Detecting Sensitive Content in Tweets using Hybrid Recurrent Neural Networks

MSc Research Project
Data Analytics

Vishwadeep
Sharma
Student ID: X22125256

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Vishwadeep Sharma
Student ID:	X22125256
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	14/12/2023
Project Title:	Detecting Sensitive Content in Tweets using Hybrid Recurrent Neural Networks
Word Count:	4810
Page Count:	14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Vishwadeep Sharma
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	Q
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	Q
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	Q

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detecting Sensitive Content in Tweets using Hybrid Recurrent Neural Networks

Vishwadeep
Sharma
X22125256

Abstract

This comprehensive report addresses the intricate task of sensitive word detection in tweets originating from Pakistan, employing a meticulously crafted methodology. The research explores stacked LSTM-GRU architectures, individual LSTM, and GRU models, investigating various hyperparameter configurations to discern their impact on the model's performance in sequence modeling tasks. The evaluation of different architectural choices reveals that a model with a larger embedding dimension of 100, coupled with LSTM units of 50 and GRU units of 50, demonstrates promising outcomes, achieving a test accuracy of approximately 56%. By providing detailed examination of sequence modeling methods & possible future study directions, paper provides helpful insights in difficulties & possibilities of identifying sensitive material into social media.

1 Introduction

1.1 Background & Motivation

In this age of instantaneous worldwide connection, social networking sites serve as ever-changing stages upon which people from every walk of life may express themselves. On contrary hand, there's special difficulties into this digital environment when it comes to tweets along with other forms of sensitive material classification. This research uses well constructed technique to investigate complex problem of sensitive word identification into Pakistani tweets. In order to provide solid groundwork for further research, our method starts by cleaning up collection of tweets then filters out language, standardizes text, & tokenizes it. A key component of our approach is dedicated function for identification & classification of politically charged, sexually explicit, religious, personally identifiable terms. Using this classification as springboard, we develop neural network model which employs state-of-the-art methods like text tokenization to extract subtlety & patterns by data. We will try to understand complicated relationship amongst many elements that affect model's ability for recognizing sensitive material by adjusting hyperparameters & experimenting using different model setups. This technique lays groundwork for detailed investigation of mechanics of sensitive word identification with tweets, which will help into building strong models that can handle complexities of online content classification.

Now that almost everyone is connected to internet, there is deluge of user-generated material due to widespread usage of social networking platforms. There are legitimate worries regarding accidental or malicious disclosure of sensitive information, despite fact that this open interchange of information promotes connectedness & cooperation. To tackle difficulties of identifying politically sensitive, religiously sensitive, sexually prioritized, etc., information in tweets, this study suggests novel technique. It focuses on how to effectively identify sensitive content, while our solution leverages deep learning techniques to make informed predictions. Within the spheres of political opinions, religious beliefs, sexual priorities, and other personal information, the potential risks of data breaches and privacy violations are very high. The most important factor in conducting this research is to protect a user's privacy by preventing the risk of spread of sensitive and harmful information which reduces the risk of cyber-bullying, doxxing and online harassment creating a safer online environment. By doing this, these online platforms will be enhancing the platform's reputation and trustworthiness. These platforms need to be following certain data protection regulations imposed by authorities such as GDPR, CCPA and others. This research will be helping these platforms in enforcing their content guidelines and policies more effectively which will reduce the spread of harmful content as well as save them from legal liabilities. By researching in this domain, these platforms can fulfil their ethical responsibilities and provide an enhanced user experience. Then they can also process this data safely to provide more personalized content as well as advertisements to the users while respecting their privacy. Overall, a project focused on sensitive data identification will not only address critical safety and privacy concerns but will also have various business growth opportunities while aligning with the data protection regulations.

This research will be focused on comparing stacked recurrent neural networks such as long short-term memory and gated recurrent unit with their individual approaches which has not been explored in the past and will provide a new avenue for research in the context of sensitive data identification in tweets. In this domain, numerous research papers have been released with the aim of precisely detecting sensitive information within unstructured text. All these papers applied different techniques to achieve their goal of identifying sensitive data such as Term Frequency – Inverse Document Frequency, Autoencoders, Convolutional Neural Networks, Bidirectional Long Short-Term Memory, and numerous other methods. All these research papers discuss different approaches and methods without evaluating a direct comparison within stacked architectures with individual models and focused mainly on their individual merits. This brings a unique perspective to the domain of sensitive data detection in tweets and has the potential to contribute novel insights to this domain.

1.2 Research Question

How accurately we can stack LSTM-GRU architecture for the detection of sensitive data in tweets in comparison to their individual separate models?

1.3 Objective

The goal is to create and assess a model for detecting sensitive words in tweets, utilizing both stacked LSTM-GRU architectures and individual LSTM and GRU models.

1.4 Structure of the Report

The report is structured as follows: Section 2 reviews related work in social media analytics, personal attribute prediction, personalized recommendation systems, disaster-related tweet classification, and document sensitivity classification. Section 3 details the methodology, covering data insertion, analysis, preprocessing, sensitive word detection, and model architecture. Model for layered neural network is described in Sec-4. Section 5 details processes taken during execution, including cleaning and organizing data, defining model, training, & evaluating it. Model's efficacy is assessed in Sec-6 using experiment sets & hyperparameter adjustment. Report is wrapped with plans for future work in Sec-7.

2 Related Work

Finding beneficial findings into user-generated information is of utmost importance in today's digital communication & social networking age. This literature review delves into past studies addressing difficulties and potential benefits of social media analytics, emphasizing topics such as identification of sensitive material, forecasting of individual characteristics, & use of data from social networks in other fields.

Geetha et al. (2020) discussed how people often feel remorse for publishing certain material upon social networking sites & how analysis of social media may help unearth useful information from user-generated content. Finding sensitive data into tweets is where much of their work is concentrated. They began by gathering tweets utilizing Twitter Streaming API with concentration upon 23 unique personal cyber-keywords. These are terms associated with sensitive information such as PII. PISC approach, which stands for Privacy Information Security Classification, uncovers these phrases. Once tweets are collected, they undergo several preprocessing techniques. These treatments involve, but are not limited to, stemming, eliminating unique characters, & deleting stopwords. In order to get tweets ready for analysis, several measures are done. Agencies such as as NIST provide privacy guidelines that are used for annotating tweets. Following text was pre-processed, they used auto-encoders to generate word embedded data. They assess model's accuracy utilizing training & test data utilizing different hyperparameter configurations to see how effectively it's performing. Discussion centers on the methodology's outputs, particularly RNN sensitivity model's predictive power for very sensitive tweets. Utilizing 3 hidden layers using ReLU & softmax activation functions, model is able to get 75% accuracy rate in detecting tweets that include personally identifiable information. Possible future study is also outlined into report, & it involves analyzing both direct & indirect exposures of sensitive data using social media.

Study conducted by Yo & Sasahara (2017) highlights utilization of machine learning algorithms, namely deep learning, to predict personal attributes such as gender, profession, & age groups via analysis of textual content extracted from tweets. Research utilizes word2vec to create word vectors & then converts twitter blocks into vectors, using generated tweet vectors as inputs in model training. Researchers investigate influence of tweet vector dimensions & block sizes upon accuracy of predictions. They find which machine learning methods attain an accuracy of 60-70% when forecasting 3 specific personal qualities of interest. Research emphasizes growing importance of algorithmic social science, emphasizing utilization of social networking sites as an abundant source of data for researching human behavior. This research investigates challenges & opportunities that arise from abundance of data & developments in computer power, specifically combined with machine learning techniques. Study positions itself in wider framework of relevant studies about deducing personal qualities from social information, while highlighting distinctive features of Twitter as a medium for such inquiries. Findings suggest potential avenues for further study, such as enhancing precision of predictions through incorporating other factors beyond textual content.

Khattak et al. (2020) Outlines suggested system architecture which emphasizes developing customized recommendation system. This system is constructed using user profiles generated from data acquired from Twitter. System has 2 primary components that seamlessly function like an plug-in application for Twitter. First module, Data Manager, functions as gateway to Twitter & employs a data fetcher to get streaming data into XML format. Data preparation involves extracting pertinent data from tweets, like user's name, timestamp, textual content, & related metadata. For improving accuracy of data, system utilizes spell checker & slang handler to rectify spelling errors while converting informal expressions in formal equivalents. Second module, Profile Builder, utilizes Alchemy API to extract user interests, emotions, & temporal trends from tweets. Gathered data, enhanced with sentiment analysis, is kept in repository of user profiles for future use. Next modules, Knowledge Extractor & Filter Engine, utilize NLP methods such as POS tagging, dependency interpreting, & sentiment analysis to categorize tweets while offering tailored information to users according to their produced profiles. System's objective is to enhance precision & pertinence by integrating domain-specific seed words, opinion words, & synonym binder. Findings demonstrate notable degree of precision, with maximum of 96% accuracy achieved for tweets connected to diabetes. This highlights system's efficacy in providing tailored data suggestions for assessing sentiment within healthcare field.

Win et al. (2017) examines increasing significance of Twitter, an extensively utilized microblogging platform, into context of natural catastrophes & emergency management. Task highlights difficulty of sorting via extensive volumes of Twitter data produced during emergencies to distinguish trustworthy & enlightening tweets which might assist emergency personnel. Proposed system for monitoring tweets leverages advanced machine learning & NLP methods, notably deploying LibLinear classifier. The main purpose of this system is to classify tweets in 3 categories: "Related & Informative," "Related & Not Informal," & "Not Related". Feature extraction technique uses language characteristics, sentiment lexicon-based features, & disaster lexicon derived from annotated datasets. System's efficacy is assessed utilizing four openly accessible annotated datasets, demonstrating superior accuracy in comparison with different classifiers & techniques of feature extraction. Paper's contributions include creation of corpora linked to catastrophes, introduction of feature extraction approach that may be simply implemented, & expansion of lexicon for natural disasters. Architecture encompasses processes of data collection, preprocessing, extraction of features, building of disaster lexicon, & categorization of tweets. Suggested characteristics were tested via experiments, which showed their efficacy. Algorithm was able to annotate Myanmar earthquake dataset having 75% accuracy rate. Authors propose further research on enhancing disaster lexicons and implementing automated annotation for various sorts of material during catastrophes.

Article by by Patil et al. (2020) addresses growing concern of data privacy in increasingly digital culture, emphasizing responsibility of enterprises to protect personal, non-personal, and sensitive personal data. Given that India is second-largest online market into world, increase in internet use & online transactions leads to significant amount of data, making it necessary to have efficient data categorization. Suggested approach is creation of machine learning tool that can classify data in three categories: non-personal, personal, and sensitive personal information. Objective is to safeguard detected data by means of encryption. Literature review examines several machine learning methodologies for document categorization, emphasizing need of training models with varied datasets to improve accuracy. Promising outcomes for document sensitivity classification have been shown by existing techniques, like deep recurrent neural networks & pre-trained Twitter-based document embedding models. Study concludes by emphasizing data safety and suggesting improvements including enhancing recognition of optical characters and examining variety of document formats.

2.1 Other Related Work

Research by Kolluri et al. (2019) Addresses data management issues in universities, businesses, research organizations, & governments. Statement stresses organized data for efficient analysis and decision-making. language analytics is important because it converts language into numerical data, enabling trend discovery and information extraction. Unsupervised learning's role in clustering data without labels is examined in this study. It also examines semi-supervised training with unlabeled and labeled data. Research covers reinforcement learning and polynomial regression. Talk covers text classification challenges such preprocessing, feature extraction, unbalanced data, and parameter estimation. This suggests that semi-supervised text categorization has grown increasingly important since it effectively controls time-varying expenditures. It additionally stresses need for further research upon simple, efficient data processing, streaming, and parameter optimization techniques.

Article by ElSayed et al. (2019) paper highlights social media unstructured data evaluation issues & importance of data analysis in decision-making. Traditional data analytics methods struggle with unstructured data, producing erroneous results. The literature study covers machine learning, decision trees, linear regression, and sentiment analysis studies upon unstructured text analysis. Commercial decision-making, product assessments, and social media sentiment analysis are covered in these studies. Authors include data input, preprocessing, feature extraction, learning techniques, NLP integration, model construction, and evaluation. Experimental findings show that the proposed framework can accurately categorize text. Text emphasizes suggested framework & recommends subjects for additional research, like extending approach to incorporate larger datasets and measuring computing efficacy.

Silva et al. (2020) The influence of EU's GDPR upon way firms manage personal data is examined as privacy concerns grow across sectors. NLP tools including NLTK, Stanford CoreNLP, and spaCy were tested utilizing generic and PII datasets. The findings show great accuracy in identifying general and context-specific data, including debates on dataset size and model performance. Their technique may be PET, & research covers risks and consequences. Literature study emphasizes importance of NLP its NER in addressing privacy infractions & absence of thorough research on PII, its ramifications, and use cases. The authors stress the applicability of their proposal in privacy-preserving data analysis, showcasing promising results from experiments with different NLP tools and datasets. The paper concludes with reflections on lessons learned, applicability as a PET, and potential risks, providing valuable insights into the intersection of privacy, NLP, and machine learning.

3 Methodology

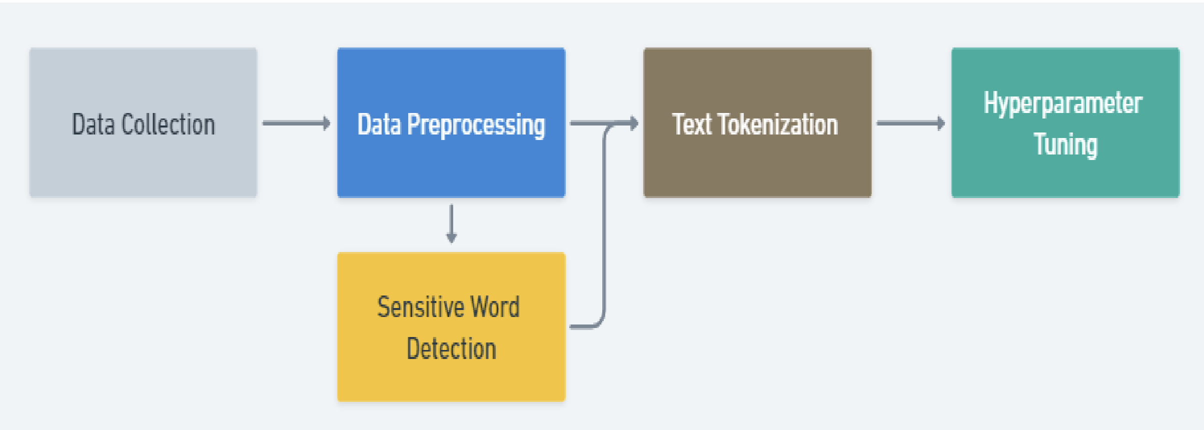

Figure 1

Figure 1 above shows the process followed to prepare the model and achieve our goal of assessing different models of stacked and individual prowess. It starts with the data collection which is done through Kaggle and then we need to perform some preprocessing operations on the data such as dealing with null values, dropping duplicates, dropping unnecessary columns as well as removing stopwords. Then there is a separate step in between which is the definition of Sensitive words dictionary as well as a function that runs and detects the sensitive words. After that the tokenization of the text is performed and further it is converted into sequences to be an input to the model. After all this is done, we prepared six hyperparameter sets of the stacked models and each individual model for LSTM and GRU.

3.1 Data Insertion & Analysis

The dataset comprising 202,202 rows and 7 columns was loaded from the CSV file titled 'Random Tweets from Pakistan- Cleaned- Anonymous.csv'. The columns comprise 'Unnamed: 0', 'created_at_tweet', 'full_text', 'retweet_count', 'favorite_count', 'reply_count', and 'location'. In this analysis, the primary focus lies on the 'full_text' column, which contains the textual content of the tweets. The initial exploration of the dataset reveals that the 'full_text' column has 106,200 non-null entries, with 104,436 unique text values. Remarkably, recurrent tweet seems as "! ! !", occurring 19 times.

For enhancing efficiency of analysis & concentrate just upon textual content, 'full_text' column was extracted by dataset. Early entries of this column exhibit combination of English & Urdu tweets, emoticons, & URLs. Additional preprocessing measures were used to guarantee consistency and pertinence in future studies. Significantly, rows in missing values were eliminated, resulting in refined dataset that is now prepared for further investigation.

An examination of first 10 tweets in 'full_text' column offers comprehensive insight of the varied material within dataset. Tweets cover a wide variety of subjects, emotions, & languages, or are particularly famous for their use of emojis & mentions. Examples of tweets consist of birthday greetings, declarations of affection, allusions to mainstream media & stuff pertaining to gaming. Coexistence of Urdu & English languages in tweets indicates dataset that is multilingual & culturally varied, providing an opportunity for detailed examination of social networking material from Pakistan.

3.2 Data Preprocessing

Tweets undergo series of data pretreatment steps to ensure their suitability for analysis and modeling. Procedure started by first data examination, entailing scrutiny of representative tweet to ascertain condition of data. Afterwards, series of text cleansing & normalization procedures were carried out. The regular expression library was employed for substituting several spaces with single space in order to maintain consistent formatting. Additionally, all text was altered to lowercase. URLs, Twitter identities (handles), hashtags, & text indicating retweets were removed, resulting into more polished dataset. To improve the quality of the data, newline characters were substituted with spaces & duplicated tweets were eliminated.

Tokenization procedure is performed via NLTK's TweetTokenizer, effectively dividing text in separate tokens, facilitating further analysis & comprehension of information. Additional cleansing included excluding emoticons from tweets, since they may be inconsequential for analysis. In order to preserve integrity of dataset & guarantee inclusion of only valuable material, tweets that consisted of less than 3 tokens were recognized & labeled as "NaN" to signify their lack of value. Data has been meticulously pretreated for NLP tasks, making it ready for further advanced analysis and modeling.

3.3 Sensitive Word Detection and Categorization

Creating function, detect_sensitive_words, for recognizing & categorizing sensitive tweet words is crucial. This tool utilizes religious, sexual, political, as well as personal data sensitive phrases from lists. Selection of sensitive keywords for each category depends on your application and definition of sensitive words within those categories. Within religion, terms about divine beings, places of worship, and belief systems may be sensitive. In a sexual environment, allusions to personal relationships, sexuality, and romantic or erotic content may be sensitive. Political speech, government, elections, and democratic processes might be sensitive. Social security numbers, phone numbers, addresses, mail, dates of birth, and financial information are PII. Personal data protection requires protecting this information. Evaluating these phrases in every tweet creates a new column named 'Sensitive_Words' This column's lexicon shows if sensitive terms from various categories are present. This categorization provides a labeled dataset for training & evaluation, enabling model construction.

3.4 Text Tokenization and Model Architecture

In subsequent stage, cleaned tweets are tokenized & model architecture is defined. Keras Tokenizer is used to transform text in sequences of numbers, & padding is utilized to maintain consistent sequence length, improving compliance for model training. Model's architecture is derived upon Sequential model provided by Keras. It comprises Embedding layer for word accountability, then LSTM & GRU layers for capturing sequential dependencies within data. Last Dense layer, using sigmoid activation function, produces classification outcomes for 4 sensitive categories. Model is prepared for training by compiling it with Adam optimizer & categorical cross entropy loss function.

3.5 Hyperparameter Tuning and Experiment Sets

In order to thoroughly investigate influence of different hyperparameter values upon efficiency of model, many sets of experiments are constructed. They consist of many configurations, such as variations into embedding dimensions, LSTM & GRU units, epochs, batches, model complexity, regularization using dropout layers, and tweaks to learning rate. All experiment sets are trained and assessed separately to compare their impact on model efficiency. Trends & patterns in precision and loss are carefully examined to determine the ideal hyperparameter setting for sensitive word detection. Organized experimentation aims to understand model sensitivity to hyperparameter changes & establish ideal configuration for dependable performance.

Essentially, technique entails meticulous data preparation, identifying relevant terms, dividing text into smaller parts, designing model structure, and carefully tweaking model parameters via tests. This approach addresses complexity of recognizing critical phrases in tweets and how hyperparameters affect model performance. Structured experiments help understand work complexities and create and deploy well-informed models in real life.

4 Design Specification

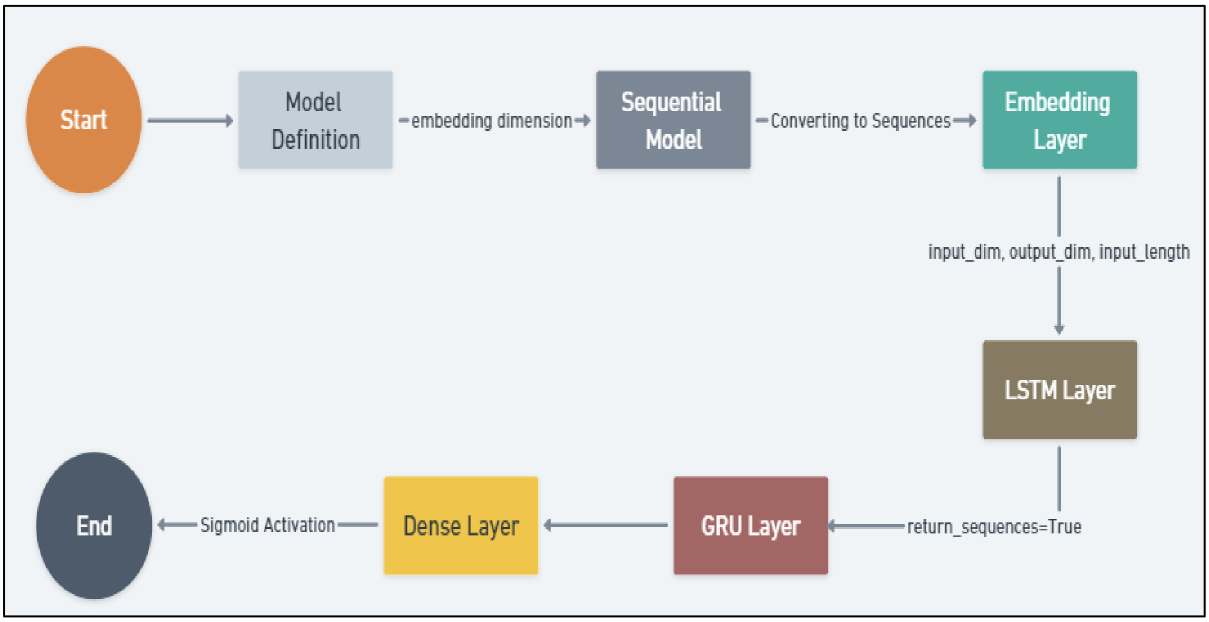

Figure 2

Figure 2 shows a powerful layered neural network model for sequence processing. Procedure begins at model definition step, when embedding dimension is specified. Next, there is Sequential Model component which transforms inputs into sequences which are appropriate for network to handle. An Embedding Layer is implemented to convert input data into compact vectors of a predetermined size. Key factors involved are input dimension, output dimension, & input length.

Subsequently, embedded representation is inputted into an LSTM Layer, which falls under category of recurrent neural networks. This particular neural network excels at capturing long-term dependencies in sequential data. LSTM layer is configured to generate sequences, meaning as output for every single step is retained & passed on to next layer. Output is sent in GRU Layer, which is form of RNN that is simpler than LSTM and is particularly effective at catching patterns of short to medium duration. Finally, network ends having Dense Layer that use sigmoid activation function, which is widely used for jobs that involve binary categorization. Dense Layer consolidates the characteristics learnt by recurrent layers in final output predictions, that are subsequently produced as outcome of model. The use of stacked model architecture enables comprehensive comprehension of sequence data by capitalizing advantages of both LSTM & GRU layers.

5 Implementation

Implementation starts with comprehensive data preparation step. Keras Tokenizer is used to tokenize raw text input, transforming it in number sequences. In order to maintain consistent sequence lengths, we use padding and divide dataset utilizing train test split function by scikit-learn module.

5.1 Model Definition

Sequence modelling architecture consists of an Embedding layer, then stacked LSTM & GRU layers, then concludes having Dense layer for classification. Keras Sequential API streamlines procedure of defining model.

5.2 Model Training

The model is trained on the pre-processed training data, specifying hyperparameters such as the number of epochs and batch size. The training process provides insights into the model's convergence and performance on the training set.

5.3 Model Evaluation

Ultimately, model is assessed upon test set for assessing its ability to generalize. Metrics such as accuracy & loss are calculated to assess effectiveness of model. This section provides concise overview of key steps involved in data preparation, defining architecture of model, training model, and evaluating its efficacy.

6 Evaluation

For this research, we conducted experiments with several hyperparameters for combined LSTM-GRU model, & separate LSTM & GRU architectures. Aim was to determine their impact upon efficiency of text categorization assignment. Hyperparameter sets investigated included changes in embedding dimensions, LSTM & GRU units, dense layer units, number of epochs, & batch size.

6.1 Stacked LSTM-GRU Architectures:

Set 1: Basic Composition (embedding_dim=50, lstm_units=50, gru_units=50, dense_units=4, epochs=10, batch_size=32)

The first model demonstrated a moderate level of accuracy, using test accuracy rate of roughly 6.19%. Although model was simple, it had limited ability to accurately represent complex features of data, resulting in less than ideal performance.

Set 2: Larger Embedding (embedding_dim=100, lstm_units=50, gru_units=50, dense_units=4, epochs=10, batch_size=32)

Test accuracy was substantially enhanced to about 56.62% by increasing embedding dimension to 100. This indicates enhancing semantic representation of words in multi-dimensional embedding space might enhance model's ability to extract significant patterns. Hence, it has the highest accuracy among all the sets.

Set 3: More Epochs (embedding_dim=50, lstm_units=50, gru_units=50, dense_units=4, epochs=20, batch_size=32)

Increasing training time to 20 epochs failed to result in substantial enhancements, suggesting as model may be susceptible to over-fitting or as dataset lacks necessary diversity for prolonged training.

Set 4: Higher Complexity (embedding_dim=75, lstm_units=75, gru_units=75, dense_units=8, epochs=10, batch_size=64)

Increasing the complexity of the model by using higher units and a larger dense layer did not result in substantial improvements. The test accuracy consistently stayed low, hovering around 25.19%, indicating that a more intricate architecture might not be appropriate for the assigned task.

Set 5: Higher GRU Units (embedding_dim=75, lstm_units=50, gru_units=100, dense_units=4, epochs=10, batch_size=32)

A higher number of GRU units did not significantly impact performance, with a test accuracy of approximately 25.24%. This suggests as increased complexity resulting from inclusion of more GRU units didn't have beneficial impact upon model's learning process.

Set 6: Higher LSTM Units (embedding_dim=50, lstm_units=100, gru_units=50, dense_units=4,

epochs=10, batch_size=32)

In same vein, augmenting LSTM units didn't result in significant enhancements, while test accuracy stayed at low level of around 12.37%. Therefore, it is required to do more investigation into different architectural variants in order to choose an appropriate configuration for specified purpose.

6.2 Individual LSTM and GRU Architectures:

Individual LSTM (embedding_dim=50, lstm_units=50, dense_units=4, epochs=10, batch_size=32)

Individual Long Short-Term Memory (LSTM) architecture exhibited subpar performance, achieving a test accuracy of roughly 24.81%. This implies as LSTM by itself may have difficulties in capturing complex patterns inherent in textual input.

Individual GRU (embedding_dim=50, gru_units=50, dense_units=4, epochs=10, batch_size=32)

In same way, individual GRU model showed less than optimum performance, achieving test accuracy of about 6.39%. This suggests as GRU architecture by itself might encounter difficulties in efficiently acquiring & expressing fundamental patterns in dataset.

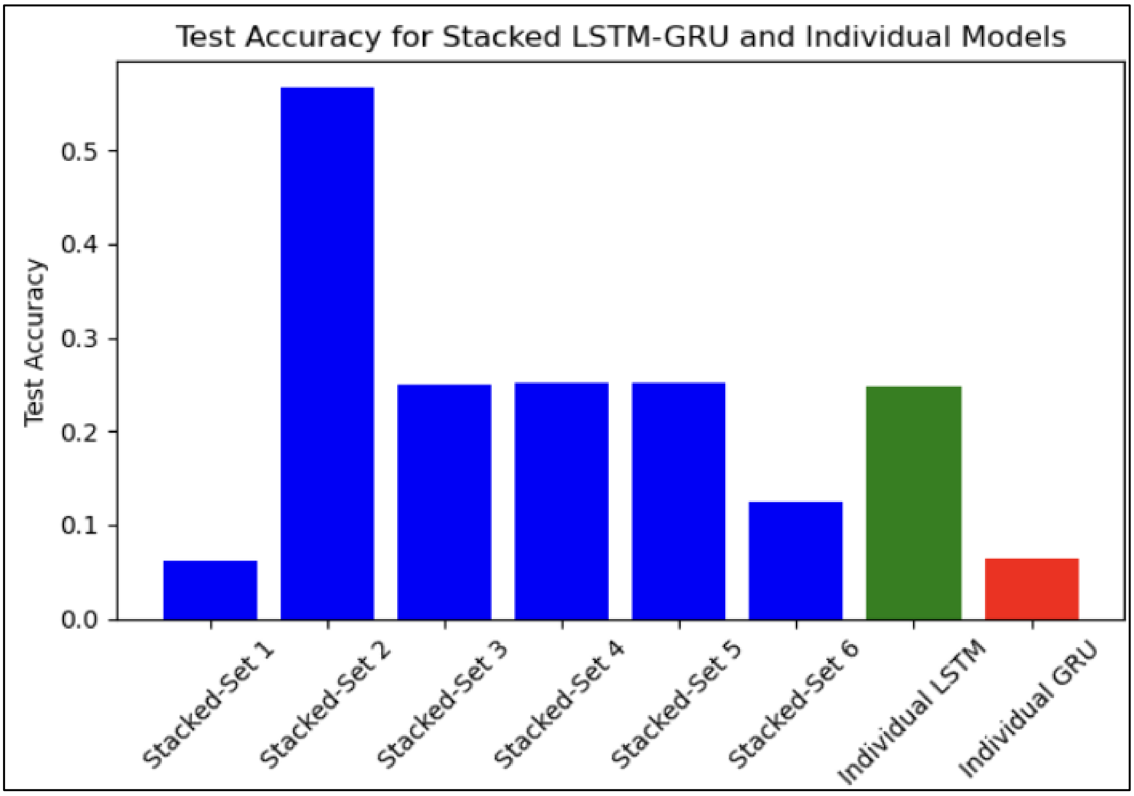

Figure 3

As shown in the above bar graph in Figure 3, it compares the test accuracy of different sets of stacked models with individual LSTM and GRU models. It can be clearly seen that Set 2 has the highest accuracy of them all as it can capture complex dependencies better than individual models. Increasing the embedding dimension helps the model learn the dependencies better than the other hyperparameters for the stacked models.

7 Conclusion and Future Work

Experiments using different hyperparameter settings for stacked LSTM-GRU architectures and individual LSTM & GRU models shown their promise for sequence modeling. Assessment stressed how architectural choices affect model convergence and testing accuracy. Amongst investigated configurations, model that had an embedding size of 100 & LSTM units of 50, together with GRU units of 50, showed promising results. This design attained test accuracy of about 56%, demonstrating enhanced performance in comparison to comparable setups.

Surprisingly, adjusting some parameters, like increasing the number of LSTM as well as GRU units in more sophisticated models, didn't result in corresponding improvements in accuracy. Models with more complexity encountered difficulties in efficient learning, resulting in worse convergence & lower test accuracies. In addition, stacked LSTM-GRU architectures outperformed solo LSTM and GRU models because to their limited performance. Although they were able to catch certain sequential patterns, their whole precision was much poorer in comparison to stacked counterparts.

Additional investigation may further explore hyperparameters by examining a wider spectrum of values for LSTM as well as GRU units, embedding dimensions, and changes in batch sizes & epoch counts. Adjusting these parameters with precision may reveal the most effective configurations that promote improved convergence and heightened model generalization. Exploring attention processes or using more advanced architectures may provide useful insights for improving the models' capacity to grasp complex patterns & long-term relationships in sequences.

These methods have potential to enhance model's understanding of sequential patterns. These methods have capacity to improve model's understanding of sequential patterns. Assessing model's efficacy with varied sequence lengths & types may provide useful insights into its capacity to adapt & generalize across various datasets and applications.

By examining these elements, next studies may enhance the comprehension of stacked LSTM-GRU architectures as well as individual LSTM and GRU models, resulting in more resilient and efficient sequence modeling solutions across many domains.

References

Geetha, R., Karthika, S. and Mohanavalli, S., 2020. Tweet classification using deep learning approach to predict sensitive personal data. In *Advances in Electrical and Computer Technologies: Select Proceedings of ICAECT 2019* (pp. 171-180). Springer Singapore.

Yo, T. and Sasahara, K., 2017, December. Inference of personal attributes from tweets using machine learning. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 3168-3174). IEEE.

Khattak, A.M., Batool, R., Satti, F.A., Hussain, J., Khan, W.A., Khan, A.M. and Hayat, B., 2020. Tweets classification and sentiment analysis for personalized tweets recommendation. *Complexity*, *2020*, pp.1-11.

Win, S.S.M. and Aung, T.N., 2017, May. Target oriented tweets monitoring system during natural disasters. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)* (pp. 143-148). IEEE.

Patil, D., Lokare, R. and Patil, S., 2020, April. Private data classification using deep learning. In *Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST)*.

Kolluri, J., Razia, D.S. and Nayak, S.R., 2019, June. Text classification using machine learning and deep learning models. In *International Conference on Artificial Intelligence in Manufacturing & Renewable Energy (ICAIMRE)*.

Elsayed, M., Abdelwahab, A. and Ahdelkader, H., 2019, December. A proposed framework for improving analysis of big unstructured data in social media. In *2019 14th International conference on computer engineering and systems (ICCES)* (pp. 61-65). IEEE.

Silva, P., Gonçalves, C., Godinho, C., Antunes, N. and Curado, M., 2020, July. Using nlp and machine learning to detect data privacy violations. In *IEEE INFOCOM 2020-IEEE conference on computer communications workshops (INFOCOM WKSHPS)* (pp. 972-977). IEEE.