer/Vectorizer	Model							
	SVM	RF	LR	XGB	NB	KNN	LSTM	SMPNet
TFIDF[1]	0.72051	0.635514	0.708288	0.707091	0.587796	0.635112	0.722077	0.855247
Count[1]	0.687293	0.637969	0.670294	0.698745	0.689769	0.55441	0.692628	0.83173
W2V[1]	0.688	0.653225	0.620874	0.646776	0.63078	0.619694	0.677765	0.834012
XLNet[2]	0.650442	0.656109	0.662551	0.689655	0.641026	0.638596	0.721689	0.795252
BERT[2]	0.699588	0.615034	0.684318	0.67364	0.583178	0.627586	0.738337	0.826376
SBERT[2]	0.748988	0.686499	0.719008	0.731809	0.721724	0.670947	0.704062	0.826376
GPT-2[2]	0.678652	0.641083	0.602298	0.6625	0.528736	0.54365	0.745167	0.818288
RoBERTa[2]	0.662393	0.650224	0.7375	0.701244	0.666667	0.664395	0.729211	0.837153
DistilBERT[2]	0.733607	0.633188	0.735354	0.670808	0.638889	0.626335	0.705645	0.824954

1. Tokenization
2. Vectorization

Fig. 8. F-1 Scores from our models

Out of the seventy-two models, SMPNet with TFIDF-Vectorization performed the best, with an accuracy of 85.4% and an F-1 score of 85.5%. It also had a precision of 83.4% and a recall of 87.7%. The SMPNet, as a whole, performed the best out of all eight models. Below is its confusion matrix.

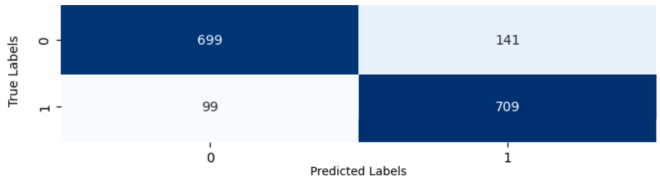

Fig. 9. Confusion matrix for SMPNet, the best-performing model

The model was most accurate in predicting lonely posts at 86% versus 85% for not lonely. Its inaccuracies majorly came from incorrectly predicting lonely for not lonely posts. This is the better of the two possible inaccuracies, as since the model overpredicts lonely posts, it means that it is less likely for an actually lonely post to be predicted not lonely. It is more important for the model to be able to detect loneliness than an

absence of loneliness, so inaccurately detecting loneliness is better than inaccurately detecting no loneliness.

In addition to the quantitative accuracy given by the models, bots coded with the best SMPNet model allowed us to evaluate the effectiveness of the model qualitatively. Incorrectly labeled posts included the model labeling a large majority of posts with the word "friend(s)" lonely, whether or not the post actually exhibited loneliness, as well as those of the form "I was alone, I felt a sad emotion," which should have been labeled lonely but were not. The model was retrained on incorrect posts like these and no longer makes these inaccuracies. As a result, its accuracy in predicting loneliness in social media posts is qualitatively and quantitatively accurate and only improves with time.

The speed of the Discord bot (The Reddit bot is programmed not to respond too fast), which is important for it to run on servers with huge amounts of comments being sent per minute, is a little less than a second. This proves to be too slow to respond to lonely messages before another message is sent in popular servers. However, the actual detection process of the model is extremely fast, performing detection on a thousand posts in five seconds. The time delay of the bot is thus likely from communication between the program and Discord. On a positive note, time is not an issue in alerting moderators, which would be more useful in popular chats where bot messages, and user messages, are easily lost in the multitudes of incoming messages. In slower-moving servers, the bot is effective at responding in time, and its message would likewise be seen by users before being pushed out by new messages. As a result, the Discord bot is effective in both hectic and slower environments. As for the Reddit bot, its capabilities in detecting loneliness in posts are not hindered by time, as in almost all communities, there aren't that many posts being created each minute (There are a few exceptions, including AskReddit), giving the bot ample time to respond. It can comment under one lonely post every ten seconds, which is fast enough to match most subreddit speeds. However, if the bot is added to more than one subreddit, it will be slower, although copies of the bot can be created to bypass this issue.

V. Conclusions

SMPNet with TFIDF-Vectorization was found to be the most accurate model at predicting loneliness in testing posts at 85.4%. Further training the SMPNet on incorrectly predicted posts in its Reddit and Discord bot format led to a higher qualitative as well as quantitative accuracy. It is constantly relearning, and thus its usefulness as a bot in detecting loneliness in social media is only improving. There are so many messages and posts being sent every minute across social media platforms that moderators of communities on these platforms can't keep up. Utilizing the bots coded with SMPNet can prove instrumental in identifying users suffering from loneliness, alerting moderators and other users, offering help resources, and overall preventing the further development of negative emotions and their consequences.

A. Limitations

To further train the SMPNet model, it has to be physically fed an incorrectly labeled post and the correct label. There is no automated method. Beyond this, the SMPNet model is not 100% accurate and does make false predictions, despite its re-training. Limitations for the bots include their speed, which can't match the speed of fast-moving Discord chats and Reddit subreddits, although the bots are programmed in such a way that this won't be a huge issue. (Discord bots report loneliness to moderators, Reddit bots look at posts instead of comments).

B. Future Work

1) X.com: Recently changed to X.com from Twitter, this social media platform likewise supports bots that could be used to identify lonely users.

2) Chrome Extension: Creating a Chrome extension that checks a user's messages as they type them could also help to identify loneliness in platforms that don't support a bot.

3) Image Classifier: The models in this study all ran on text data. Creating an image classifier to predict loneliness from images could transition to identifying loneliness in users from image-dominated social media like Instagram and Snapchat as well.

C. Acknowledgment

We would like to thank the SSTP Program for allowing us to undertake research under Dr. Fan and his research students.

D. Citations

[1] Atske, Sara. "Social Media Use in 2021." Pew Research Center: Internet, Science & Tech, 7 Apr. 2021, www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/.

[2] Bonsaksen, T., Ruffolo, M., Price, D., Leung, J., Thygesen, H., Lamph, G., Kabelenga, I., & Geirdal, A. Ø. (2023). Associations between social media use and loneliness in a cross-national population: do motives for social media use matter?. Health psychology and behavioral medicine, 11(1), 2158089. https://doi.org/10.1080/21642850.2022.2158089

[3] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion, 2017, doi: https://doi.org/10.1145/3041021.3054223.

[4] Ahmed, A., Aziz, S., Toro, C. T., Alzubaidi, M., Irshaidat, S., Serhan, H. A., Abd-Alrazaq, A. A., & Househ, M. (2022). Machine learning models to detect anxiety and depression through social media: A scoping review. Computer methods and programs in biomedicine update, 2, 100066. https://doi.org/10.1016/j.cmpbup.2022.100066

[5] N. Mustakim, R. Rabu, G. Mursalin, E. Hossain, O. Sharif, and M. Hoque, "CUET-NLP@TamilNLP-ACL2022: Multi-Class Textual Emotion Detection from Social Media using Transformer," in Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Jan. 2022. doi: https://doi.org/10.18653/v1/2022.dravidianlangtech-1.31

[6] Doryab, Afsaneh, et al. "Identifying Behavioral Phenotypes of Loneliness and Social Isolation with Passive Sensing: Statistical Analysis, Data Mining and Machine Learning of Smartphone and Fitbit Data." JMIR mHealth and uHealth, vol. 7, no. 7, 2019, https://doi.org/10.2196/13209.

[7] Scott Bello-Valle, Amado, et al. "A Predictive Model for Automatic Detection of Loneliness and Social Isolation Using Machine Learning." Computación y Sistemas, vol. 26, no. 1, 2022, https://doi.org/10.13053/cys-26-1-4157.

[8] M. M. Qirtas, D. Pesch, E. Zafeiridi and E. B. White, "Privacy Preserving Loneliness Detection: A Federated Learning Approach," 2022 IEEE International Conference on Digital Health (ICDH), Barcelona, Spain, 2022, pp. 157-162, doi: 10.1109/ICDH55609.2022.00032.

[9] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[10] A. Gillioz, J. Casas, E. Mugellini and O. A. Khaled, "Overview of the Transformer-based Models for NLP Tasks," 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 2020, pp. 179-183, doi: 10.15439/2020F20.

[11] Reimers, Nils, and Iryna Gurevych. "Sentence-Bert: Sentence Embeddings Using Siamese Bert-Networks." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, https://doi.org/10.18653/v1/d19-1410.

[12] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, pp. 785–794, 2016, doi: https://doi.org/10.1145/2939672.2939785.

[13] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[14] P. Kherwa and P. Bansal, "Latent Semantic Analysis: An Approach to Understand Semantic of Text," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India, 2017, pp. 870-874, doi: 10.1109/CTCEEC.2017.8455018.

[15] Chawla, N. V., et al. "Smote: Synthetic Minority over-Sampling Technique." Journal of Artificial Intelligence Research, vol. 16, 2002, pp. 321–357, https://doi.org/10.1613/jair.953.