iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VII **Month of publication:** July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36336

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

An Improved Sentiment Extraction Model for Social Media Contents using spaCy Based Deep Neural Networks

Thivaharan. S
Asst.Prof (Sel. Grade), CSE, PSG Institute of Technology and Applied Research, Coimbatore

Abstract: ***Modern communication devices generate huge amount of data through the manifold usage of various social media applications. Among the entire generated data more than 40% are unstructured in nature. The industry also is reluctant to retain the data with the following characteristics: data containing asynchronous time stamp, replicated data, data which are broken while transmission and data that leads to misclassification. It is high time to drop-out the irrelevant data and considering the synchronous ones. In this article, a sentiment extraction model is proposed that governs the various social media contents. SpaCy is used as the preferred implementation language as it has many readily available libraries for the purpose of content classification. To avoid the over-fitting problem the actuators like "relu" and "sigmoid" are used. Even though many such classifiers are available for content classification, this article with the appropriate setting of Epoch counts, a categorical accuracy of 68% is obtained. The entire model is implemented in the TensorFlow based platform.***
Keywords: Corpus, Data analytics life cycle, Kaggle, Tuples, Vectors, Keras, RELU, Sigmoid.

I. INTRODUCTION

Sentiment analysis has applications ranging from social media content mitigation to product suggestion. It helps the producers to plan and target the specific category to maximize the sales target and promotional activity. Now-a-days almost all major industry incorporations have a high relying factor spent for the sentiment extraction and mitigation.

A. Spacy Library And Packages

"SpaCy" is a python library which is available as an open source package. It has many inherent modules specific for natural language processing applications. "SpaCy" [1] has the capability to process large volumes of text irrespective of structured or un-structured [2]. These libraries can not only process the text under consideration but based on the importing of functions, can "understand" the meaningful portions out of the volumes of text corpus [3]. Though "SpaCy" has no in-built functionalities to pre-process the dataset, it is good at effectively extracting meaningful tags as "Named Entities" [4]. Majority of these applications are implemented using the deep learning algorithms [5].

In this article, the "Kaggle" [6] dataset from https://www.kaggle.com/c/social-sentiment-extraction/information" is used for analysis. The article proposes a approach using which the sentiments of the social media content is classified as one in "positive", "negative" and "neutral".

The Figure 1, below shows the view of train dataset having the following title tags: textID, text, Selected_text, sentiment. Based on the sentiment classification in the train dataset, the test dataset is correspondingly configured, such that accurate prediction is attained.

	A	B	C	D
1	textID	text	selected_text	sentiment
2	cb774db0d1	I`d have responded, if I were going	I`d have responded, if I were going	neutral
3	549e992a42	Sooo SAD I will miss you here in San Diego!	Sooo SAD	negative
4	088c60f138	my boss is bullying me...	bullying me	negative
5	9642c003ef	what interview! leave me alone	leave me alone	negative
6	358bd9e861	Sons of ****, why couldn`t they put them c	Sons of ****,	negative
7	28b57f3990	http://www.dotheboucy.com/smf - some	http://www.dotheboucy.com/smf - some sh	neutral
8	6e0c6d75b1	2am feedings for the baby are fun when he	fun	positive
9	50e14c0bb8	Soooo high	Soooo high	neutral
10	e050245fbd	Both of you	Both of you	neutral
11	fc2cbefa9d	Journey!? Wow... u just became cooler. he	Wow... u just became cooler.	positive

Figure 1: Sample Kaggle dataset with the tags.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

The following figure 2 shows the view of test dataset. The title tag under investigation in the test dataset is "text", which is refined and classified based on the "SpaCy" categorizers.

	A	B	C
1	textID	text	sentiment
2	f87dea47db	Last session of the day http://twitpic.com/67ezh	neutral
3	96d74cb729	Shanghai is also really exciting (precisely -- skyscrapers galore). Good tweeps in	positive
4	eee518ae67	Recession hit Veronique Branquinho, she has to quit her company, such a shame	negative
5	01082688c6	happy bday!	positive
6	33987a8ee5	http://twitpic.com/4w75p - I like it!!	positive
7	726e501993	that`s great!! weee!! visitors!	positive
8	261932614e	I THINK EVERYONE HATES ME ON HERE lol	negative
9	afa11da83f	soooooo wish i could, but im in school and myspace is completely blocked	negative
10	e64208b4ef	and within a short time of the last clue all of them	neutral

Figure 2: Sample view of test dataset

B. Pre-Processing The Datasets

The following code snippet removes the punctuation marks and any other white spaces [7]. A white space is a set of blank spaces that are redundant than the needed single unit of blank space. The emoticons are in the Unicode range u"\U0001F600-\U0001F64F". Pictographs are in the Unicode [8] range u"\U0001F300-\U0001F5FF". The transport and other map symbols are in the Unicode range u"\U0001F680-\U0001F6FF". These Unicode ranges are compiled using the python regular expression package "re" [9]. # is a single line comment indicator in python.

```
def remove_white_space(text):
    pattern = re.compile("["
                u"\U0001F600-\U0001F64F"  # emoticons
                u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                u"\U0001F680-\U0001F6FF"  # transport & map symbols
                u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                "]+", flags=re.UNICODE)
return pattern.sub(r'', text)
```

The following code snippet remove any occurrence of hyperlinks and reference-jumps through uniform resource locations (url). The pattern "http[s]://[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|[0-9a-fA-F]" is used which matches all the sentences that start with either "http" or "https" followed by one or more occurrences of digits, alphabets and special symbols.

```
def remove_hyperlink(text):
pattern  = re.compile('http[s]?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|[0-9a-fA-F]))+')
return pattern.sub(r'', text)
```

C. Cleaning The Intermediary Dataset

After the removal of white spaces and hyperlinks, from the intermediary pruned dataset is cleaned using the following code snippet. Here a dictionary called "del_dict" is used, where a dictionary needs to comply with the "key: value" pair format [10]. It is initialized to hold the screening value pairs as all the punctuation marks from the "string.punctuation" [11] property. A list is constructed with name "tbl1" to hold all the transformed dataset records. The final processed dataset is all transferred to lower case to minimize any problem that could arise because of case sensitive markers in the regular expression.

```
def clean_text(text ):
    del_dict = {char: '' char in string.punctuation}
    del_dict[' '] = ' '
    tbl1 = str.maketrans(del_dict)
    txt1 = text.translate(tbl1)
    txt2 = ' '.join([w for w in txt1 if ( not w.isdigit() and  ( not w.isdigit() and len(w)>3))])
    return txt2.lower()
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

II. CREATION OF TRAIN AND TEST DATASET

The cleaned dataset is converted to the "SpaCy" format. This is needed to apply the spaCy libraries. The spaCy format has to be in the following pattern.

[{'text', {'cats': {'Class_Label1': 1, 'Class_Label12': 1, ...'Class_Labeln': 0},
{'text', {'cats': {'Class_Label1': 0, 'Class_Label12': 1, ...'Class_Labeln': 0}, { ...}]

The following code snippet is applied over the dataset and produces the dataset in the above format.

```
train_data,train_text,train_grp  = load_data_spacy("\\TweetSenitment\\train.csv")
test_data,test_text,test_grp  = load_data_spacy("\\TweetSenitment\\test.csv")
print(training_data[:10])
```

The outcome of the above code snippet is captured in the following Figure 3. From the figure it is clearly seen that the outcome of intermediary processed dataset complied with the above mentioned spaCy format.

```
Name: sentiment, dtype: int64
[('have responded were going', {'cats': {'positive': 0, 'negative': 0, 'neutral': 1}}), ('sooo will miss here diego', {'cats':
{'positive': 0, 'negative': 1, 'neutral': 0}}), ('boss bullying', {'cats': {'positive': 0, 'negative': 1, 'neutral': 0}}), ('wh
at interview leave alone', {'cats': {'positive': 0, 'negative': 1, 'neutral': 0}}), ('sons couldnt they them releases already b
ought', {'cats': {'positive': 0, 'negative': 1, 'neutral': 0}}), ('some shameless plugging best rangers forum earth', {'cats':
{'positive': 0, 'negative': 0, 'neutral': 1}}), ('feedings baby when smiles coos', {'cats': {'positive': 1, 'negative': 0, 'neu
tral': 0}}), ('both', {'cats': {'positive': 0, 'negative': 0, 'neutral': 1}}), ('journey just became cooler hehe that possibl
e', {'cats': {'positive': 1, 'negative': 0, 'neutral': 0}}), ('much love hopeful reckon chances minimal never gonna cake stuf
f', {'cats': {'positive': 0, 'negative': 0, 'neutral': 1}})]
```

Figure 3: Intermediary processed dataset

A. Text Categorization Model For Metrics Evaluation

This entire text categorization model creation is done using "sci-kit" [12] learn libraries which are to be imported as part of the classification implementation. The following code snippet does the role of text categorization. This method has four parameters namely tokenizer list, preprocessed dataset, test data test group. The package util.tokenizer creates a tuple with name src based on the tuple [13] comprehension "(util.tokenizer(text) for text in test_text)". Once tuple "src" is created, the same is iterated and appended to the text categorizers in a sorted fashion.

```
def grp_evaluate(util.tokenizer, textgrp, test_text, test_grp ):
   src = (util.tokenizer(text) for text in test_text)
   for i, src in enumerate(textgrp.pipe(src)):
      metrics = Sort(src.cats.items())
      for i in metrics:
         txtList.append(i[0])
```

B. Regularizers In Sentiment Classification Model

For the purpose of sentiment model creation, the existing "en_core_web_de" [14] is used, which is a medium sized named entity recognizer. The existing reference, as it has the already POS (Parts-Of-Speech) tagged, has lesser necessity for parser. Thus re-training is not needed. In this model "ngram" vectors are used, where an "ngram" is a parameterized nth dimensional array or array of arrays. As spaCy shows no difficulty while processing the varying length records in a dataset, this proposed model does not have any record length restrictions. All the "ngram" vectors are primed using the drop-out regularizers namely "relu" [15] and "sigmoid" [16]. These are applied in succession and forbids the over fitting chance that could arise while classification is in progress.

C. Training The Model

The following code snippet show the stages involved in model training. This function accepts seven parameters namely the dataset, iteration count, test data, test categories, the metrics model from the previous step, drop-out rate and finally the necessity for any token to vector conversion. In this case the last parameter is set to the default value of none. The primitive basic evaluation model ("en_core_web_md") is loaded by the spaCy. Then the add_pipe method improves the NLP pattern. After the NLP processor pattern is populated dynamically, the same is applied to the dataset iteratively.

This iteration should equal to the Epoch counts, such that misclassifications are avoided. To ensure that the final train dataset is unbiased, the processing and record selection is happening in a shuffled order.

```
def train_Dataset(  dataset, itr,test_text,test_grp, model_arch, dropout = 0.3, tok2vec=None):
    nlpPattern = spacy.load("en_core_web_md")
    nlpPattern.add_pipe(textgrp, last=True)
    pipe_EX = ["textcat", "trf_wordpiecer", "trf_tok2vec"]
    with nlpPattern.disable_pipes(*other_pipes):
        opt = nlpPattern.begin_training()
        for i in range(itr):
            random.shuffle(train_data)
            batSize = minibatch(train_data, size=batSize)
            for txt in batSize:
                texts, annotations = zip(*txt)
                nlpPattern.update(texts, annotations, optimizer=”sgd”, drop=dropout)
 return nlpPattern
```

The outcome of the training model and the accuracy of the dataset classification are monitored is shown in the Figure 4(a) and 4(b) of first and last respectively.

Iteration: 0

	precision	recall	f1-score	support
positive	0.78	0.60	0.68	1075
negative	0.75	0.39	0.51	983
neutral	0.54	0.82	0.65	1376
accuracy			0.63	3434
macro avg	0.69	0.60	0.61	3434
weighted avg	0.68	0.63	0.62	3434

Figure 4(a): Training model accuracy (Iteration – 0)

Iteration: 9

	precision	recall	f1-score	support
positive	0.78	0.68	0.72	1075
negative	0.68	0.57	0.62	983
neutral	0.59	0.73	0.65	1376
accuracy			0.67	3434
macro avg	0.69	0.66	0.67	3434
weighted avg	0.68	0.67	0.67	3434

Figure 4(b): Training model accuracy (Iteration – 9)

It is clearly seen from the outcome that, an overall precision of 69% is attained through the model, where 78% accuracy in predicting “positive” sentiments, 68% accuracy in predicting the “negative” sentiments and 59% accuracy in predicting the “neutral” sentiments is experienced.

III. ANALYZING THE SPACY MODEL

For model training a parameter called “Epochs” [17] is used and defined by giving the value which is the cycle / loop count value. In this study, after many trials and the value of 15 is assigned to the parameter “Epochs”.

Outcome_Model_stage1 = model.fit(Train_DS.Shuffle(2000), batch(128), Epochs = 15)

The code has given the provision for shuffling with the value 2000. The batch size shuffled units is 128. The iteration wise outcome is needed for reference, so the verbose property is set to 1. The outcome of the fitting is iterated for 15 times and the history data is collected. The loss function starts with the value of 1.0498 and at the end of 15th Epoch the loss is decreased by 07536. But it is also noted that in the validation lose the values are not following any pattern (Neither ascending nor descending). Pictorially this data is plotted between the epoch count and the loss accuracy over the training data. The figure 5 shows the plot.

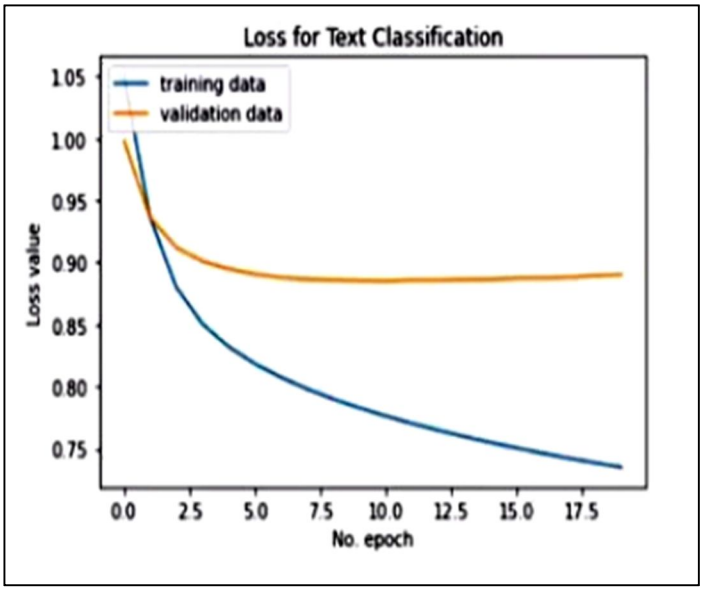


Figure 5. Loss for text classification between Epoch Vs. Loss accuracy (Over training data)

IV. TECHNIQUES TO OPTIMIZE THE ACCURACY

The technique deployed for optimizing the accuracy are “drop-out” [18] and “Kernel regularizers” [19]. Drop-out is a deep neural network version which randomly sets few of the misleading records to zero. Kernel regularizers on the other hand, assign the penalty on the layers ensuring the smooth functioning of layer activities. These penalty parameters are also optimized at the later stage by the same kernel regularizers in the form of housekeeping activity [20]. The code for the above modification is shown below:

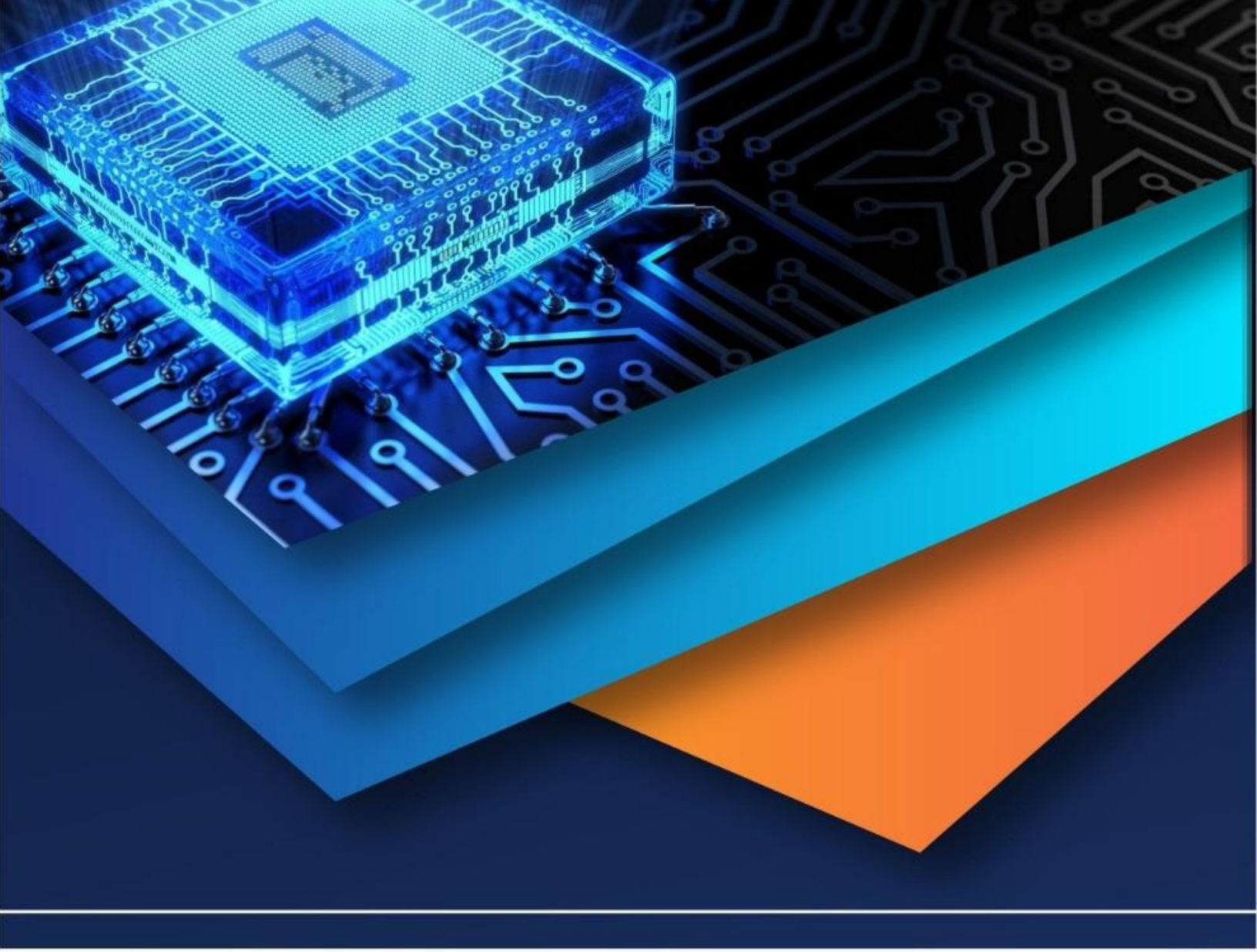
```
NN_Model1 = TF.Keras. Sequential()
Model.add(NN_Model1)
NN_model1.add(TF.Keras.layers.Dense(10, ”relu”, Kernel_regularizer= regularizer.L1))
NN_model1.add(TF.Keras.layers.Dropout(0.5))
NN_model1.add(TF.Keras.layers.Dense(10, ”relu”, Kernel_regularizer= regularizer.L2))
```

V. CONCLUSION

In this article a tensor flow framework based neural network model is deployed. “Relu” and “Sigmoid” activation layers are used to mitigate the over-fitting problem, which will eventually result in degradation in the prediction accuracy. As an improvised attribute, the “drop-out” optimizers and “Kernel regularizers” are used. The model is trained with 15 Epochs, such that optimized prediction accuracy is attained. The study captured the accuracy outcomes of training loss and test loss based on the Kaggle dataset. Epoch count of 15, which is fixed based on trial-and-error, iteration the new model is allowed to train. This enhanced the prediction accuracy to the extent of 25% more compared to the earlier neural network layers. The future challenge would be to test the model with dynamism of contents.

REFERENCES

[1] Di Gangi, Mattia A., et al. "MuST-C: a multilingual speech translation corpus." 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2019.
[2] Yadav, Nikhil, Omkar Kudale, Aditi Rao, Srishti Gupta, and Ajitkumar Shitole. "Twitter Sentiment Analysis Using Supervised Machine Learning." In Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020, pp. 631-642. Springer Singapore, 2021.
[3] Sungheetha, Akey, and Rajesh Sharma. "A Comparative Machine Learning Study on IT Sector Edge Nearer to Working From Home (WFH) Contract Category for Improving Productivity." Journal of Artificial Intelligence 2, no. 04 (2020): 217-225.
[4] Smilkov, Daniel, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang et al. "Tensorflow. js: Machine learning for the web and beyond." arXiv preprint arXiv:1901.05350 (2019).
[5] Manoharan, J. Samuel. "Capsule Network Algorithm for Performance Optimization of Text Classification." Journal of Soft Computing Paradigm (JSCP) 3, no. 01 (2021): 1-9
[6] Chakrabarty, Navoneel, and Sanket Biswas. "Navo Minority Over-sampling Technique (NMOTe): A Consistent Performance Booster on Imbalanced Datasets." Journal of Electronics 2, no. 02 (2020): 96-136.
[7] S. Thivaharan., G. Srivatsun. and S. Sarathambekai., "A Survey on Python Libraries Used for Social Media Content Scraping," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp. 361-366, doi: 10.1109/ICOSEC49089.2020.9215357.
[8] Haoxiang, Wang, and S. Smys. "Big Data Analysis and Perturbation using Data Mining Algorithm." Journal of Soft Computing Paradigm (JSCP) 3, no. 01 (2021): 19-28.
[9] Rodman, Emma. "A Timely Intervention: Tracking the Changing Meanings of Political Concepts with Word Vectors." Political Analysis 28.1 (2020): 87-111.
[10] Hettikankanama, H. K. S. K., Shanmuganathan Vasanthapriyan, and Kapila T. Rathnayake. "Machine Learning-Based Approach for Opinion Mining and Sentiment Polarity Estimation." In Inventive Computation and Information Technologies, pp. 601-613. Springer, Singapore, 2021.
[11] Babu, P. Raghavendra, S. Sreenivas, U. S. VinayVarma, and N. Neelima. "A Hybrid Approach to Review Mining—Restaurant Data in Depth Analysis." In Innovative Data Communication Technologies and Application, pp. 831-841. Springer, Singapore, 2021.
[12] Sivaganesan D, A Hybrid Architecture Combining Artificial Intelligence and Block chain for IoT Applications, J. Sustain Wireless system, Vol. 2, No. 3, pp. 138-142, 2020,
[13] Zou, Difan, et al. "Gradient descent optimizes over-parameterized deep ReLU networks." Machine Learning 109.3 (2020): 467-492.
[14] Goel, Priyanka, and S. Sivaprasad Kumar. "Certain class of starlike functions associated with modified sigmoid function." Bulletin of the Malaysian Mathematical Sciences Society 43, no. 1 (2020): 957-991.
[15] Chakraborty, Rupak, Rama Sushil, and M. L. Garg. "An improved PSO-based multilevel image segmentation technique using minimum cross-entropy thresholding." Arabian Journal for Science and Engineering 44.4 (2019): 3005-3020.
[16] Hamdan, Yasir Babiker. "Faultless Decision Making for False Information in Online: A Systematic Approach." Journal of Soft Computing Paradigm (JSCP) 2, no. 04 (2020): 226-235.
[17] Siddique, Fathma, Shadman Sakib, and Md Abu Bakr Siddique. "Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers." 2019 5th International Conference on Advances in Electrical Engineering (ICAEE). IEEE, 2019.
[18] Moreno-Marcos, Pedro Manuel, et al. "Temporal analysis for dropout prediction using self-regulated learning strategies in self-paced MOOCs." Computers & Education 145 (2020): 103728.
[19] Li, Zhu, et al. "Kernel dependence regularizers and gaussian processes with applications to algorithmic fairness." arXiv preprint arXiv:1911.04322 (2019).
[20] Pooja.C, Thivaharan.s, "Workload based Cluster Auto Scaler using Kuberbet Monitors", International Journal Compliance Engineering Journal (IJCENG), 2021, Vol.12, Issue.6, pp. 40-47, ISSN:0898-3577, DOI:16.10089.CEJ.2021.V12I6.285311.36007.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY