An Optimized Customers Sentiment Analysis Model Using Pastoralist Optimization Algorithm (POA) and Deep Learning

[1st] Safiya A. Shehu
Department of Computer Science
Federal University of Technology, Minna
Minna, Nigeria
salsaf3637@gmail.com

[2nd] Abdulmalik D.Mohammed
Department of Computer Science
Federal University of Technology, Minna
Minna, Nigeria
drmalik@futminna.edu.ng

[3rd] Ibrahim M Abdullahi
Department of Computer Engineering
Federal University of Technology, Minna
Minna, Nigeria
amibrahim@futminna.edu.ng

***Abstract*—Users usually express their sentiments online which has great influence on the product customers buy. Sentiment analysis is the computational study of people's emotions toward an entity. Sentiment analysis often faces the challenge of insufficient labeled data in Natural Language Processing (NLP) and other related areas. Long Short-Term Memory (LSTM) is one of the deep learning models widely used by researchers in solving sentiment analysis problem. However, they possess some drawbacks such as longer training time, more memory for training, easily overfits, and sensitivity to randomly generated parameters. Hence, there is a need to optimize the LSTM parameters for enhanced sentiment analysis. This paper proposes an optimized LSTM approach using a newly developed novel Pastoralist Optimization Algorithm (POA) for enhanced sentiment analysis. The model was used to analyze sentiments of customers retrieved from Amazon product reviews. The performance of the developed POA-LSTM model shows optimal accuracy, precision, recall. and F1 measure of 77.36%, 85.06%, 76.29%, and 80.44% respectively when compared with the LSTM model with 71.62%, 78.26%, 74.23%, and 76.19% respectively. It was also observed that POA with 20 pastoralist population size performs better than other models with 10, 15, 25, and 30 population size.**

Keywords—*Sentiment Analysis, Natural Language Processing (NLP), Deep Learning, Pastoralist Optimization Algorithm*

I. INTRODUCTION

In recent technology, a huge amount of information, data, reviews, or opinions is being stored in the websites of social media or e-services in the form of raw data. In other to work with those raw data proper methods are required. A study that describes peoples' opinions concerning products, services, and other characteristics is termed sentiment analysis. [1]. Sentiment analysis systematically identifies, quantifies, extracts, and studies affective states and subjective information. It is often used in Web, text, and data mining, and for information retrieval [24]. Sentiment analysis covers other sciences such as; computer, social and management sciences, and so on. To analyze sentiments, objects, and characteristics, viewpoints holder, and direction are the three terms that are used. Sentiment Analysis involves some challenges such as object recognition, opinion orientation classification, and feature extraction. Popular supervised and unsupervised machine learning algorithms have been successfully applied to sentiment analysis [7].

Deep learning, an advanced machine learning model has solved some of the challenges brought about by the lack of vocabulary resources and the improvement of sentiment classification in this field. There are several deep learning models deployed for sentiment analysis. Some of the popular deep learning models include Convolutional Neural network (CNN), Deep Belief Network (DBN), and Recurrent Neural Networks (RNN) [26]. Deep learning has been successful in solving various challenging problems such as Speech recognition, Natural language Processing, (NLP), and Computer vision applications like face recognition. Despite its success, determining the appropriate layers, the number of hidden variables a hidden layer should have, slow training is among the greatest challenge of deep learning [25]. Pastoralist Optimization Algorithm (POA) is a novel metaheuristic inspired by the herding strategies of nomadic pastoralists and developed for optimization [22]. The algorithm has been very successful in solving combinatorial optimization problems and therefore, possesses a suitable candidate for optimizing the deep learning model for sentiment analysis.

In this paper, an optimized sentiment analysis model using deep learning (LSTM) and POA for optimizing the model was proposed. The model will be tested on datasets obtained from the social interactions of users. When developed, the model will improve sentiment analysis tasks by improving the LSTM model and presents an opportunity to explore ideas of audience members and study the state of the product from the opposite perspective. This makes sentiment analysis an ideal tool for expanding product analysis and other market and public business analysis.

The rest of this paper is structured as follows: Section II introduces related works which comprise of sentiment analysis review, deep learning, and POA. In section III, the materials and methods required to accomplish the research objectives are presented. Section IV is a presentation of the expected results and in section V, the conclusion is presented.

II. REVIEW OF RELATED WORKS

A. Sentiments Analysis

The research interest in sentiment analysis has grown over the years due to its importance in various sectors of life. Sentiment analysis can be classified based on some criteria which include; techniques used, dataset structure, and rating level [12]. The figure shows several categories of sentiment analysis. The various ways in which sentiment analysis can be implemented are;

i. Machine learning-based: This involves training a sentiment analysis model with the existing dataset before deployment [12].

ii. Rule-based: Extracts information from a dataset and tries to assess them according to the polarity of words. There are different rules such as negation words, idioms, dictionary polarity, emoticons [13].

iii. Lexicon-based: Using Semantic orientation in the measurement of opinion and subjectivity of a review or comment generates sentiment polarity either positive or negative [14].

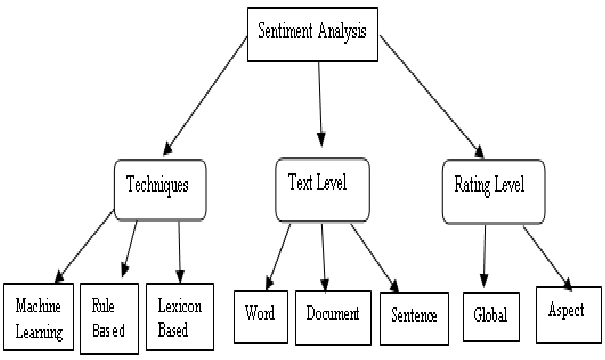


Fig. 1: Categorization of Sentiment Analysis

There are several kinds of research works on sentiment analysis deploying several techniques. This review focuses on reporting the several techniques that have been deployed for sentiment analysis with more emphasis on deep learning architectures. Wang et al. [3] established a bilingual attention network model for sentiment prediction of code conversion. LSTM model was applied for changing each post to its document level representation from which an attention mechanism was used to obtain from different contexts. [4] applied LSTMs to predict sentiments of social media users using multilingual connotation frames as its key method. In [5], an attention-based LSTM for effective sentence recognition.

In [27], a two attention-based two-way LSTM was developed to improve sentiment analysis performance. [6] extends the attention model by distinguishing the attention obtained from the left and right contexts of a given target. They further controlled their attention contribution by adding multiple gates. In [9], a DNN was proposed for information collection. A cascade of LSTM and DNN have constructed document representation and sentiment analysis respectively. Akhtar et al. [7] presented the analysis of ensemble models for emotional grading of financial microblogs and news. In [8], product and user feature and preferences while classifying their sentiments.

B. Deep Learning

Deep learning is an advanced ANN that deployed multiple deep network layers for learning. Due to its ability to solve problems faster than shallow g networks, deep learning has gained more and more attention in recent years. The advancement in computing and big data analytics have made its' deployment feasible [23]. Deep learning models are capable of solving both supervised and unsupervised problems [24]. Popular deep learning models include Convolutional Neural network (CNN), Deep Belief Network (DBN), and Recurrent Neural Networks (RNN) [26]. Deep learning has been successful in solving various challenging problems such as Speech recognition, Natural language Processing, (NLP), and Computer vision applications like face recognition. Despite its success, determining the appropriate layers and number of hidden variables a hidden layer should have has been among the greatest challenge of deep learning [25].

Sentiment analysis is a challenging problem that is being solved using deep learning. Some characteristics of Deep learning include; possessing nonlinear nodes that are arranged in several layers used for transforming and extracting features. [13]. A Deep Coupled Adjective and Noun (DCAN) neural model was proposed by Wang *et al.*, [10]. The key to this technique is harnessing the adjective and noun text descriptions for emotional expressions learning and subsequent sentiment classification. [11] propose a deep neural network model based on LSTM- and CNN-which utilizes word2vec and language embedding to classify claims (classifying sentences to be factual or feeling). [15] proposed a visual sentiment framework using a convolutional neural network and implemented their model on Flickr and Twitter images.

Long Short-term Memory (LSTM) is a special type of RNN used to learn long-term dependencies [23: 25]. Like other RNNs LSTM has a repetitive model, but it is complicated. It has four layers interacting especially together with a hidden state and cell state. Figure 2 shows a typical LSTM model.

.

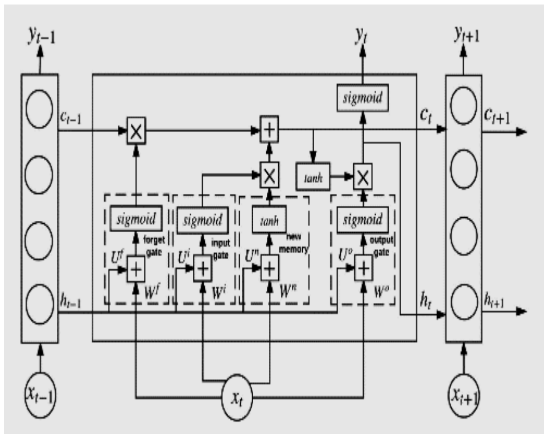


Fig. 2: LSTM deep learning model [23]

Mathematically, LSTM are represented mathematically as follows [23:26]:

$$I_t = \Phi(W_i x_t + V_i H_{t-1} + B_i) \quad (1)$$

$$F_t = \Phi\left(W_f x_t + V_f H_{t-1} + B_f\right) \quad (2)$$

$$O_t = \Phi(W_o x_t + V_o H_{t-1} + B_o) \quad (3)$$

$$C_t = F_t \odot C_{t-1} + I_t \odot tanh(W_c x_t + V_c H_{t-1} + B_c) \quad (4)$$

$$H_t = O_t \odot \tanh(C_t) \quad (5)$$

Where, W and V are the weight parameters and B is the bias vector, x_t and H_t are the input and hidden state vector of LSTM unit at t time respectively while, I_t, O_t, and F_t are the activation vector of input, output, and forget gate respectively. Finally, Φ and C_t are the sigmoid function and memory cell state vector.

C. Pastoralist Optimization Algorithm

Pastoralist Optimization Algorithm (POA) is a novel metaheuristic inspired by the socio-cultural lifestyle of nomadic pastoralists [22]. It mimics the behavior of nomadic pastoralists in the quest for quality pasture, water, and environment for their livestock. In POA, the search agent is called pastoralist and the i[th] pastoralist is represented as:

$$P_i = [P_{1,1}, P_{1,2}, P_{1,3}, \ldots, P_{i,D}] \quad (6)$$

Where D is the dimension of the problem. POA process is made up of two basic phases; the scouting and camping phases. In the scouting phase, pastoralist moves faster with longer step size, and the best scout location is used as camp. The camping phase is characterized by slower movement with a shorter step size. Herding pastoralist split to different locations to minimize local optima entrapment [20]. The initial location of the j[th] pastoralist (S_j) is given in Equation (7) and the new location of the j[th] pastoralist is given in Equation (8) and the evaluation continues until maximum scouting rate is reached [22]

$$S_j = rand([L_b, U_b]^D) \quad (7)$$

$$S'_j = (S_b - S_j) + \varepsilon_j * \eta_j * \zeta \quad (8)$$

Where S'_j is the new location of scout j around the best-found location S_b, ε_j is the energy of scout j ($\varepsilon \in \{-1,1\}$), η_j are the step size of scout j ($\eta \in \{0, (0.001 * \mathrm{Ub})\}$) and ζ is a positive constant that represents the number of times Scouters move faster than herders. The best scout location (S_b) is initialized as the camp location. Splitting by herd pastoralist was achieved using Equation (9) and after each split, the camp size is shrunk using Equation (10).

$$P'_K = P_b + (rand(0,r) * \varepsilon_k * \eta_k) \quad (9)$$

$$r'' = \frac{r'}{nP} \quad (10)$$

Where P'_K is the new location of the k[th] pastoralist, P_b is the best pastoralist so far, $rand(0,r)$ is a random number between 0 and r, $\boldsymbol{r}$ is the camp radius, $\varepsilon_{\boldsymbol{k}}$ is the energy of the k[th] pastoralist ($\varepsilon \in \{-1,1\}$) and $\eta_{\boldsymbol{k}}$ is the step size of the k[th] pastoralist ($\eta \in \{0, (0.001 * \mathrm{Ub})\}$). Also, r'' is the camp radius of current iteration, and r' is the camp radius of the previous iteration. If all locations have been exploited, the best camp location is returned and if all locations have been explored, the best camp location is returned as the global optimum solution, else, the process is repeated with the new scout locations determined using Equation (11).

$$S''_j = rand(L_b, U_b]^D) - S_b \quad (11)$$

Where S''_j is the new scout location, L_b and U_b are the lower and upper limit of the search space respectively. POA was evolved using biological evolution strategy and has been very successful when tested on numerical optimization and other combinatorial optimization problems. Other variants of POA with cultural evolution strategy have also been [21], Fig. 3 shows POA steps.

Algorithm 1: Pastoralist Optimization Algorithm
Input: POA Parameters; Search space
Output: Optimal Solution

i. Start
ii. Initialize POA parameters
iii. Select scout pastoralist randomly and initialize scout location
iv. Evaluate the fitness of each scout, update scout locations and normalize scouts' locations within the search space until maximum scouting rate is reached
v. Select best camping location
vi. Evaluate fitness of pastoralist and determine best pastoralist within a camp
vii. Split pastoralist to different locations within camp and evaluate fitness of each pastoralist
viii. Repeat step vii until maximum splitting rate is reached. For each split, divide the current camp radius by the number of pastoralists
ix. Update the best camp pastoralist
x. If all regions within the search space have not been explored,
   a. Update scout location
   b. Repeat steps iv to x and update the global camp best pastoralist
xi. Else, return the global best-found pastoralist
xii. Stop

Fig. 3: Pastoralist Optimization Algorithm (POA) [22]

III. EXPERIMENTAL SETUP

Fig. 4 shows the designed framework to achieve the aim and objectives of this paper. It starts with data collection, preprocessing, and feature extraction. Then, the optimized LSTM model design, training, testing, and finally, the performance evaluation.

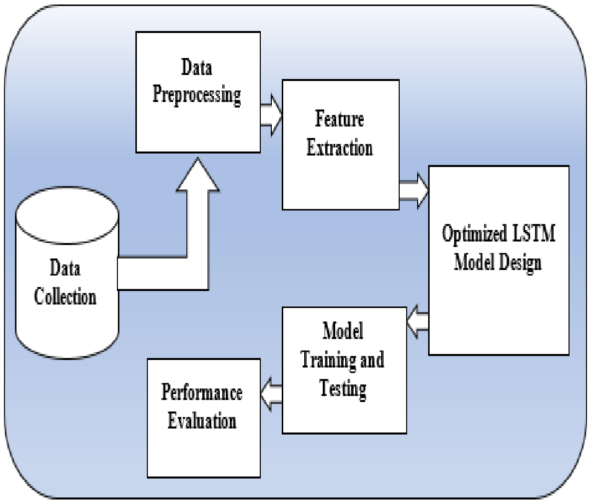


Fig 4: Proposed Methodology

The dataset used for the experiments comprises three review datasets of the Computer dataset which contains 531 sentences. The reviews of all the datasets are from amazon which can be obtained from amazon.com. Table I shows a sample of the computer dataset obtained from amazon. After data collection, the user's post will be preprocessed to transform it from its raw form into a form that enables the machine learning algorithms to understand it. Preprocessing also provides the opportunity to remove noise from the data, which can give more accurate learning algorithms. The pre-processing steps are;

- Removal of URLs

- Removal of special symbols/emoticons
- Removal of stop words from the dataset
- Tokenize the dataset (converting a sequence of strings into pieces of strings/tokens with an assigned or identified meaning).

After data preprocessing, the features required by the learning algorithm for training are extracted. There are two different approaches for text representation which are; word vector representation which represents each word in a document as a vector of N-dimensions. This approach does not capture semantic information and the order or structure of words in a document. It is only concerned with the occurrence of words. The second approach adopted in this paper is word embeddings representation. Word embeddings ensures that the deep learning model receives appropriate syntactic and semantic information by grouping similar words of a text collection in a vector space. Table (I) shows a sample of the dataset.

TABLE I: DATASET SAMPLE

Sentiment	Sentiment Text
1	This item was the most inexpensive 17-inch monitor available to me at the time I made the purchase.
-1	My overall experience with this monitor was very poor.
-1	When the screen wasn't contracting or glitching the overall picture quality was poor to fair.
-1	I've viewed numerous different monitor models since I 'm a college student and this particular monitor had as poor of picture quality as any I 've seen.
-1	A week out of the box and I began to see slight contractions of the screen from time to time , growing more frequent each day.
-1	Display glitches and flashes also occurred.
-1	I could tell this was a `` cheap " monitor as soon as I set it up.

A. Proposed Model Implementation Steps

Design optimized POA-LSTM and LSTM models by setting appropriate parameters of the LSTM model, such as number of epochs, learning rate, number of hidden units/nodes in the LSTM layer, number of layers, and sequence input. The optimal number of hidden nodes and learning rate are the two parameters that were optimized by the POA. Also, the parameters of the POA, number of pastoralist or population size was investigated.

Six models were trained using 70% of the dataset and tested using 30% untrained data. The first five models are optimized POA-LSTM for 10, 15, 20, 25, and 30 hidden nodes, while the sixth model is the LSTM model. The fitness function (F) used by the algorithm for fitness evaluation is the mean squared error given as:

$$F = maximize\ \frac{TP + TN}{TP + TN + FP + FN} \qquad (12)$$

Where, TP (True Positive) is correctly classified as positive sentiments, TN (True Negative) is correctly classified as negative sentiments, FP (False positive) is incorrectly classified as negative sentiments, and FN (False Negative) is incorrectly classified as positive sentiments. Fig. 5 shows the steps in implementing the proposed POA-LSTM sentiment analysis.

Algorithm 2: POA-LSTM Sentiment Analysis Algorithm
Input: Sentiment data; LSTM model
Output: Optimal position; Best model

i. *Start*
ii. *Data collection and pre-processing*
iii. *Feature extraction*
iv. *LSTM model design*
v. *POA parameters initialization and initial population generation*
vi. *Select scout pastoralist randomly and initialize scout location*
vii. *Train and test LSTM model*
viii. *Evaluate the fitness of each scout, update scout locations and normalize scouts' locations within the search space until maximum scouting rate is reached*
ix. *Select best camping location and initialize pastoralist in the camp*
x. *Train and test LSTM model using new camp locations*
xi. *Evaluate fitness of pastoralist and determine best pastoralist within camp*
xii. *Split pastoralist to different locations within camp*
xiii. *Train and test LSTM model using new camp locations*
xiv. *Evaluate fitness of each pastoralist and update local best pastoralist*
xv. *Repeat step xii – xiv until maximum splitting rate is reached. For each split, divide the current camp radius by the number of pastoralists*
xvi. *If all regions within the search space have not been explored,*
    a. *Update scout location*
    b. *Repeat steps vii - xv and update the global camp best pastoralist*
xvii. *Else, return the global best-found pastoralist*
xviii. *Return best model*
xix. *Stop*

Fig. 5: Proposed POA-LSTM implementation steps

B. Performance Evaluation

The performance of the developed models was evaluated using accuracy, precision, and F1 score performance metrics. They are represented mathematically as;

- **Accuracy**

The accuracy of a model is the ratio of all correctly classified samples over all samples and is given as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \qquad (13)$$

- **Precision**

Precision is the fraction of samples that were correctly classified and is given as;

$$Precision = \frac{TP}{TP + FP} \times 100\% \qquad (14)$$

- **Recall**

Is the ratio of the number of correctly classified positive sentiment and all the positive samples. It is given as:

$$Recall = \frac{TP}{TP + FN} \times 100\% \qquad (15)$$

- **F1 Score**

The F1 score is the balance between precision and recall and it is given as:

$$F1 = 2\left(\frac{Precision * Recall}{Precision + Recall}\right) \qquad (16)$$

IV. RESULTS AND DISCUSSION

The results obtained for the experiments performed are presented and discussed in this section. Fig 6 shows the convergence curves of the POA-LSTM models for the population size of 10, 15, 20, 25, and 30. The curve indicates the optimum fitness value and the convergence rate for each population size. The result shows that at a population size of 20, the algorithm converges obtaining an optimum value of 0.7736 which is the best value compared to other population sizes. At a population size of 10, 15, and 25, the fitness value is 0.761 and reduces to 0.7484 when the population size was increased to 30. The optimal learning rate and number of hidden nodes selected for 10, 15, 20, 25 and 30 population sizes are; 0.01-17, 0.02-18, 0.01-37, 0.01-5 and 0.02-21 respectively as shown in Table II.

The confusion matrices of the five POA-LSTM models and LSTM models are shown in Fig. 7. The matrix is a count of actual sentiments against predicted sentiments. The matrix shows that the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) are; 80, 41, 21, 17 for POA-LSTM (10), 83, 38, 24, 14 for POA-LSTM (15), 74, 49, 13, 23 for POA-LSTM (20), 81, 40, 22, 16 for POA-LSTM (25), 78, 41, 21, 19 for POA-LSTM (30), and 83, 38, 24, 14 for LSTM.

Fig. 8 shows the performance evaluation results of the developed models. It includes the accuracy, precision, recall, and F1 score performance metrics calculated from the TP, TN, FP, and FN values obtained in the confusion matrices in Fig. 7. The accuracy, precision, recall, and F1-Score for the developed models are as follows; for POA-LSTM (10) model, the values are 76.10%, 79.21%, 82.47%, and 80.81% respectively, for POA-LSTM (15) model are 76.10%, 77.57%, 85.57%, and 81.37% respectively. For POA-LSTM (20) model, the values are 77.36%, 85.06%, 76.29%, and 80.44% respectively, for POA-LSTM (15) model are 76.10%, 78.64%, 83.51%, and 81% respectively, while for POA-LSTM (30) model are 74.84%, 78.79%, 80.41%, and 79.59% respectively.

Finally, the accuracy, precision, recall, and F1-Score achieved by the LSTM model trained with 100 nodes is 71.62%, 78.26%, 74.23%, and 76.19% respectively. The performance of the POA-LSTM (20) model outperformed the other models in terms of accuracy and precision, while POA-LSTM (15) outperforms other models in terms of recall, and F1-score. Overall, the optimized models perform better than the un-optimized LSTM model in-terms of all the metrics measured. This could be attributed to training the LSTM models with optimal parameters selected by the POA optimizer.

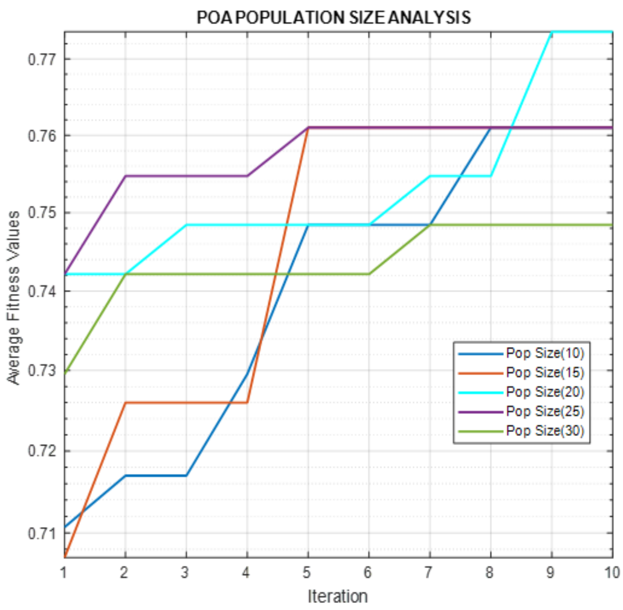


Fig 6: Convergence Curve of POA-LSTM

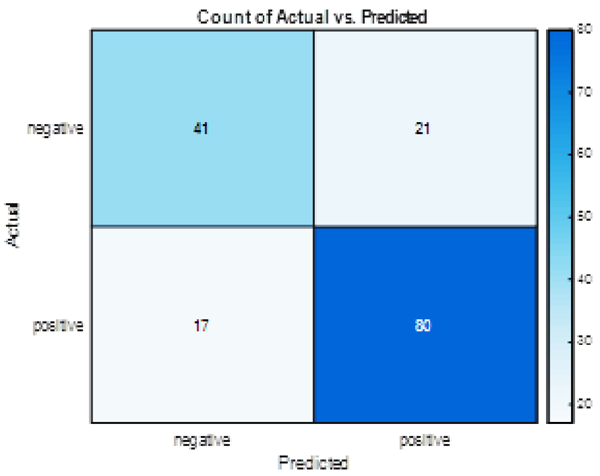
Table II: POA-LSTM optimization

POA Population Size	Optimal Learning Rate	Optimal Hidden Nodes	Optimal Fitness Value
10	0.01	17	0.7610
15	0.02	18	0.7610
20	0.01	37	0.7736
25	0.01	5	0.7610
30	0.02	21	0.7484

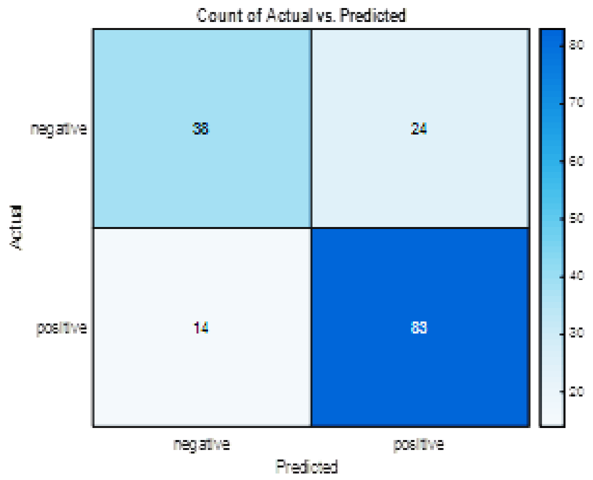
V. CONCLUSION AND FUTURE DIRECTION

In this paper, an optimized sentiment analysis model using Pastoralist Optimization Algorithm (POA) and LSTM deep learning model. The learning rate and node size of the LSTM were optimized and tested on customer's sentiments on a product from Amazon. Then Also, the appropriate population size of the POA algorithm was investigated. The result indicated that the optimized LSTM model performs better in terms of accuracy, precision, recall, and F1-score than the LSTM model. The optimization produces optimal learning rate was found to be 0.02 and an optimal node size of 37 was obtained using POA with 20 pastoralists. This shows that optimized LSTM's can perform better than un-optimized LSTM for sentiment analysis. Furthermore, the POA is capable of being used as a parameter optimizer.

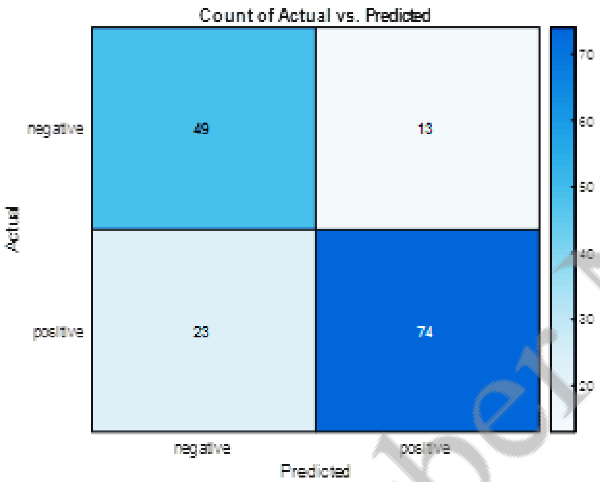
In the future, other sentiment analysis datasets will be explored to evaluate the effectiveness of the developed models. Also, other optimization algorithms such as PSO, ABC, GOA, and BA will be explored to determine their optimization effects on the LSTM model



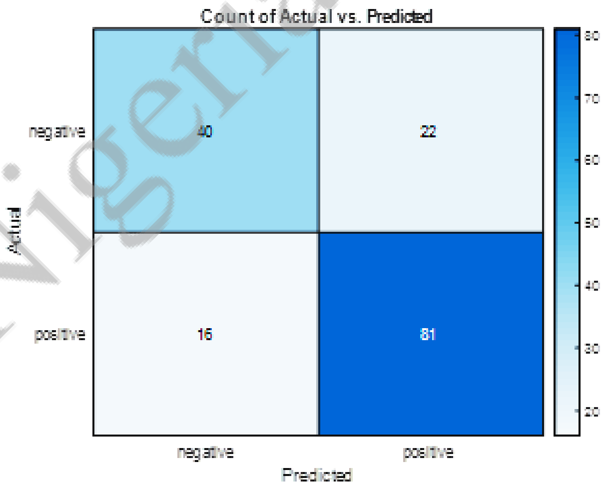
(a) 10 pastoralist



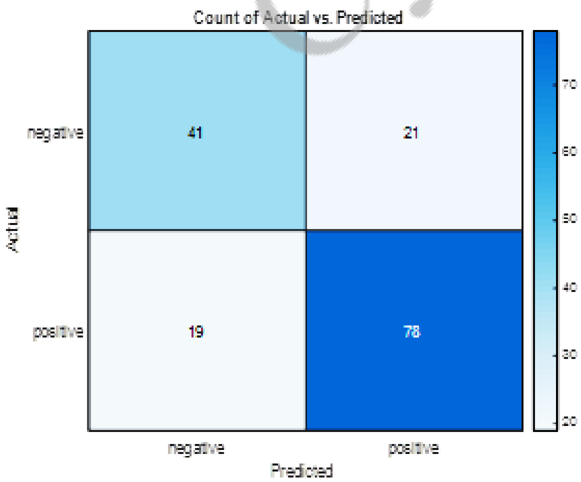
(b) 15 Pastoralist



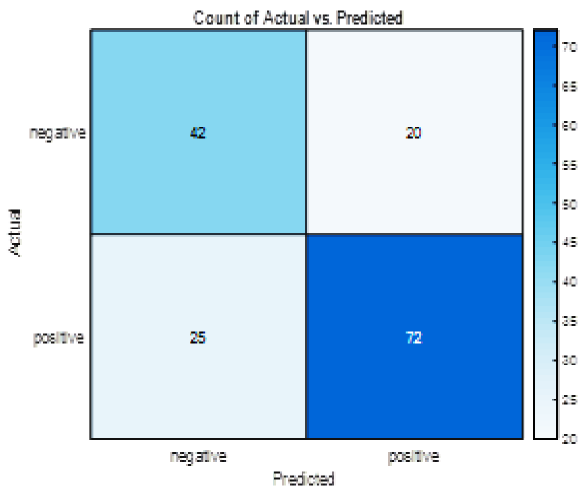
(c) 20 Pastoralist



(d) 25 Pastoralist



(e) 30 Pastoralist



(f) LSTM Model

Fig 7: Confusion matrix of POA-LSTM and LSTM models

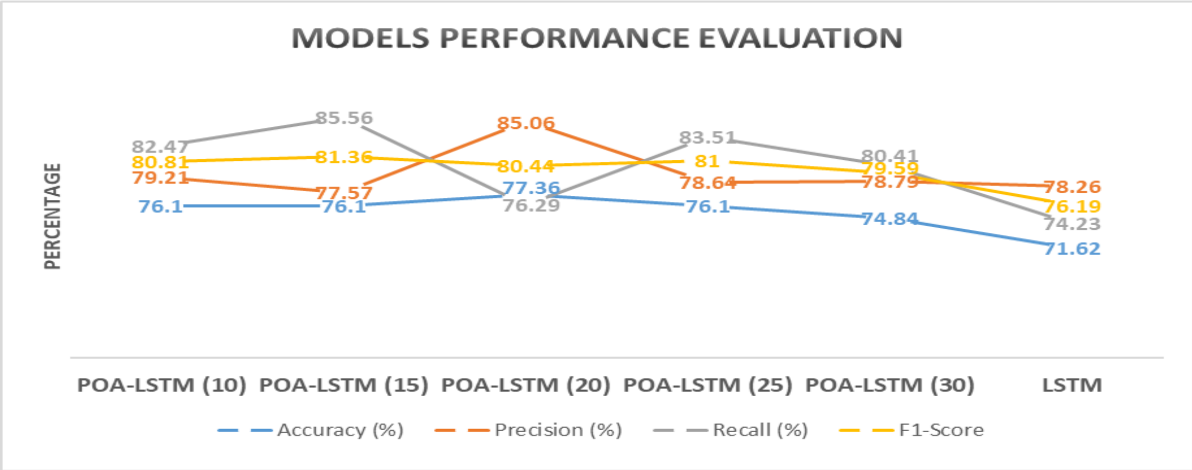


Fig 8: Accuracy, Precision, Recall, and F1-Score graph of all the models evaluated.

REFERENCES

[1] Liu B. Sentiment analysis: mining opinions, sentiments, and emotions. The Cambridge University Press, 2015.

[2] Carlos A. Iglesias and Antonio Moreno. Sentimental Analysis for Social Media, October 2019; Accepted: 19 November 2019; Published: 22 November 2019.

[3] Wang Z, Zhang Y, Lee S, Li S, and Zhou G. A bilingual attention network for code switched emotion prediction. In Proceedings of the International Conference on Computational Linguistics (COLING 2016), 2016.

[4] Rashkin H, Bell E, Choi Y, and Volkova S. Multilingual connotation frames: a case study on social media for targeted sentiment analysis and forecast. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2017), 2017.

[5] Wang Y, Huang M, Zhu X, and Zhao L. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), 2016.

[6] Liu J, Zhang Y. Attention modeling for targeted sentiment. In Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017), 2017.

[7] Akhtar MS, Kumar A, Ghosal D, Ekbal A, and Bhattacharyya P. A multilayer perceptron-based ensemble technique for fine-grained financial sentiment analysis. In Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP 2017), 2017.

[8] Chen H, Sun M, Tu C, Lin Y, and Liu Z. Neural sentiment classification with user and product attention. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), 2016.

[9] Dou ZY. Capturing user and product Information for document level sentiment analysis with deep memory network. In Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP2017), 2017.

[10] Wang J, Fu J, Xu Y, and Mei T. Beyond object recognition: visual sentiment analysis with deep coupled adjective and noun neural networks. In Proceedings of the Internal Joint Conference on Artificial Intelligence.

[11] Guggilla C, Miller T, Gurevych I. CNN-and LSTM-based claim classification in online user comments. In Proceedings of the International Conference on Computational Linguistics (COLING 2016), 2016.

[12] Z. Xiang and U. Gretzel, "Role of social media in online travel information search. T," Researchgate, 2010.

[13] Lee H, Grosse R, Ranganath R, and Ng A.Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the International Conference on Machine Learning(ICML 2009), 2009.

[14] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, and Kuksa P. Natural language processing (almost)from scratch. Journal of Machine Learning Research, 2011.

[15] Kumar, A., &Jaiswal, A. (2019). Systematic literature review of sentiment analysis on Twitter using soft computing techniques. Concurrency ComputatPractExper. https://doi.org/10.1002/cpe.5107.

[16] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Inf Process Manag. 1988;24(5):513–23.

[17] Robertson SE, Walker S. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval. New York: Springer Inc.; 1994. p. 232–41.

[18] Heaton, J., Polson, N., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. Applied Stochastic Models in Business and Industry, 33(1), 3–12.

[19] Kraus, M., &Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. Decision Support Systems, 104, 38–48.

[20] Abdullahi I. M., Muázu M. B., Olaniyi O. M., & Agajo, J, (2019), "An Investigative Parameter Analysis of Pastoralist Optimization Algorithm (Poa): A Novel Metaheuristic Optimization Algorithm", Journal of Science Technology and Education 7(3), pp. 267-272. available at: www.atbuftejoste.com.

[21] I. M. Abdullahi, M. B. Mu'azu, O. M. Olaniyi and J. Agajo, (2019), "A Novel Cultural Evolution-Based Nomadic Pastoralist Optimization Algorithm (NPOA): The Mathematical Models," 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf), Zaria, Nigeria, 2019, pp. 1-7, doi: 10.1109/NigeriaComputConf45974.2019.8949635.

[22] Abdullahi I. M., Mu'azu M. B., Olaniyi O. M. & Agajo J. (2018), "Pastoralist Optimization Algorithm: A Novel Nature-Inspired Metaheuristic Optimization Algorithm". *Proceedings International Conference on Global and Emerging Trends, (ICGET 2018)*, Baze University, Abuja, pp. 101-105.

[23] Zhang L., Wang S., Liu B., (2018), Deep Learning for sentiment Analysis: A Survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), pp. 1-34.

[24] P. Vateekul and T. Koomsubha, A Study of Sentiment Analysis Using Deep Learning Techniques on Thai Twitter Data, 2016.

[25] S. Bengio, L. Deng, H. Larochelle, H. Lee, and R. Salakhutdinov, Guest Editors Introduction: Special Section on Learning Deep Architectures, IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1795-1797, 2013.

[26] M. Usama, W. Xiao, B. Ahmad, J. Wan, M. M. Hassan and A. Alelaiwi (2019), Deep Learning Based Weighted Feature Fusion Approach for Sentiment Analysis, IEEE Access, 7, pp. 140252-140260.

[27] Yang M, Tu W, Wang J, Xu F, and Chen X. Attention-based LSTM fortarget-dependent sentiment classification.In Proceedings of AAAI Conference on Artificial Intelligence (AAAI 2017), 2017.