Synthesized Deep Learning Model in Chatbot Feature Learning

Prince Verma[1], Kiran Jyoti[2], Nirmaljeet Kaur[3], Varsha[4], Shruti[5]

[1] IKG Punjab Technical University, Kapurthala, Punjab, India
[2] Guru Nanak Dev Engineering College, Ludhiana, Punjab, India
[3,4,5] Lovely Professional University, Phagwara, Punjab, India

princeverma.cse@gmail.com[1], kiranjyotibains@gndec.ac.in[2], nirmaljeet.29395@lpu.co.in[3],
varsha.28384@lpu.co.in[4], shruti.24906@lpu.co.in[5]

DOI: 10.48047/ecb/2023.12.si4.1013

Abstract. The gigantic increase in data amounts over the timeline leads to deep learning, as it is a promising approach to resolving the purpose of data processing and management. Due to the limitless scope in all sectors, chatbots have recently been in keen interest and are on the roadmap of improvement. Additionally, deep learning models are nowadays a promising option for feature learning. This paper has evaluated a hybrid deep feature learning model to implement for Chatbot development. The paper evaluates this proposed approach with varying batch sizes for feature learning to measure the impact of batch size on performance. It concluded that the model performance is uplifted for batch size 64 and performs least for batch size 256.

Keywords: Deep Learning, DNN, CNN, RNN, LSTM.

1 INTRODUCTION

A chatbot is an operating system that acts as a virtual assistant medium to produce human-like general communication in informal language with a user to limit data access. Chatbot acts as an economic alternative for businesses that incorporates Natural Language Processing with Artificial Intelligence. Handwritten rules, and/or templates were preliminarily the chatbot armature strategies were substantial leading to unchanged standard chatbot communication technologies. In case there is an analogous pattern between a predefined report of communication within the knowledge base and user exposure, then a response can be provided to the user [36]. The natural capacity to recognize implicit skills, specificity, and associations from traditional chatbots is lacking

in intelligent people with their extended list of names. Information-driven, feedback-driven, output-driven, and program-driven are the four Chatbot categories.

Currently, the feedback-driven chatbot acts as the foundation as the data handed by them is responsible for the expansion of intents & utterances. The type of action taken throughout their strategic approach is essentially what governs the strategy. The dialogue manager requires three phases for Chatbot's motivation. The first is to examine all the response models for feedback generation. The second is to provide a predicated response or comment on priority. Thirdly, the model evaluation policy is utilized to produce a backup response when there is no priority order. The traditional chatbot models are substituted with trainable deep neural chatbots with the influence of deep learning. DL techniques have mostly taken control of conversational modeling. Deep Neural Networks (DNN) infrastructures acclimated from the neural machine translation domain that formerly outperforms extremely well are CNN and RNN.

To execute a chatbot, the machine learning approach can either be generative or retrieval-based model. Retrieval-based models learn to gather a befitting response by use of a predefined reaction repository. The former is more effective in the case of responses as it is Machine Translation based, which will induce new responses from scratch but are probable to fabricate grammatical misapprehensions and challenge a revelatory volume of training data.

1.1 Deep Learning

In 2006, Hinton and Salakhutdinov proposed Deep Learning [46] as a systematical machine literacy model to negotiate unsupervised or supervised point literacy [12] for mining, storing, anatomizing, and illustrating big data accompanies by a thesis of tensor [1]. Tensor Flow is an open-source software library that performs the arithmetical calculations of the expressions to apply deep literacy. The multiple database arrays are depicted as edges in information inflow networks, as well as the actions are denoted as nodes. DL has been forged for numerous NLP issues consequently & is outstanding with forecasting, classification, modeling of findings, and wondrous results in semantics. Also, it determines the task of disdainful training in a non-sequential way for Image recognition which is a vital field nowadays.

Today, a huge quantum of data is eclectic with bottomless ranges and different relative velocities. Yet, the diversity of the data has been a key issue in the defense of massive records [11]. That could happen with the help of perception, which includes all evidence of records valuing both business and technological know-how. Therefore, it has a pivotal perspective on sociable media, healthcare, instructive services, and manufacturing, commerce [1]. Due to the scalability of point learning, deep learning is favorable for the huge volumes of data in point literacy algorithms. Additionally, for effective representations, it can handle the complex correlations of miscellaneous data [1]. The automated feature extraction leads to effort reduction and underneath the big data, volumes reduce relationship complications [2]. Furthermore, energy consumption can be managed by utilizing a small fraction of the neurons. Thus, communication costs can be managed by maintaining connections among the required neurons as per requirement [14]. Based on the kind of information, DL approaches have two categories [11] i.e., Supervised & Unsupervised learning. The use of labeled inputs & outputs to classify information or predict outcomes accurately is the supervised learning used for classification and regression. Whereas unsupervised learning discovers hidden patterns from unlabeled data without human intervention and is used for dimensionality reduction, association, and clustering.

1.2 Unsupervised Deep Learning

SAE (Sparse Auto-encoder) is dependent on the encoder-decoder paradigm [16]. Here, the Encoder predicts the optimal code using simple feed-forward propagation and the Decoder produces good reconstructions using sparse code. In both encoder and decoder, linear filters are scaled automatically as activations i.e., within a layer instead of regulating weights neurons are penalized. In DAE (De-noising Auto-encoders) despite the same training process as the basic autoencoder, there lies the differentiation inside each layer [17]. In, this the input data is slightly corrupted but still uncorrupted data is maintained as the target output. It enforces robustness to partially corrupted inputs by training it to rebuild a clean “repaired” input from a corrupted or partially destroyed input. The reconstruction of the uncorrupted version requires a gradient step. PSC (Predictive Sparse Coding) uses a gradient-based sparsity function that models the interconnection among the neurons by sparsifying their activations [18]. The algorithm is a combination of two convex optimization algorithms. The algorithm learns an active set of coefficients using the feature-sign search method and bifurcates the neurons for any sparsity penalty function using the Lagrange dual method.

A feed-forward NN named MLP (Stacked Multi-Layer Perceptron) has neurons at each hidden layer that carries them [46]. There must be at least one hidden layer. The self-organizing maps and backpropagation algorithms perform the training. The approach requires the hidden layers to be nonlinear while the output layer performs in either a linear or nonlinear fashion. RBM is a NN with two layers where one is hidden and the other is the outer layer [12,15,47]. All the nodes are fully connected between the layers. DBN (Deep Belief Network) has several hidden phases with one input & one output layer [12,47]. In this, pre-training is performed layer-by-layer and the input of one layer comes from the output of the previous phase. The overfitting problem and local optima issues are resolved in this network. Every neuron's weights are first assigned at random.

1.3 Supervised Deep Learning

ANN (Artificial Neural Networks) performs through processing elements consisting of a set of interconnected neurons [55,56]. The network connectivity can either be feedback or feedforward. It acts as a directed graph in which every node conducts a transformation function that receives the weighted inputs for the output [55]. CNN is a feed-forward network that processes multiple array data [20]. The hidden layers are structured in a series of stages for classification and Feature Detection [20,51]. An RNN is a feedback system that processes the input one element at a time [20]. The outputs of the hidden units are trained in discrete time to keep track of all past elements of the sequence [20]. It is beneficial when the input-output relation is required to be known ahead of time [37].

2 DEEP LEARNING AND CHATBOT-RELATED WORK

The implementation of deep learning for the construction of a working Chatbot is not a simplified task. There have been various chatbots developed with ground-breaking achievements. Qingchen Zhang et al. [1] suggested an HBP-based deep computational model for feature learning on heterogeneous data that converges from the vector space to the high-order tensor space. The model is evaluated on the datasets like STL-10, CUAVE, SANE, and INEX. Nevertheless, compared to the stacked autoencoder, this model requires more time to train the variables.

Maryam M Najafabadi et al. [2] have studied Deep Learning in a variety of domains for supervised as well as unsupervised data and their purposes in Big Data Analytics. The issues like complicated designs extraction from massive volumes of information for fast information retrieval can be resolved. The deep learning algorithms deal with issues of big data like high data dimensionality and tagging, scalability of the models, and semantic indexing for information retrieval. Keke Gai et al. [4] proposed SEBSIC design supported by DTRP algorithm. It avoids releasing sensitive information to either financial service providers but requires an additional computation workload. Liwei Kuang et al. [5] proposed a new IHOSVD method for dimensionality reduction based on tensor flow. It can handle the processing of large-scale heterogeneous data created from many sources. The issues encountered in this model involve heterogeneity in data representation and decomposition problems during incremental data.

R. Krishana et al. [7] have proposed Hierarchical Dimension Reduction (HDR) method as a pre-processing level to DNN which improves convergence. The two-step method extracts value from complex and uncertain big data. The model even with reduced model size performs accurate fault predictions. De-graded converge for a shallow network as a consequence. Y. Tamura et al. [8] proposed a model using DNN for a bug-tracking de-vice to learn the fault information. Established an application software for imaging fault data recorded on OSS. The model is cross-platform with an improved reliable operational support system. Ridha Soua et al. [9] have proposed a big-data-based structure for short-term traffic flow estimation involving DBN using heterogeneous input data and Dempster's provisional rule to update the beliefs. It considers event-based evidence and does not compromise integrity.

Xindong Wu et al. [11] have performed a groundbreaking task in the collaboration on the issues for implementation of the mining operations on Big Data and thus there is gigantic data processing in science and engineering domains. There can be complicated issues like the requirement of High-performance computing platforms for handling uncertain values due to autonomous and varied information sources. Xue-Wen Chen et al. [12] have discussed DL implementation to benefit big data with deep structures. The domain adaption can help in training and testing the data samples from different distributions for advancements in medicine, science, and business. It has been noted that the application of DL to the Big Data field is relatively new for improved performance and managing the heterogeneity of data along with the noisy labels. Yisheng Lv et al. [13] introduces a model for a traffic flow estimation method with a DL method that involves Stacked Auto Encoder (SAE) and then trains the deep network in an unsupervised greedy layer-wise fashion. The model outperforms the BP NN, the SVM, and the RBF NN model.

Ilya Sutskever et al. [37] proposed a multi-layered LSTM which uses the WMT-14 dataset. The input sequence mapping with the vector of fixed dimensionality is performed in the first layer of LSTM and the decoding of the target sequence from the vector is done in another LSTM layer. The paper concluded that LSTM outperforms for out-of-vocabulary words, long sentences, and even if the order of the source sentence words is reversed. Yulius Denny Prabowo et al. [40] implement simple RNN and LSTM algorithms and compared them for the chatbot. It uses the Indonesian dataset with customer service talks in several business areas for training. It has been seen that in testing, LSTM responds faster than simple RNN. Additionally, when the training data used is too little, the simple RNN approach is unable to read phrase pairs that often appear and the LSTM approach takes longer than the straightforward RNN. The paper proposed that both LSTM and RNN approaches could be tested utilizing much more information for the learning process.

P. Muangk-ammuen et al. [41] have proposed FAQs Chatbots that respond to customers using RNN and LSTM for text classification. The proposed chatbot uses Thai text questions and provides answers. According to the test results, an LSTM-based chatbot could accurately identify 86.36% of the queries and provide 93.2% accurate responses. In tests, 13.64% of the problems were skipped because the right answers could not be discovered. Manyu Dhyani et al. [44] proposed a chatbot model as Neural Machine Translation (NMT) based on Bidirectional RNN. In this model, two independent RNNs work together for a collaboration of forward and backward information to provide a single output. The model can build the chatbot in any domain-specific chatbot whose Learning rate is low and negligible and whose weights are changed externally while training. The paper concludes that if the model is trained for a limited period on fewer data then there is no necessary change in the learning rate.

3 PROPOSED WORK

In 1966, ELIZA [57] an early chatbot was a physiatrist program that holds conversational interaction with people with a promised automatic translation ability. It uses decomposition rules as input keys and pattern matching with associated reassembly rules for providing responses by involving the knowledge base of 200 categories. Yet, AI experiences its first hype cycle when ELIZA, despite being regarded as a significant turning point in the history of chatbots or AI, is criticized for its improper responses or irrational executive functions.

FFN are universal approximations, i.e., they can solve any complex computations. CNN is an FFN that is quicker, cheaper, and accepts a fixed-size vector as input to produce a fixed-size output [71]. CNN is less efficient to manage temporal data and extended word embeddings which are required for the concatenation of entire sentences. RNN is the deepest, most self-modular, and most powerful of all FNNs and can mix sequential and parallel information processing efficiently [14]. RNN acts more brain-like by allocating the neighboring RNN parts to related behaviors & less related ones to distant RNN parts. Furthermore, it can operate on many input sequences and can handle temporal data by reuse of the previous node's activations [71]. Additionally, it can identify the data without the requirement of any input symbol, by its structure, the form of the text, and its relative symbol but take more time. Furthermore, among RNN algorithms LSTM is preferred over the others for the following reasons:

- LSTM performed well while evaluating long prison sentences since it transfers input of different lengths into a fixed-dimensional vector representation [37].
- LSTM performs even when punished for out-of-vocabulary terms, and improving performance can be achieved by completely flipping the ordering of the words in source sentences (but not in target phrases) [37].
- The LSTM method replies more quickly than the straightforward RNN method since it takes into account word order [40].
- For situations where there is no need to obtain more recent responses, LSTM is a retrieval-based theory [41].
- LSTM performs best on large and complex data sets [14] while in simple RNN there is no significant change in the learning rate [44].

On the contrary, LSTM deals with overfitting which can be dealt with a dropout algorithm. (Dropout approach excludes some input and recurrent connections from activation and can update the weight.)

Thus, as per the above discussion, it has been observed that to enhance the performance of feature learning, an improved or hybrid approach involving both CNN and LSTM could be beneficial.

3.1 Feature Learning through a hybrid method

According to the explanation above, the suggested hybrid model, which combines the CNN & LSTM models, will contain 2 HL. In the feature learning design, firstly certain features are activated via a set of convolutional filters in the Con-volution layer. This layer extracts and recognizes different features regardless of their position accompanied by a max pooling layer for down-sampling. Later, an LSTM later is inserted and the flattening method transforms the results into a 1D array. To avoid overfitting issues, a dropout layer is paired with the convolution and LSTM layers. The model is implemented using Keras, a high-level Python API used on a TensorFlow platform to reduce cognitive load. It offers simple and consistent APIs to minimize the complexity of designing a deep learning model by reducing user actions considerably and providing clear- actionable error messages. It also has developer guides along with extensive documentation [74]. The machine used for implementation has AMD Ryzen 5 5500U Processor with 8GB RAM. The suggested feature learning methodology is put into practice through a two-phase process:

Phase 1: Preparation Phase: does FL and gets the data ready for the model's training.

Phase 2: Implementation Phase: involves designing the model and model training on the pre-processed data.

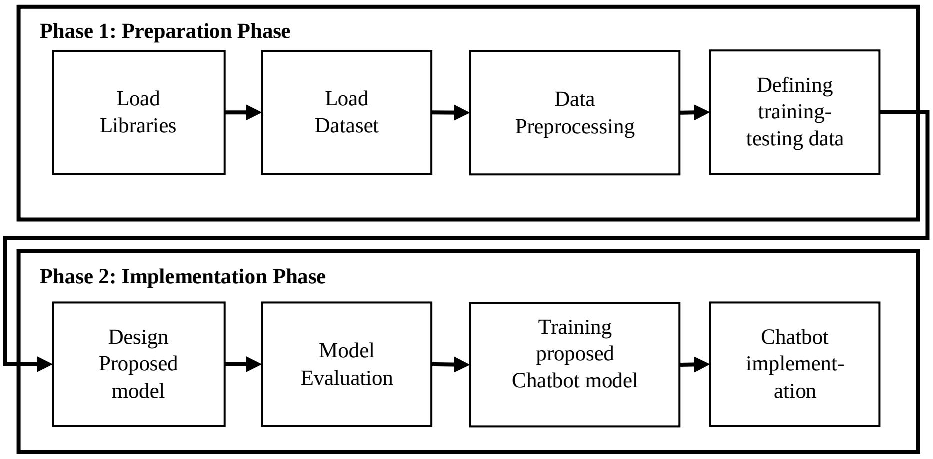

Phase 1: Preparation Phase
Load Libraries
Load Dataset
Data Preprocessing
Defining training-testing data
Phase 2: Implementation Phase
Design Proposed model
Model Evaluation
Training proposed Chatbot model
Chatbot implement-ation

Figure 2: Hybrid Model: Working Feature Learning Model

The above steps are performed in the following sequence:

Step 1: Load Libraries: the various libraries are imported for the execution of code and the functioning of the designed model like pandas, seaborn, time, psutil, keras, sklearn, and other required libraries for neural phases. Additionally, a function for the calculation of memory consumption is defined.

Step 2: Load Dataset: The necessary datasets .csv file is loaded and duplicated into a new file for pertinent attributes.

Step 3: Data Pre-processing: The data is pre-processed as per the process elaborated below.

Step 4: Defining training-testing data: the pre-processed data is split into 80:20 training and testing ratio, which is referred to as the golden ratio. If the training data size is less than 80%, it can lead to under-training, and more than 80% leads to over-training of data.

Step 5: Construct the models according to the specified build: CNN and LSTM layers are used to create the models for each layer. Later, the model is compiled and summarized for training on the dataset.

Step 6: Model Evaluation: analyze the model on various parameters over the various variations of epochs and batch sizes to pinpoint the best case for chatbot training and utilization.

Step 7: Training the chatbot model: The model is trained over a chatbot training set for the best case that emerged in the previous step.

Step 8: Chatbot execution: The Chatbot is taken into execution for the question-answer phase and actual implementation.

Procedure for Pre-processing of data: Pre-processing is to be handled with great precision as it has always been a crucial task in training a model on a dataset. It has always been tailored as per model requirements due to its case-specific approach. It is handled as per the following procedure:

Step 1: From the loaded dataset file in the .csv format, a new database file is created that only has relevant features needed for FE.

Step 2: To avoid overfitting issues, a sample function is used to shuffle all the records in the file.

Step 3: The required computations are performed to overcome the existing null values in the dataset.

Step 4: Only text data is kept in the file only and all other irrelevant data is pruned. i.e., letters except for A-Z and a-z are removed.

Step 5: Lowercase conversion is performed on the data to eradicate the case-sensitive issues. i.e., data is always case-sensitive.

Step 7: The data is further tokenized to reduce the long sentences into manageable words or simpler sentences with a meaning for training the model.

Step 8: By using the stemming strategy, the phrases are broken down into their most basic forms, or stopwords that are frequently used but do not provide new knowledge to the content are eliminated.

3.1.1 Procedure for feature learning model Evaluation

Procedure for feature learning model Evaluation: Later pre-processing, the feature learning model is designed, and implementation is performed in the form of a chatbot. The pre-processed and tokenized data in this phase performs the training using the designed feature learning model. The training loss graph can be constructed that depicts the model's information gain, it is afterward checked for assessment parameters or trustworthiness. The process involved in designing, training, and evaluating the model is as follows:

Step 1: The pre-processed dataset is loaded.

Step 2: The proposed model is designed with the following layers:

a. Convolution layer: Input is passed through a set of CF and ReLU acts as a non-linearity function that adjusts the generated output for more effective training.

b. Max-pooling layer performs partitioning on words and returns the maximum value among the partitions.

c. Dropout layer is utilized to avoid overfitting.

d. LSTM layer will add the feedback benefits to the model.

e. Fully connected Layer outputs the predicted class among k classes.

f. Flattening layer converts the data into a 1-dimensional array.

g. Softmax layer provides the classification output by incorporation of the SoftMax function.

Step 3: The designed model is compiled with accuracy and cross-entropy as training loss.

Step 4: The design is trained with the various number of epochs and batch sizes to measure the effect on space complexity, time complexity, training loss, validation loss, and Accuracy.

3.2 Building the Chatbot

The dataset is trained using the designed model and a chatbot is developed for conversation. The steps are performed in the following sequence:

Step 1: Load Libraries: the various libraries are imported for the execution of code and the functioning of the designed model like pandas, nltk.corpus, psutil, keras, sklearn, and other needed libraries for NL.

Step 2: Data Pre-processing: The data is pre-processed as per the process de-scribed above in section 4.1.1 from the .json file containing tags, patterns, and respective responses.

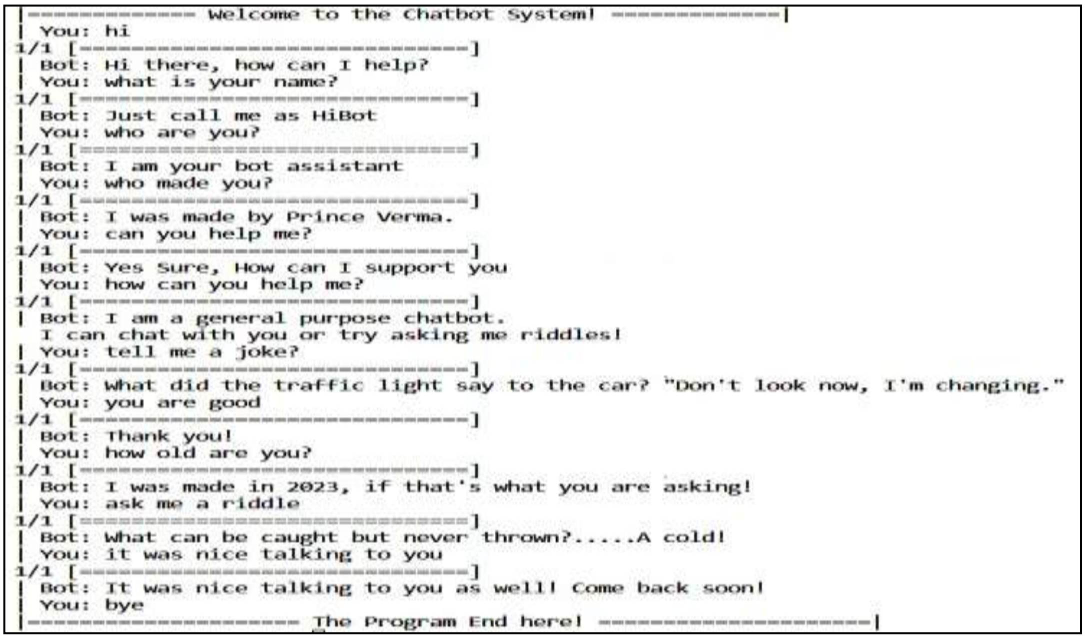
Step 3: Building training data set: Create the pickle files for words and respective classes and updated them. i.e., from the pre-processed file the tokens are saved into a words file and respective tags in a class file.

Step 4: The suggested FL is trained on the aforementioned produced files, and then it is saved. To develop the model layer by layer, CNN & LSTM levels are utilized in sequential techniques.

Step 5: Chatbot execution: The Chatbot is taken into execution for the question-answer phase and actual implementation as follows:

a. Load the classes and words prepared before training
b. Load the saved trained model
c. The input taken from the user is tokenized and cleaned.

d. Use the loaded trained model to give responses to queries asked by the user as per mapping with classes and words provided.



```
|============= Welcome to the Chatbot System! =============|
| You: hi
1/1 [==============================]
| Bot: Hi there, how can I help?
| You: what is your name?
1/1 [==============================]
| Bot: Just call me as HiBot
| You: who are you?
1/1 [==============================]
| Bot: I am your bot assistant
| You: who made you?
1/1 [==============================]
| Bot: I was made by Prince Verma.
| You: can you help me?
1/1 [==============================]
| Bot: Yes Sure, How can I support you
| You: how can you help me?
1/1 [==============================]
| Bot: I am a general purpose chatbot.
  I can chat with you or try asking me riddles!
| You: tell me a joke?
1/1 [==============================]
| Bot: What did the traffic light say to the car? "Don't look now, I'm changing."
| You: you are good
1/1 [==============================]
| Bot: Thank you!
| You: how old are you?
1/1 [==============================]
| Bot: I was made in 2023, if that's what you are asking!
| You: ask me a riddle
1/1 [==============================]
| Bot: What can be caught but never thrown?.....A cold!
| You: it was nice talking to you
1/1 [==============================]
| Bot: It was nice talking to you as well! Come back soon!
| You: bye
|===================== The Program End here! =====================|
```

Figure 3: Screenshot for Chatbot in action

4 FEATURE LEARNING MODEL EVALUATION

The suggested design is evaluated on the News Aggregator database for the best feature learning scenario. The source of the dataset is the UCI Machine Learning Repository for ML algorithms. On February 28, 2016, the AI Lab at the Faculty of Engineering at Roma Tre University, Italy, made the information available online [72,73]. This dataset has news stories for a total of 5 months collected between March 10th, 2014, & August 10th, 2014. There is a total of 422937 news stories constituting the news items with the categories: t- science & approach, b- business, m- health, and e- entertainment. Every news item in the dataset has the following details: article ID, article headline, URL, publisher details, news category, the alphanumeric ID of the news story, hostname, and news publication timestamp.

In the context of sentence translation into another language, the performance is counted in the Bleu score which should be as high. As in this scenario, no translation is required, leading to the least significant Bleu Score. So, the parameters taken into consideration are the training time, memory, accuracy, and training rate. For training, the pre-processed data is split into training data & test data in an 80:20 proportion. The models are contrasted on parameters for 32, 64, 128, and 256 batch sizes for 25 epochs to evaluate the best case. Then the best case is taken to develop a chatbot.

4.1 Evaluation of Proposed Hybrid Model in Terms of training time

It is the total time consumed for training the model. The effect of various variations in batch sizes and the number of epochs on training time is shown in Table 1. From Figure 4, the model performs best for Batch size= 256 and worst in the case of Batch size=128. It can also be observed that there is a positive linear growth of training time in accordance with the improvement in the set of epochs. Additionally, the slope of the graph gets steeper with the decrease in batch size.

It is the total time consumed for training the model. The effect of various variations in batch sizes and the number of epochs on training time is shown in Table 1. From Figure 4, the model performs best for Batch size= 256 and worst in the case of Batch size=128. It can also be observed that there is a positive linear growth of training time in accordance with the improvement in the set of epochs. Additionally, the slope of the graph gets steeper with the decrease in batch size.

Table 1: Comparison of Training Time of a hybrid model for variation in epochs and Batch size

Training Time (in Seconds)		Number of Epochs					
		1	5	10	15	20	25
Batch Size	32	1447.517	4912.784	9244.369	13575.95	17907.53	22239.12
	64	1282.077	4589.235	8516.382	12443.53	16370.67	20132.38
	128	1488.679	5328.77	9888.76	14448.76	19008.75	23376.65
	256	1265.211	3309.225	5864.244	8419.262	10974.28	13529.29

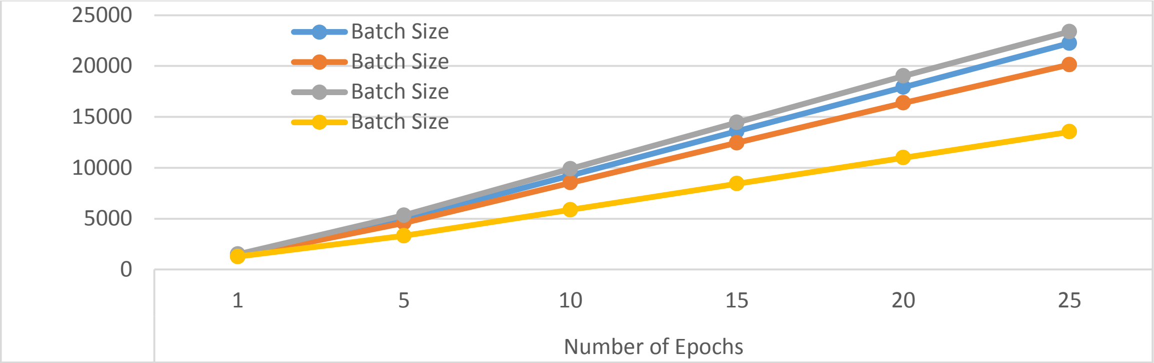


Figure 4: Comparison of Training Time of Hybrid model for variation in epochs and Batch size

4.2 Evaluation of Hybrid Model in Terms of Memory

It is the total memory consumed by the model for training and providing the answer to the given query. To study the effect of various batch sizes on the model batch sizes 32, 64, 128, and 256 are used, and for the study of the effect of epochs, the memory usage is calculated for 1, 5, 10, 20, and 25. The effect on memory consumption is shown in Table 2 for variations on batch and epochs. From Figure 5, it is observed that memory usage is the least in all cases of variant epochs for the batch size of 256 and highest for a batch size of 64 or 32. It could also be seen that the set of epochs and memory usage are unrelated but remains in a specific range. Moreover, storage use drops as the batch size rises.

Table 2. Comparison of Memory usage of for variation in epochs and Batch size

Memory (in bytes)		Number of Epochs					
		1	5	1	15	1	25
Batch Size	32	338334976	338337792	338326208	338334976	338329600	338261664
	64	338349878	338370560	338296832	338309120	338331124	338305792
	128	338118616	338162160	338118616	338142880	338131706	338174928
	256	337953472	337953760	337940400	337976640	338052288	338074064

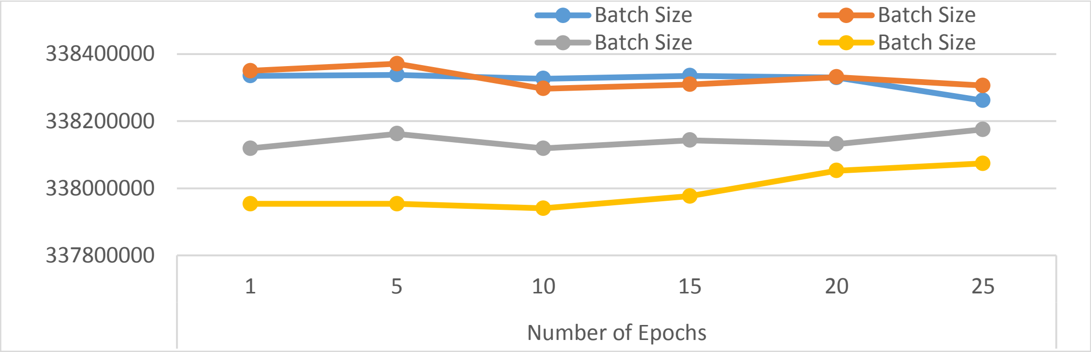


Figure 5: Comparison of Memory Usage of Hybrid model for variation in epochs and Batch size (in Bytes)

4.3 Evaluation of Hybrid Model in Terms of Accuracy

The Hybrid model is tested for various variations in batch sizes. During training, each epoch returns the accuracy till the epoch is 25 (Table 3). It could be seen from Figure 6 that there is a higher slope for batch size 256 in comparison to others. After 6 epochs, for batch size 32 the accuracy decreases in comparison to all the other batch sizes. The accuracy with the batch size of 64 outperforms all others in comparison and it is least in the case of a batch size of 32. Moreover, the accuracy of the design enhances with the number of epochs subsequently.

Table 3. Evaluation of Accuracy for Changes in Batch Size & Epochs (in%age)

Accuracy (in %)		Number of Epochs							
		1	2	3	4	5	6	7	8
Batch Size	32	86.64	90.04	91.11	91.79	92.38	92.76	93.03	93.29
	64	86.37	90.31	91.45	92.19	92.77	93.13	93.43	93.71
	128	85.51	90.1	91.25	91.97	92.55	93.03	93.35	93.64
	256	83.89	89.84	90.91	91.72	92.31	92.77	93.18	93.51

Accuracy (in %)		Number of Epochs							
		9	10	11	12	13	14	15	16
Batch Size	32	93.57	93.76	93.97	94.13	94.27	94.39	94.5	94.65
	64	93.99	94.21	94.39	94.55	94.7	94.86	94.99	95.14
	128	93.92	94.16	94.29	94.45	94.61	94.8	94.93	95.06
	256	93.81	94.04	94.22	94.38	94.53	94.69	94.82	94.97

Accuracy (in %)		Number of Epochs								
		17	18	19	20	21	22	23	24	25
Batch Size	32	94.7	94.82	94.88	95	94.97	95.13	95.2	95.26	95.35
	64	95.24	95.36	95.42	95.51	95.5	95.65	95.7	95.77	95.83
	128	95.2	95.31	95.36	95.4	95.42	95.55	95.57	95.63	95.65
	256	95.07	95.19	95.25	95.34	95.33	95.48	95.53	95.6	95.66

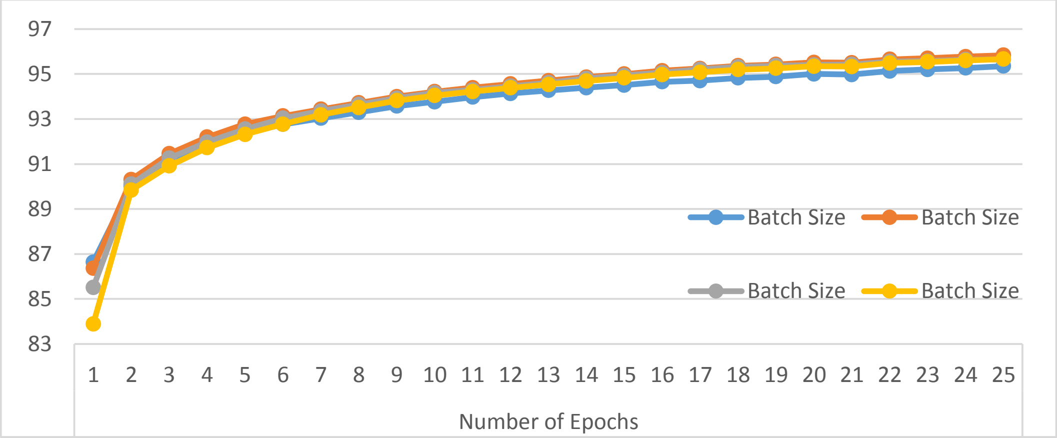


Figure 6: Comparison of Accuracy of Hybrid model for variation in epochs and Batch size (in %age)

5 CONCLUSION & FUTURE SCOPE

From the above results, it is clear that the batch size and the set of epochs have a direct effect on the model performance. It has been observed that by reducing the batch size the accuracy increases but at the cost of memory consumption. It has also been seen that at batch size 256 the training time and memory consumption is minimum at the stake of accuracy which is non-negotiable. It is observed that with an increase in batch size over 64, the accuracy reduces for the defined dataset which leads to a conclusion that the hybrid design conducts better with moderate batch size. In the future, the Hybrid model could be optimized with other combinations of deep learning layers to get more accuracy in the smaller batch size as concluded in this paper.

REFERENCES

[1] Zhang, Q., Yang, L. T., and Chen, Z.: Deep Computation Model for Unsupervised Feature Learning on Big Data. IEEE Transactions on Services Computing, 9(1), 161-171 (2016)

[2] Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald R., and Muharemagic, E.: Deep learning applications and challenges in big data analytics, Journal of Big Data, Springer, 2(1), 1-21 (2015).

[3] Li, Y, Gai, K., Ming, Z., Zhao, H., and Qiu, M.: Intercrossed Access Controls for Secure Financial Services on Multimedia Big Data in Cloud Systems, ACM Transactions on Multimedia Computing, Communications, and Applications, 12(4), 1–18 (2016).

[4] Gai, K., Qiu, M., and Elnagdy, S. A., Security-Aware Information Classifications Using Supervised Learning for Cloud-Based Cyber Risk Management in Financial Big Data. IEEE International Conference on Big Data Security on Cloud (BigDataSecurity), High Performance and Smart Computing (HPSC) and Intelligent Data and Security (IDS) 2016, IEEE, New York, NY, USA, 197-202 (2016).

[5] Kuang, L., Hao, F., Yang, L. T., Lin, M., Luo, C. and Min, G.: A Tensor-Based Approach for Big Data Representation and Dimensionality Reduction. IEEE Transactions on Emerging Topics in Computing, 2(3), 280-291 (2014).

[6] Zhu, L., He, Y., Yu, F. R., Ning B., Tang, T., and Zhao, N.: Communication Based Train Control System Performance Optimization Using Deep Reinforcement Learning. IEEE Transactions on Vehicular Technology, 66(12), 10705-10717 (2017.

[7] Krishnan, R. S., Jagannathan, S., and Samaranayake, V. A.: Deep learning inspired prognostics scheme for applications generating big data. International Joint Conference on Neural Networks, IEEE, Anchorage, AK, USA, 3296-3302 (2017).

[8] Tamura Y., Ashida, S., and Yamada, S.: Fault Identification Tool Based on Deep Learning for Fault Big Data. International Conference on Information Science and Security (ICISS) 2016, IEEE, Pattaya, Thailand, 1-4 (2016).

[9] Soua, R., Koesdwiady, A., and Karray, F.: Big-data-generated traffic flow prediction using deep learning and dempster-Shafer theory. International Joint Conference on Neural Networks (IJCNN), IEEE, Vancouver, BC, Canada, 3195-3202 (2016).

[10] Naidu, A., Deshmukh, A. Y., and Bhure, V.: Design of high throughput and area-efficient advanced encryption system core. International Conference on Communication and Signal Processing 2014, IEEE, Melmaruvathur, India, 1974-1978 (2014).

[11] Wu, X., Zhu, X., Wu, G. Q., and Ding, W.: Data mining with big data. IEEE Transactions on Knowledge and Data Engineering, 26(1), 97-107 (2014).

[12] Chen, X. and Lin, X.: Big Data Deep Learning: Challenges and Perspectives. In: IEEE Access, vol 2, 514-525 (2014).

[13] Lv, Y., Duan, Y., Kang, W., Li, Z., and Wang, F.: Traffic Flow Prediction with Big Data: A Deep Learning Approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), 865-873 (2015).

[14] Schmidhuber, J.: Deep learning in neural networks: An overview. Journal of Neural Networks, Elsevier, 61, 85–117(2015).

[15] Hinton, G. E., and Salakhutdinov, R. R.: Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507 (2006).

[16] Ranzato, M.A., Boureau, Y., and LeCun., Y.: Sparse feature learning for deep belief networks. 20th Proceedings of the International Conference on Neural Information Processing Systems (NIPS'07), ACM, 1185-1192 (2007).

[17] Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. 25th Proceedings of the International Conference on Machine learning (ICML '08), ACM, 1096-1103 (2008)

[18] Lee, H., Battle A., Raina, R., and Ng, A. Y.:

Efficient sparse coding algorithms. in Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, MIT Press, IEEE, 801-808 (2007).

[19] Simonyan K., and Zisserman: A Very deep convolutional networks for large-scale image recognition. Proceedings of the International Conference on Learning Representations, 1-14 (2014).

[20] LeCun, Y., Bengio Y., and Hinton, G.: Deep learning. Nature, vol 521, 436–444 (2015).

[21] Srivastava N. and Salakhutdinov, R.: Multimodal learning with deep Boltzmann machines. The Journal of Machine Learning Research, 15(1), 2949-2980 (2014).

[22] Rifai, S., Mesnil, G., Vincent, P., Muller, X., Bengio, Y., and Dauphin, Y.: Higher-order contractive auto-encoder. Machine Learning and Knowledge Discovery in DB, Springer, 6912, 645-660 (2011).

[23] B. Schölkopf, B., Platt, J., Hofmann T.: Efficient sparse coding algorithm. Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, MIT Press, 801-808 (2007).

[24] Coates, A., Ng, A. Y. and Lee, H.:An analysis of single-layer networks in unsupervised feature learning. 14th Proceedings of International Conference on AI and Statistics, in Proceedings of Machine Learning Research, vol 15, 215-223 (2011).

[25] Lin, Y., Shen, S., Liu, Z., Luan, H., and Sun, M.: Neural relation extraction with selective attention over instances. 54th Proceedings of the Annual Meeting of the Association for Computational Linguistics, vol 1, 2124–2133 (2016).

[26] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J: Distributed representations of words and phrases and their compositionality. 26th Proceedings of International Conference on Neural Information Processing Systems, ACM, 2, 3111–3119 (2013).

[27] Jiang, X., Wang, Q., Li, P., and Wang B.: Relation extraction with multi-instance multilabel convolutional neural networks. 26th Proceedings of International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 1471–1480 (2016).

[28] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research, 11(1), 3371-3408 (2010).

[29] Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. 28th Proceedings of International Conference on Machine Learning, 833-840 (2011).

[30] Hinton, G., Osindero, S., and Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation, ACM, 18(7), 1527-1554 (2006).

[31] Zeng, W., Lin, Z., Liu, Z., and Sun, M.: Incorporating relation paths in neural relation extraction. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 2017.

[32] Wenpeng, Y., Katharina, K., Mo Y., and Hinrich, S.: Comparative Study of CNN and RNN for Natural Language Processing. Proceedings in IBM Research, USA, 2017.

[33] Kumar, S.: A Survey of Deep Learning Methods for Relation Extraction. arXiv preprint arXiv:1705.03645, 2017.

[34] Understanding Convolutional Neural Networks for NLP. http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/#more-348., last accessed 2023/03/01.

[35] Stemler, S.E.: Content Analysis. Emerging Trends in the Social and Behavioral Sciences, John Wiley & Sons Inc, 1-14 (2015).

[36] Cleverbot, http://www.cleverbot.com/, last accessed 2022/12/31.

[37] Sutskever, I. Vinyals, O., and Le, Q.V.: Sequence to sequence learning with neural networks. 27th Proceedings of the International Conference on Neural Information Processing Systems(NIPS'14), Vol 2. MIT Press, ACM, 3104–3112 (2014).

[38] Mauli, M., Galley, M., Quirk, C., and Zweig G.: Joint language and translation modeling with recurrent neural networks. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1044–1054 (2013).

[39] D. Bahdanau, K. Cho, and Y. Bengio..: Neural machine translation by jointly learning to align and translate. Proceedings of the 3rd International Conference on Learning

Representations, (2015).

[40] Denny Prabowo, Y., Warners, H.L.H.S., Budiharto, W., Kistijantoro, A.I., Heryadi, Y., and Lukas.: LSTM and Simple RNN comparison in the problem of sequence to sequence on conversation data using Bahasa Indonesia. Proceedings of the Indonesian Association for Pattern Recognition International Conference (INAPR), IEEE, 51-56 (2018).

[41] Muangkammuen, P., Intiruk, N., and Saikaew, K.R.: Automated Thai-FAQ Chatbot using RNN-LSTM. 22nd Proceedings of the International Computer Science and Engineering Conference (ICSEC), IEEE, 1-4 (2018).

[42] Su, M.H., Wu, C.H., Huang, K.Y., Hong, Q.B., and Wang, H.M.: A chatbot using LSTM-based multi-layer embedding for elderly care. Proceedings of the International Conference on Orange Technologies (ICOT), 70-74 (2017).

[43] Sherstinsky, A.: Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Journal of Physica D: Nonlinear Phenomena. Special Issue on Machine Learning and Dynamical Systems, Elsevier, 404, 1-28 (2020).

[44] Dhyani, M., andKumar, R.: An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. Proceedings of Materials Today, Elsevier, 34(3), 817-824 (2021).

[45] Huang, D-H., and Chueh, H-E.: Chatbot usage intention analysis: Veterinary consultation, Journal of Innovation & Knowledge. 6(3), 135-144, (2021).

[46] Karhunen, J., Raiko, T., & Cho K.: Unsupervised deep learning: A short review. Advances in Independent Component Analysis and Learning Machines, Elsevier, 125–142 (2015).

[47] Hua, Y., Guo, J., and Zhao, H.: Deep Belief Networks and deep learning. Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things, IEEE, 1-4 (2015).

[48] Kim, B., Kim, J., Chae, H., Yoon, D., and Choi, J. W.: Deep neural network-based automatic modulation classification technique. International Conference on Information and Communication Technology Convergence (ICTC), IEEE, 579-582 (2016).

[49] Zhao, G., Zhang, G., Ge, Q., and Liu X.: Research advances in fault diagnosis and prognostic based on deep learning. Prognostics and System Health Management Conference (PHM-Chengdu), IEEE, 1-6 (2016).

[50] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324 (1998).

[51] Introducing Deep Learning with MATLAB. Source: https://in.mathworks.com/campaigns/offers/next/deep-learning-ebook.html, last accessed 2023/03/01.

[52] Albawi, S., Mohammed T. A., and Al-Zawi, S.: Understanding of a convolutional neural network. International Conference on Engineering and Technology (ICET), Antalya, Turkey, IEEE, 1-6 (2017).

[53] Palasundram, K., Sharef, N. M., Nasharuddin, N. A., Kasmiran, K. A., and Azman, A.: Sequence to Sequence Model Performance for Education Chatbot. International Journal of Emerging Technologies in Learning, vol. 14(24) No. 24, 56-67, 2019.

[54] Dey, R., and Salem.: Gate-variants of Gated Recurrent Unit (GRU) neural networks," 60th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 1597-1600 (2017).

[55] Yao, X.: Evolving Artificial Neural Networks: Proceedings of the IEEE, 87(9), 1423-1447 (1999).

[56] Sharma, V., Rai, S., and Dev, A.: A Comprehensive Study of Artificial Neural Networks. International Journal of Advanced Research in Computer Science and Software Engineering, 2(10), 2012.

[57] Weizenbaum, J.: ELIZA: A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(6), 36-45 (1966).

[58] Masche J., and Le, N-T.: A Review of Technologies for Conversational Systems. Advanced Computational Methods for Knowledge Engineering. Advances in Intelligent Systems and Computing book series, AISC, vol. 629, Springer International Publishing, 5th Proceedings of the

International Conference on Computer Science, Applied Mathematics and Applications (ICCSAMA), 212-225 (2017).

[59] Wallace R.S.: The Anatomy of A.L.I.C.E, in Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer. Epstein, R., Roberts, G., and Beber G., Springer Netherlands: Dordrecht. 181-210 (2009).

[60] Cleverbot, Source: https://www.cleverbot.com/. last accessed 2023/03/01.

[61] Robert W.G.: Teaching to the Turing Test with Cleverbot. Transformations: The Journal of Inclusive Scholarship and Pedagogy, JSTOR, 24(1-2), 56–66 (2014).

[62] Weizenbaum, J.: A response to Donald Michie. International Journal of Man-Machine Studies. 9(4), 503-505 (1977).

[63] Shawar, B.A. and Atwell E.: A comparison between Alice and Elizabeth chatbot systems" The University of Leeds, School of Computing research report, 2002.

[64] Lenchner, J.: Knowing what it knows: selected nuances of Watson’s strategy. in IBM Research News 2011, IBM.

[65] Wilcox, B.: ChatScript Basic User’s Manual. Source: http://chatscript.sourceforge.net/Documentation/ChatScript%20Basic%20User%20Manual.pdf, last accessed 2023/03/01.

[66] Google. Dialog flow 2010 23/04/2018]; Source: https://dialogflow.com/.

[67] Worswick, S.: Mitsuku Chatbot: Mitsuku now available to talk on Kik messenger. 2010 last accessed 2023/03/01.

[68] Janarthanam S.: Hands-On Chatbots and Conversational UI Development: Build chatbots and voice user interfaces with Chatfuel, Dialog flow, Microsoft Bot Framework, Twilio, Alexa Skills. Packt Publishing, 2017.

[69] Microsoft Cognitive Services: LUIS. Source: https://learn.microsoft.com/en-us/azure/cognitive-services/luis/, last accessed 2023/03/01.

[70] Amazon Web Services, Amazon Lex – Build Conversation Bots, Source: https://docs.aws.amazon.com/lex/latest/dg/what-is.html, last accessed 2023/03/01.

[71] Zhang, R., Yuan, Z., and Shao, X.: A New Combined CNN-RNN Model for Sector Stock Price Analysis. 42nd IEEE International Conference on Computer Software & Applications, 546-551 (2018).

[72] News Aggregator Dataset. Source:https://www.kaggle.com/datasets/uciml/news-aggregator-dataset, last accessed 2023/03/01.

[73] News Aggregator Dataset. Source:http://archive.ics.uci.edu/ml/datasets/News+Aggregator, last accessed 2023/03/01.

[74] Keras. Source: https://keras.io/, last accessed 2023/03/01.

[75] Verma, P., Jyoti, K.: Chatbots Employing Deep Learning for Big Data. International Journal of Innovative Technology and Exploring Engineering (IJITEE). 8(11), 1005-1010 (2019).

[76] Verma, P., Jyoti, K., Sahni, V. et al.: Synthesized feature learning model on news aggregator for a chatbot. Soft Computing, (2023). https://doi.org/10.1007/s00500-023-08069-0