Toward a New Hybrid Intelligent Sentiment Analysis using CNN- LSTM and Cultural Algorithms

Imtiez Fliss[a]
National School of Computer Science, Manouba University, Tunisia

Keywords: Sentiment Analysis, Hybrid CNN-LSTM Classifier, Cultural Algorithms, Hyper-parameters Optimization.

Abstract: In this paper, we propose a new sentiment analysis approach based on the combination of deep learning and soft computing techniques. We use the GloVe word embeddings for feature extraction. For sentiment classification, we propose to combine CNN and LSTM to decide whether the sentiment among the text is positive or negative. To tune hyperparameters, this classifier is optimized using cultural algorithms.

1 INTRODUCTION

Sentiment analysis (SA)(Mukherjee, 2021) is widespread across all fields and has become one of the most active topics in recent research. Whether we held a webinar, virtual event, or conference, collecting feedback from attendees and event stakeholders helps us plan and improve future events.
In other hand, contact centers strive to improve customer experiences across the customer journey. From evaluation and product purchase to delivery and after-sales support the need to ensure customers are happy is an ongoing priority.

There is a crucial question in each of these situations: stakeholders enjoyed their experience? The answers will help the event planner to improve the experience at his future events and on the other hand help to inform best practices and improve the customer experience (Jain and Kumar, 2017).

Sentiment analysis (Zhao et al., 2010; Medhat et al., 2014), is a type of subjective analysis which examines sentiment in each textual unit with the objective of understanding the sentiment polarities (for example: positive, negative, or neutral) of the opinions toward various aspects of a subject.

Sentiment Analysis is a multi-step process covering various sub-tasks: data collection, feature extraction and selection, and finally sentiment classification. The newest trend in sentiment analysis field has brought up additional demand for understanding the contextual representation of the language. Word embedding is one of the most popular representation of document vocabulary that is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. Thus, we intend to use Glove (Pennington et al., 2014) as an embedding technique for the step of feature extraction.

In the classification step, we can take advantage of the power of deep learning. Sentiment analysis models can be trained to understand the text context and recognize the opinion of the writer. CNN's (Albawi et al., 2017), a class of deep, feed-forward artificial neural networks, could be a good solution for sentiment analysis where sentiments are expressed by some key phrases. As it demonstrates its effectiveness in extracting local and position-invariant features. For long texts, another model of deep learning recurrent neural network based on long short-term memory (LSTM) (Graves, 2012) work better as it can learn the long-term dependence of text. Therefore, in this paper, we propose to combine the advantages of CNN and LSTM, a hybrid model CNN- LSTM is constructed for the sentiment classification task.

Despite the excellent results achieved, the performance of deep learning models in classification problems, they are strictly correlated to their hyperparameters (eg. the learning rate, the number of training epochs/iterations, the batch size, etc.). It is then crucial to select an appropriate optimization technique to detect optimal hyperparameters.
For hyperparameters tuning, a class of robust algorithms is required, which does not depend upon the context. To solve this problem, we propose in our work to use cultural algorithms (Reynolds, 1994) which is an evolutionary algorithm that takes advan-

[a] https://orcid.org/0000-0003-2229-7004

Fliss, I.
Toward a New Hybrid Intelligent Sentiment Analysis using CNN- LSTM and Cultural Algorithms.
DOI: 10.5220/0010990400003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 1*, pages 467-477
ISBN: 978-989-758-547-0; ISSN: 2184-433X

tage of knowledge domains and faster convergence and is used in many optimization problems (Maheri et al., 2021).

The details of our contributions are as follows:

- Proposing an end-to-end sentiment analysis approach
- Developing a combined CNN-LSTM model optimized using Cultural Algorithms to determine the best accurate results in sentiment analysis.

The rest of this paper is arranged as follows: Related works are introduced in Section2. A background about CNN, LSTM and cultural algorithms is presented in Section 3. In Section 4, we focus on the proposed sentiment analysis approach. The experiments are given in Section 5. Section 6 presents our contribution to the literature. Finally, some concluding remarks and perspectives are presented.

2 LITERATURE REVIEW

Sentiment analysis is concerned with the identification and classification of sentiments. To obtain the opinion from a text, it is necessary to extract some interesting information then proceed to text classification. In literature, two main approaches to sentiment analysis are proposed: Lexicon-based approaches and Machine learning-based approaches

2.1 Lexicon-based Approaches

Several lexicon-based approaches are proposed (Prakash and Aloysius, 2021). Researchers use corpora (Turney and Littman, 2005), lexicon (Kour et al., 2021) or more complex Wordnet (Fellbaum, 1998) and other language resources (Dragut et al., 2010) to generate dictionaries to support sentiment analysis in different contexts.

On the other hand, researchers in (Taboada, 2016) apply lexical resources named opinion lexicon, that associate words to sentiment orientation represented for example by positive and negative “scores.” It widely applies in sentiment analysis and begins with the assumption that a single word can be considered a unit of opinion information so that it can indicate the sentiment and subjective nature of texts. Emotional annotations can be done manually or through an automated, semi-supervised.

The sentiment classification result can be expressed as a positive or negative score in the form of a binary or can be expressed as a multi-emotional classification.

2.2 Machine Learning based Approaches

In case of Machine learning based approaches (Mitra, 2020), a collection of documents is “tagged” for some features. These documents are used to “train” the statistical model, which then is applied to new text. To get a better or larger data set to get improved results, it is necessary to retrain the model as it “learns” more about the documents, it analyzes. This supervised approach also applies to the sort of retraining that can happen with some models where some viewer gives a “star” rating – and the algorithm adds that rating to its ongoing processing (Redmore, 2013). There have been many studies on classifying sentiments using machine learning models, such as Support Vector Machine (SVM), Naive Bayes (NB), Maximum Entropy (ME), Stochastic Gradient Descent (SGD), and other techniques.

Deep learning (Dargan et al., 2020) is a subbranch of machine learning that uses deep neural networks. Recently, deep learning algorithms have been widely applied for sentiment analysis (Yadav and Vishwakarma, 2020; Dang et al., 2020; Minaee et al., 2021).

Deep Neural Networks are Artificial Neural Networks that present multiple hidden layers between input and output and exist in a plethora of diverse architectures depending on the network topology of neurons and their connections; among them, some have brought notable outcomes especially: Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

CNNs have shown the finest results in computer vision and image processing, and the same architecture has been widely applied to text processing (Yadav and Vishwakarma, 2020). The most renowned CNN-based sentiment analysis model was introduced by (Kim, 2014), extensively used by (Kalchbrenner et al., 2014) and enhanced by (Pota et al., 2020). Furthermore, Chen et al. (Chen et al., 2017) improved sentiment detection through a two-step’s architecture, leveraging separated CNNs trained on sentences clustered according to the number of opinion targets contained.

On the other hand, RNNs are used for modelling sequential data in a variety of applications. Methods based on RNNs fed the sentiment classifier with the complete sentence representation building it with a bottom-up approach (Socher et al., 2013). Moreover, the Long Short-Term Memory (LSTM) variant of RNNs (Hochreiter and Schmidhuber, 1997) can handle the declining gradient problem of basic RNNs, catching long-term dependencies. Therefore, LSTM networks were proven to perform better that standard RNNs for sentiment analysis (Li and Qian, 2016). Alayba et al. (Alayba et al., 2018) have shown the benefits of integrating CNNs and LSTMs, reporting a better accuracy on diverse data sets for Arabic sentiment analysis.

2.3 Synthesis

Using lexicon-based approaches to identify polarity of words, is not effective for all cases because it is hard to prepare a huge corpus to cover all words we can use.

However, these approaches can help to find domain-specific opinion words and their polarities (positive, negative or else) if a corpus from only the specific domain is used in the discovery process.

Alternatively, the machine learning approaches to sentiment analysis, also described as a supervised learning approach, is often reported to be more accurate according to (Chaovalit and Zhou, 2005; Wawre and Deshmukh, 2016; Al-Hadhrami et al., 2019) and has also been used in marketing research (Pathak and Pathak-Shelat, 2017; Rambocas and Pacheco, 2018). However, the machine learning approach and especially deep learning techniques (Mahendhiran and Kannimuthu, 2018; Yuan et al., 2020) requires a large corpus of training data and their performance depends on a good match between the training and testing data and parameters of the used algorithms.

Thus, we propose in this paper, to develop a new effective sentiment analysis model that aggregates deep learning sentiment analysis and soft computing techniques. The proposed hybrid sentiment analysis model can be applied to any textual data set (there is no restriction on structure of considered texts), that can be even cross-domain and cross-source as considered in (Zola et al., 2019).

3 BACKGROUNDS

3.1 Convolutional Neural Network

Convolutional Neural Network (CNN's) (Albawi et al., 2017), also known as ConvNets, consist of multiple layers and are widely used to identify satellite images, process medical images, forecast time series, detect anomalies, and intervenes in many other classification problems. CNN's have multiple layers, as shown in Figure 1, that process and extract features from data (input layer):

- Convolution Layer: The convolution layer has several filters to perform the convolution operation.
- Rectified Linear Unit (ReLU): ReLU layer is used to perform operations on elements. The output is a rectified feature map. The rectified feature map next feeds into a pooling layer.
- Pooling Layer: Pooling is a down-sampling operation that reduces the dimensions of the feature map. The pooling layer then converts the resulting two-dimensional arrays from the pooled feature map into a single, long, continuous, linear vector by flattening it.
- Fully Connected Layer: A fully connected layer forms when the flattened matrix from the pooling layer is fed as an input, which classifies and identifies the results (output layer) depending on the field (images, texts, etc.).

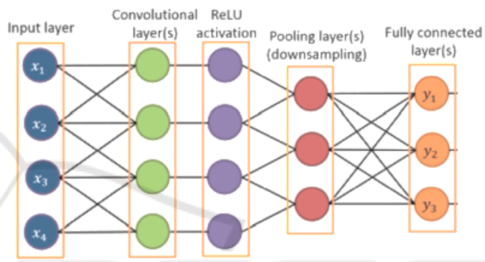

Figure 1: CNN architecture.

3.2 Long Short-Term Memory

Long Short-Term Memory (LSTMs)(Graves, 2012) are a type of Recurrent Neural Network (RNN) that can learn and memorize long-term dependencies. LSTMs retain information over time. LSTMs have a chain-like structure where four interacting layers communicate in a unique way. First, they forget irrelevant parts of the previous state. Then, they selectively update the cell-state values and the output of certain parts of the cell state are finally given.

The network takes three inputs. X_t is the input of the current time step. h_t-1 is the output from the previous LSTM unit and C_t-1 is the "memory" of the previous unit. As for outputs, h_t is the output of the current network. C_t is the memory of the current unit. Therefore, this single unit makes decision by considering the current input, previous output, and previous memory. And it generates a new output and alters its memory.

LSTM consists of three types of gates, namely forget gate, input gate and output gate which decides relevant and irrelevant information from the input data. Forget gate decides which previous information c(t-1) is not required, input gate selects relevant information from the input data x(t), and output gate produces new the hidden state h(t) for time 't.' At each timestamp 't,' h(t) also serves as the output produced by the long short-term cell for timestamp 't.' as presented in Figure 2.

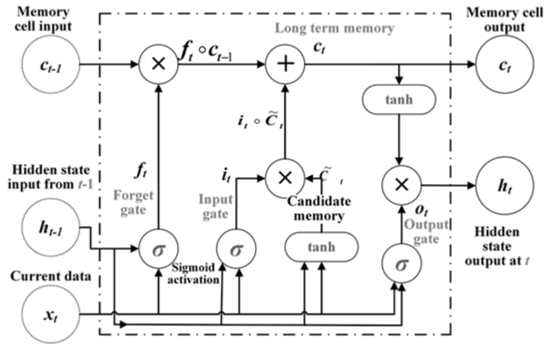

Figure 2: LSTM architecture.

3.3 Cultural Algorithms

Cultural Algorithms are proposed by Reynolds (Reynolds, 1994). They maintain two search spaces: the population representing the genetic component and the belief space representing the cultural component as shown in Figure 3. Both these search spaces

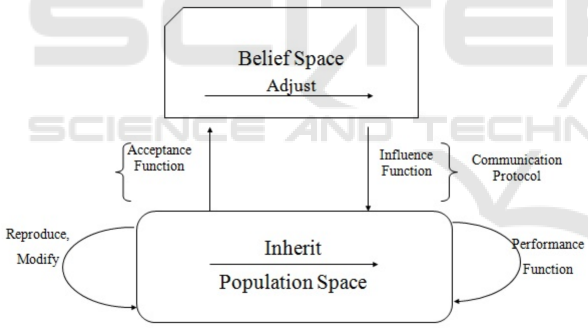

Figure 3: Cultural algorithm architecture.

evolve in parallel and each of them influence the other. The experiences of individuals in the population space, identified through an acceptance function, are used for the creation of knowledge residing within the belief space. An acceptance function determines which individual's experiences should be considered to contribute to the current beliefs. This knowledge is stored and manipulated in the belief space. These adjusted beliefs then influence the evolution of the population.

Population space and belief space communicate through the acceptance function and the influence function. The acceptance function determines which individuals from the current population are selected to impact the belief space. The selected individuals' experiences are generalized and applied to adjust the current beliefs in the belief space via the update function. The new beliefs can then be used to guide and influence the evolutionary process for the next generation knowledge circulation is described as follows:

1. The belief space receives the top best individuals within generation g from the population space using acceptance function.
2. The belief space updates its own knowledge.
3. In the next generation g+1, the belief space sends the updated knowledge through the influence function to the population space.
4. The population space combines the knowledge to generate offspring from generation g and produce next generation g + 1.
5. The top individuals within generation g + 1 are sent to the belief space to update its knowledge.

4 PROPOSED APPROACH

In our study, we consider Sentiment analysis as a supervised task since a labelled data set containing text documents and their labels is used for training a classifier. In fact, given a collection of labeled records (training set), each record contains a set of features (attributes), and the true sentiment class (label). We aim to find a model for the sentiment analysis as a function of the values of the features and previously unseen records should be assigned a sentiment class as accurately as possible. A test set is used to determine the accuracy of the model.

4.1 Data Set Preparation

The first step is the data set Preparation step which includes:

- loading a data set which is the input of our system.
- performing basic pre-processing: In preprocessing of the data, we:
  - Convert all the words of reviews into lowercase.
  - Remove punctuation from reviews (like @,!)
  - Remove all the stop words like a, an, the etc. from the reviews.
  - Convert all the words into stemming words.
  - Finally remove extra white spaces from the reviews.
- splitting the data set into train and validation sets.

4.2 Feature Extraction

After the data has been cleaned, formal feature extraction methods can be applied. In general, texts and documents are unstructured data sets. However, these unstructured text sequences must be converted into a structured feature. The purpose of text representation is to convert preprocessed texts into a form which computer can process.

In our work, we propose to use an embedding world technique. Word embedding is a method in which each word of a vocabulary is mapped to a real vector.

Within this context, we propose to use the Global Vectors for Word Representation (GloVE)(Pennington et al., 2014) which considers the statistic of occurrence of a word in a large corpus Glove leverages statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus.

The model produces a vector space (Word Embedding) with a meaningful substructure and performs well on similarity tasks. The thing that GloVe is trying is the following statement: if two words often appear within the same context, their meanings are strongly correlated.

4.3 Feature Selection

Text sequences in term-based vector models consist of many features. Thus, time complexity and memory consumption are very expensive for these methods and we should select the best features to build the model.

To address this issue, we propose to use a greedy optimization algorithm: Recursive Feature Elimination (Guyon et al., 2002). This algorithm allows to find the best performing feature subset by creating models and keeping aside the best or the worst performing feature at each iteration. The next model is created with the left features until all the features are exhausted. Finally, the features are ranked based on the order of their elimination.

4.4 Classification

This is the most important step of the sentiment analysis pipeline because it classifies the emotions in the text.

4.4.1 The Proposed Sentiment Analysis Classifier

As Deep neural networks have been shown to outperform classical machine learning algorithms in solving real-world problems and based on the survey. Thus, we propose to choose a deep learning classifier in this work.

More specifically, deep convolutional neural networks (CNNs) and deep Long Short-Term Memory networks (LSTM's) obtained the interesting results in several classification benchmarks, surpassing the classification capabilities of human experts.

Embedding vector forms the first layer followed by Convolutional network and its finally wrapped by LSTM. The design of our classifier is given in Figure 4. The input data layer is represented as an embedding

Figure 4: The design of the proposed CNN-LSTM sentiment classifier.

matrix. Then, the multiple convolutional filters slide over the matrix to produce a new feature map and the filters have various sizes to generate different features.

For example, we used the filter size 3 to extract the 3-gram features of words. The Max-pooling layer is used to calculate the maximum value as a corresponding feature to a specific filter. The max operation or function is the most used technique for this layer, and it is used in this experiment. The reason of selecting the highest value is to capture the most important feature and reduce the computation in the advanced layers.

Then the dropout technique is applied to reduce overfitting with the dropout value is 0.5. The output vectors of the Max-pooling layer become inputs to the LSTM networks to measure the long-term dependencies of feature sequences. The output vectors of the LSTMs are concatenated, and an activation function is applied to generate the final output class positive or negative in case of binary classification and very positive, positive, very negative, negative, and so in case of multi-class sentiment analysis.

4.4.2 Optimization of the Hyperparameters of Our Sentiment Analysis Classifier

The hyperparameters of deep learning networks have an important influence on the network's performance, as they directly control the training process. The selection of appropriate hyperparameters plays a vital role in the training of networks. For example, if the learning rate is too low, the network may lose important details in the data. By contrast, if the learning rate is too high, it may lead the model to converge too quickly.

Therefore, there is a need to optimize the hyperparameters of networks for proper training and optimum performance results. In our work, we aim to optimize the following hyperparameters: number of neurons, batch_size, number of iterations (epochs), the best activation function and the best optimizer.

Thus, we propose to use the cultural algorithms to tune these hyperparameters. In this algorithm, the belief space and the population space are first initialized. Then, the algorithm will repeat processing for each generation until a termination condition is achieved. The structure of Culture Algorithm can be described as given in the Figure 5. Population initialization is

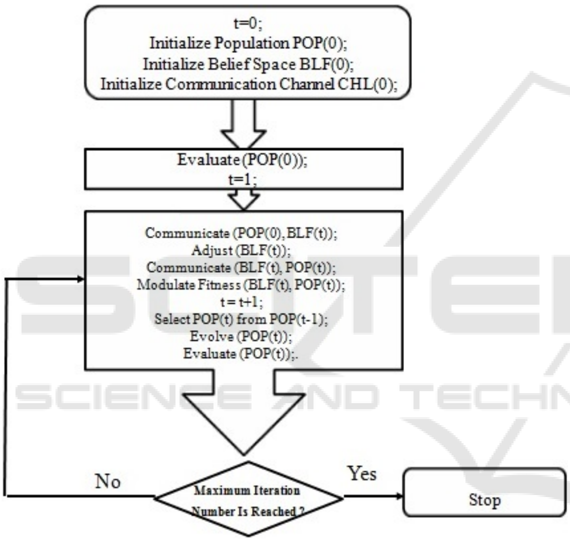

Figure 5: Flow-chart of the cultural algorithm.

the procedure where the first generation of the population is determined within the search space.

Within our study, the initial population is randomly generated. It is composed of Q individuals. Each individual represents a possible solution.

Within our study, we optimally adjust the following hyperparameters: number of neurons, batch_size, number of, the best activation technique and the best optimizer.

A set of D optimization parameters is called an individual and is represented by a D-dimensional parameter vector.

As previously mentioned, the belief space is the information repository in which the individuals can store their experiences. These experiences can lead the other individuals to learn them indirectly to help preserve diversity in the search.

The belief space initialization consists in initially creating Q empty belief spaces each belief space is associated to each individual.

The individuals (in the initial population) are then evaluated by the fitness function. As our aim is to have the best accuracy of our classifier. This information should be communicated, and accuracy ought to be maximized.

Then, the information on the performance of the fitness function is used as a basis to produce generalizations for next generations. The experiences of the individuals selected will be used to make the necessary adjustments on the knowledge of the current belief space.

A selection process is used to choose the parents to be evolved in the next generation. In the present work, ranked replacement is considered. To produce new solutions based on existing solutions in the population, we need combining and mutation operators.

Combining is the procedure of recombining the information carried by two individuals to produce new offspring. In our study, one point crossover is used.

Initially two individuals are chosen at random from the population. A crossover point which is a random integer whose value is less than the size of the individual is chosen at random and the contents of the individual after the crossover point are swapped. Crossover produces two children.

Mutation, on the other hand, alters one individual to produce a single new solution. Within our work, uniform mutation operator is used. The population component of the cultural algorithm is approximately the same as that of the genetic algorithm. The belief space is updated after each iteration by the best individuals of the population. The best individuals are selected using the fitness function.

Within our work the termination condition is the reaching of a maximum iteration number.

5 EXPERIMENTATION AND RESULTS

5.1 Datasets

In this work, we used two bechmark datasets: first for the training and the second one to test step. For training, we used Twitter US Airline Sentiment Dataset. The data originally came from CrowdFlower's Data for Everyone library. Contributors scraped Twitter data of the travelers who traveled through six US airlines since February 2015. They provided the data on

Kaggle as a dataset, named Twitter US Airline Sentiment (https://www.kaggle.com/crowdflower/twitter-airline-sentiment). The dataset has around 14640 records and 15 attributes. It contains whether the sentiment of the tweets in this set was positive, neutral, or negative for six US airlines services.

For testing, we are based on benchmark dataset Large Movie Review (Maas et al., 2011) downloaded from Kaggle (https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews).This dataset contains 50000 highly polar movie reviews. Each instance contains an entire review written by one individual (the length of reviews are different).

5.2 Evaluating Indicator

In order to evaluate the performance of our model, we use accuracy as the evaluation criteria of our experiments. Accuracy is the fraction of predictions our model got right.

Formally, accuracy has the following definition:
Accuracy=$\frac{Number of correct predictions}{Total number of predictions}$ (1)
Accuracy can also be calculated in terms of positives and negatives as follows:
Accuracy=$\frac{TP+TN}{TP+TN+FP+FN}$ (2)
Where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative.
TN(True Negative) represents the number of true negative classes, that is, the number of samples predicted as containing negative sentiment and actually contain negative sentiment.
FN(False Negative) represents the number of false negative classes, that is, the number of samples predicted as containing negative sentiment and actually contain positive sentiment.
FP(False Positive) represents the number of false positive classes, that is, the number of samples predicted as containing positive sentiment and actually contain negative sentiment.
TP(True Positive) represents the number of true positive classes, that is, the number of samples predicted as containing positive sentiment and actually contain positive sentiment.

5.3 Experimental Results

First, the datasets are loaded. They are then preprocessed as described previously by lower casing, removing stop words, etc..

Besides, we use a word embedding from pre-trained GloVe to Build the vocab. GloVe was pre-trained on a data set of one billion tokens (words) with a vocabulary of 400 thousand words at ”glove.6B/glove.6B.100d.txt” using 300D vectors. Using the Recursive Feature Elimination algorithm, we reduce the word embeddings to 50D.

For each dataset, the word embedding matrix is finally obtained.

5.3.1 Initialization of the Cultural Algorithm Parameters

To initialize the cultural algorithm, we use the parameters listed in Table 1.

Table 1: Initial optimization parameters.

Parameter	Value
Population Size	200
Number of Iterations	10000
Optimizer	in [0,7]
Batch_size	in [0, 200]
Activation function	in [0,9]
Neurons	in [10, 100]
Epochs	in [1, 50]

5.3.2 The Resulted Cultural Algorithm Parameters

After the evolution process of the Cultural Algorithm previously presented, we get the hyperparameters given in Table 2.

Table 2: Optimized hyperparameters.

Parameter	Value
Optimizer=	’adam’
Activation function=	’sigmoid’
batch_size=	64
Neurons=	38
Epochs=	47

5.3.3 CNN and LSTM Classifier Optimized by Cultural Algorithm Predictions

Once the hyperparameters of our CNN and LSTM classifier are adjusted, we use this classifier to predict the sentiments in the test data set. At each epoch (iteration), we save the model accuracy. The accuracy metric measures the ratio of correct predictions over the total number of instances evaluated. The results of our CNN-LSTM classifier optimized by cultural algorithm predictions are given in Figure 6. The accuracy average of the sentiment analyser using both CNN and LSTM with hyperparameters’ optimization equals to 96.116%.

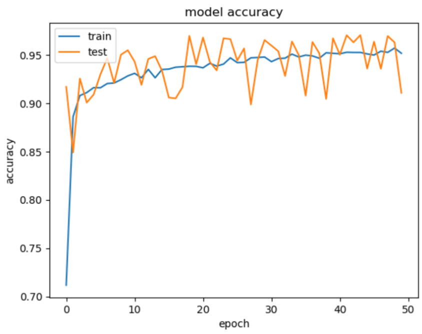

Figure 6: CNN and LSTM classifier optimized by cultural algorithm prediction results.

5.4 Sentiment Analyzer Evaluation

To evaluate the proposed sentiment analyzer, we first evaluate the accuracy of sentiment analyzer using only CNN(Figure 7-(a), then using only LSTM(Figure 7-(b), then using both CNN and LSTM without hyperparameters optimization(Figure 7-(c))and finally compared their accuracies with the accuracy of the proposed sentiment analyzer using both CNN and LSTM with cultural optimization of hyperparameters.

The results of the accuracy of each classifier on entire Test set are summed up in Table 3.

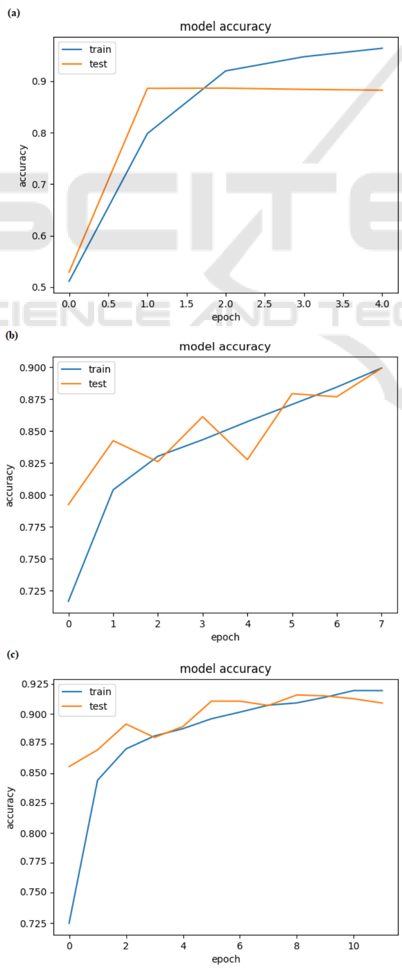

Figure 7: (a) CNN predictions' results; (b) LSTM predictions' results; (c) CNN-LSTM(without hyperparameters optimization) predictions' results.

Table 3: Sentiment analysers evaluation.

Sentiment Analyser	Accuracy
Using CNN	86.524%
Using LSTM	85.763%
Using CNN+LSTM without Optimization	90.744%
Using CNN+LSTM+Cultural Algorithm	96.116%

5.5 Discussion

Based on our experimental results, we can conclude that associating CNN and LSTM optimized by cultural algorithm would be an interesting solution to sentiment analysis.

In fact, the experimental results show that using both CNN and LSTM (Figure 7-(c) and Figure 6 respectively) ameliorate the accuracy of the sentiment classifier in the case of the used data set.

They also surpass the effectiveness of the only use of CNN (Figure 7-(a)) and LSTM (Figure 7-(b)). The using of the cultural algorithm optimizes and ameliorates the accuracy of CNN-LSTM sentiment analyzer.

6 CONTRIBUTION TO THE LITERATURE

The integration of several techniques for sentiment analysis has already been presented elsewhere (Appel et al., 2016; Al Amrani et al., 2018; Srivastava et al., 2019; Raisa et al., 2021; Jain et al., 2021). However, the main contribution of our study concerns the definition of a new approach to classify sentiments in several texts which benefits from the pros of text mining, deep learning and soft computing. The main contributions of our study can be summed up in three points.

First of all, our proposed approach is generic, it can be applied to any textual data set (there is no restriction on the length or structure of the considered text that can be even cross-domain and/or cross-source).

Besides, the proposed approach is based on the combination of GloVe and deep learning techniques: CNN and LSTM algorithm which are of importance for overall performance, computation cost and convenience of real-world problems and especially in text classification according to (Cai et al., 2018). In fact, CNN is a good solution for sentiment analysis where sentiments are expressed by some key phrases as it can extract local and position-invariant features. For long texts, (LSTM) works better as it can learn the long-term dependence of text. Thus, combining both algorithms allows us to analyse sentiments efficaciously for all types of texts (short and long ones). Using Glove word embeddings on the other hand proves to perform one numeric representation of a word (which we call embedding/vector) regardless of where the words occurs in a sentence and regardless of the different meanings they may have. Glove is context independent and its output is just one vector (embedding) for each word, combining all the different senses of the word into one vector.

Finally, performance of deep learning algorithms is quiet related to their hyperparameters. The best way to determine the best values to hyperparameters is to optimize them. The search process in the standard Evolutionary algorithms is unbiased; it uses only a little or no domain knowledge to direct the search process. The performance of the Evolutionary algorithms can be improved considerably by using domain knowledge; it makes the search process biased. Exchange of knowledge between the individuals in the environment can help them to explore and exploit conditions around them more precisely. Thus, we propose, in this work to use Cultural algorithms which is one of the popular types of Evolutionary algorithms. Cultural algorithms incorporates knowledge to guide the search process and to find better solutions with high quality. The experimental results prove the pros of using this optimization technique.

7 CONCLUSION

If we have thousands of feedbacks per month, it is impossible for one person to read all these responses. By using automating sentiment analysis, we can easily gauge how customers feel and improve our business.

In this context, we are particularly interested in proposing an intelligent approach for properly automate analyzing sentiments. This approach is based on the combination of CNN and LSTM.

The main problem of deep learning algorithms is related on the determination of the best hyperparameters. Thus, to optimize the classifier hyperparameters a cultural algorithm is used.

The proposed classifier is a step of an end to-end sentiment analysis pipeline. This pipeline begins by choosing a data set. A preprocessing step is then made. Then an embedding matrix is generated using Glove approach. The features are then reduced. The CNN-LSTM optimized by cultural algorithm is used and finally the sentiment within the text is predicted.

The experiment results are promising compared to classifying sentiments using CNN(only), LSTM(only) and combining CNN and LSTM without optimizing the hyperparameters.

In future works, we will focus on other sentiment analysis data sets and we will try to combine other models and features.

REFERENCES

Al Amrani, Y., Lazaar, M., and El Kadiri, K. E. (2018). Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Computer Science*, 127:511–520.

Al-Hadhrami, S., Al-Fassam, N., and Benhidour, H. (2019). Sentiment analysis of english tweets: A comparative study of supervised and unsupervised approaches. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–5. IEEE.

Alayba, A. M., Palade, V., England, M., and Iqbal, R. (2018). A combined cnn and lstm model for arabic sentiment analysis. In *International cross-domain conference for machine learning and knowledge extraction*, pages 179–191. Springer.

Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee.

Appel, O., Chiclana, F., Carter, J., and Fujita, H. (2016). A hybrid approach to sentiment analysis. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 4950–4957. IEEE.

Cai, J., Li, J., Li, W., and Wang, J. (2018). Deeplearning model used in text classification. In *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 123–126. IEEE.

Chaovalit, P. and Zhou, L. (2005). Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th annual Hawaii international conference on system sciences*, pages 112c–112c. IEEE.

Chen, T., Xu, R., He, Y., and Wang, X. (2017). Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230.

Dang, N. C., Moreno-García, M. N., and De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3):483.

Dargan, S., Kumar, M., Ayyagari, M. R., and Kumar, G. (2020). A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4):1071–1092.

Dragut, E. C., Yu, C., Sistla, P., and Meng, W. (2010). Construction of a sentimental word dictionary. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1761–1764.

Fellbaum, C. (1998). A semantic network of english: the mother of all wordnets. In *EuroWordNet: A multilingual database with lexical semantic networks*, pages 137–148. Springer.

Graves, A. (2012). Long short-term memory. In *Supervised sequence labelling with recurrent neural networks*, pages 37–45. Springer.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jain, P. K., Saravanan, V., and Pamula, R. (2021). A hybrid cnn-lstm: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5):1–15.

Jain, V. K. and Kumar, S. (2017). Improving customer experience using sentiment analysis in e-commerce. In *Handbook of Research on Intelligent Techniques and Modeling Applications in Marketing Analytics*, pages 216–224. IGI Global.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv:1404.2188*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Kour, K., Kour, J., and Singh, P. (2021). Lexicon-based sentiment analysis. In *Advances in Communication and Computational Technology*, pages 1421–1430. Springer.

Li, D. and Qian, J. (2016). Text sentiment analysis based on long short-term memory. In *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, pages 471–475. IEEE.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Mahendhiran, P. and Kannimuthu, S. (2018). Deep learning techniques for polarity classification in multimodal sentiment analysis. *International Journal of Information Technology & Decision Making*, 17(03):883–910.

Maheri, A., Jalili, S., Hosseinzadeh, Y., Khani, R., and Miryahyavi, M. (2021). A comprehensive survey on cultural algorithms. *Swarm and Evolutionary Computation*, 62:100846.

Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2021). Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.

Mitra, A. (2020). Sentiment analysis using machine learning approaches (lexicon based on movie review dataset). *Journal of Ubiquitous Computing and Communication Technologies (UCCT)*, 2(03):145–152.

Mukherjee, S. (2021). Sentiment analysis. In *ML. NET Revealed*, pages 113–127. Springer.

Pathak, X. and Pathak-Shelat, M. (2017). Sentiment analysis of virtual brand communities for effective tribal marketing. *Journal of Research in Interactive Marketing*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pota, M., Esposito, M., Pietro, G. D., and Fujita, H. (2020). Best practices of convolutional neural networks for question classification. *Applied Sciences*, 10(14):4710.

Prakash, T. N. and Aloysius, A. (2021). Textual sentiment analysis using lexicon based approaches. *Annals of the Romanian Society for Cell Biology*, pages 9878–9885.

Raisa, J. F., Ulfat, M., Al Mueed, A., and Reza, S. S. (2021). A review on twitter sentiment analysis approaches. pages 375–379.

Rambocas, M. and Pacheco, B. G. (2018). Online sentiment analysis in marketing research: a review. *Journal of Research in Interactive Marketing*.

Redmore, S. (2013). Machine learning vs. natural language processing.

Reynolds, R. G. (1994). An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming*, volume 24, pages 131–139. World Scientific.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Srivastava, A., Singh, V., and Drall, G. S. (2019). Sentiment analysis of twitter data: A hybrid approach. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, 14(2):1–16.

Taboada, M. (2016). Sentiment analysis: An overview from linguistics.

Turney, P. D. and Littman, M. L. (2005). Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1):251–278.

Wawre, S. V. and Deshmukh, S. N. (2016). Sentiment classification using machine learning techniques. *International Journal of Science and Research (IJSR)*, 5(4):819–821.

Yadav, A. and Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385.

Yuan, J., Wu, Y., Lu, X., Zhao, Y., Qin, B., and Liu, T. (2020). Recent advances in deep learning based sentiment analysis. *Science China Technological Sciences*, pages 1–24.

Zhao, Y.-Y., Qin, B., Liu, T., et al. (2010). Sentiment analysis. *Journal of Software*, 21(8):1834–1848.

Zola, P., Cortez, P., Ragno, C., and Brentari, E. (2019). Social media cross-source and cross-domain sentiment classification. *International Journal of Information Technology & Decision Making*, 18(05):1469–1499.