Journal of Theoretical and Applied Information Technology

15th October 2023. Vol.101. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A LEARNING BASED OPTIMIZED HYBRID MODEL FOR EFFICIENT AND SCALABLE LONG DOCUMENT CLASSIFICATION

AYESHA MARIYAM[1], SK. ALTAF HUSSAIN BASHA[2], S. VISWANADHA RAJU[3]

[1]Research Scholar, Jawaharlal Nehru Technological University, Department of Computer Science and Engineering, Hyderabad, India

[2]Professor, Krishna Chaitanya Institute Of Technology And Sciences, Department of Computer Science and Engineering, Markapur, India

[3]Senior Professor, Jntuh University College Of Engineering, Department of Computer Science and Engineering, Jagtiyal, Karimnagar, India

E-mail: [1]ayesha.mariyam84@gmail.com, [2]althafbashacse@gmail.com, [3]svraju.jntu@gmail.com

ABSTRACT

There is exponential growth of text documents over cloud and Internet based storage infrastructures. There are many applications in the real world that exploit such documents. One such application is document classification which attracts academia and researchers in industry. However, classification of long length documents is found to be not trivial and complex phenomenon. Traditional approaches in deep learning could work on them with full information availability for feature representation leading to deteriorated performance. There were efforts by researchers combining multiple deep learning approaches to realize simplified feature representation of long documents. However, they suffer from issues with hyper parameter optimization and consumption of more resources. To address these problems, in this paper, we proposed a framework known as Long Document Classification Framework (LDCF) which exploits multiple deep learning models appropriately besides enhancing them with hyper-parameter optimization. It is an Artificial Intelligence (AI) enabled approach for scalable solution. Our empirical study has revealed that random search based approach for hyper-parameter optimization could improve the performance of the framework. We proposed an algorithm known as Learning based Optimized Hybrid Model for Long Document Classification (LOHM-LDC) to realize LDCF. It exploits Deep Reinforcement Learning (DRL) to pick text blocks intelligently. Our empirical study with ArXiv dataset, consisting of 1.7 million long documents, has revealed that our hybrid model has potential to outperform existing models with 96.67% accuracy.

Keywords – *Deep Learning, Artificial Intelligence, Hyper-Parameter Optimization, Long Document Classification, Hybrid Deep Learning Model*

1. INTRODUCTION

Classification of long documents is challenging in terms of computational complexity and the time taken for doing so. Traditional approaches that are heuristics based does not provide required convergence and accuracy. Of late, it is observed that learning based approaches do have their potential in processing long documents for classification. Particularly deep learning models, as explored in [1], [2, [3] and [4], are found to have potential to learn from content of documents and perform efficient classification. The significance of machine learning based approaches is that they can handle large volumes of data. As the training data increases, it is observed that the learning based models provide improved convergence in long document classification. Many researchers contributed to the problem of long document classification. Imon et al. [1] explored CNN and RNN models for processing text documents. They considered to work with radiology texts in healthcare industry. The models were explored to know the suitability of them. It is found from their empirical study is that the deep learning models cloud provide acceptable performance.

Kowsari et al. [4] proposed a framework known as HDLTex. It is a hierarchical deep learning framework used for text classification. Their empirical study revealed that the framework

is capable of classifying text documents. Zeshan et al. [9] explored classification of document images using advanced training strategies and deep CNN model with customized configurations. Their research showed that document images can be classified using deep learning. Nasser et al. [12] explored the notion of word embeddings along with CNN for classification text documents. In their research the documents are nothing but resumes containing candidates academic and other information. From the literature it is observed that there were efforts by researchers combining multiple deep learning approaches to realize simplified feature representation of long documents. However, they suffer from issues with hyper parameter optimization and consumption of more resources. Our contributions in this paper are as follows.

1. We proposed a framework known as Long Document Classification Framework (LDCF) which exploits multiple deep learning models appropriately besides enhancing them with hyper-parameter optimization.
2. We proposed an algorithm known as Learning based Optimized Hybrid Model for Long Document Classification (LOHM-LDC) to realize LDCF. It exploits Deep Reinforcement Learning (DRL) to pick text blocks intelligently.
3. We built an application to evaluate performance of LDCF and its underlying algorithm LOHM-LDC. Empirical results revealed the significance of the proposed framework when compared with the state of the art.

The remainder of the paper is structured as follows. Section 2 reviews relevant literature on existing method used for document classification. Section 3 presents our methodology based on deep learning for classification. Section 4 presents our empirical study and its results. Section 5 concludes our work and talk about future possibilities.

2. RELATED WORK

This section reviews literature on existing methods pertaining to long document classification. Imon et al. [1] explored CNN and RNN models for processing text documents. They considered to work with radiology texts in healthcare industry. The models were explored to know the suitability of them. It is found from their empirical study is that the deep learning models cloud provide acceptable performance. Pervez et al. [2] used CNN model along with multi-sized filters to investigate on document classification. Their approach has potential to deal with document level text with NLP and neural networks for efficient classification. Jun et al. [3] proposed a methodology based on recurrent attention learning to obtain local word glimpses. Thus the required intelligence is gained for automatic long document classification. Kowsari et al. [4] proposed a framework known as HDLTex. It is a hierarchical deep learning framework used for text classification. Their empirical study revealed that the framework is capable of classifying text documents. Mohammadreza et al. [5] used CNN for classification of texts. However, their methodology includes predatory conversations found in social media platforms to segregate them into different classes for further intelligence. Andreas et al. [6] combined extreme learning machines and document image classification in the real time in order to have a learning based system for automatic text classification. Their study showed the importance of CNN and extreme learning machines towards text classification. Chris et al. [7 used CNN towards classification of fonts in the processes of analysing text documents. Shaobo et al. [8] proposed a deep learning framework based on CNN with word embeddings. Their framework is named as DeepPatent as they considered classification of patent documents. Zeshan et al. [9] explored classification of document images using advanced training strategies and deep CNN model with customized configurations. Their research showed that document images can be classified using deep learning. Nix and Zhang [10] proposed a methodology using deep neural networks for automatic classification of Android applications in terms of benign and malware affected ones. Mahalakshmi and Fatima [11] proposed a methodology for intelligent retrieval of documents using both text and images. Their methodology is based on CNN models. Nasser et al. [12] explored the notion of word embeddings along with CNN for classification text documents. In their research the documents are nothing but resumes containing candidates academic and other information. In [13] convolutional and recurrent models are combined to classify sentences while in [14] sentiment classification is made on given text documents using deep learning approach. In [15], documents are subjected to multi-class classification using deep learning models. Other important research contributions include legal document classification [16], hybrid model for text classification [17],

hybrid sentiment classification model [18], sentiment classification with linear and non-linear models [19] and CNN based problem classification [20]. From the literature it is observed that there were efforts by researchers combining multiple deep learning approaches to realize simplified feature representation of long documents. However, they suffer from issues with hyper parameter optimization and consumption of more resources.

3. PROPOSED FRAMEWORK FOR LONG DOCUMENT CLASSIFICATION

We proposed a framework known as Long Document Classification Framework (LDCF), as shown in Figure 1, which exploits multiple deep learning models appropriately besides enhancing them with hyper-parameter optimization. It is an Artificial Intelligence (AI) enabled approach for scalable solution.

its consideration to deal with efficient sampling which reduces space and time complexity. Our framework is influenced by the ideas presented in [25][26].

Our framework does not take complete information in given documents to reduce complexity in its modus operandi. Instead, it has efficient sub-sampling that effectively represents content of a given document D. As a result, the model is optimized in its functionality. Moreover, our methodology is designed to work with long length documents collected from ArXiv dataset [26] which has large volumes of research articles. Our framework performs random sub-sampling to represent given document, generates word vectors and extracts features using CNN model. Since we use random sampling and the samples are expected to reflect the essence of D. However, in reality, it may not be so.

To overcome this problem, we introduced

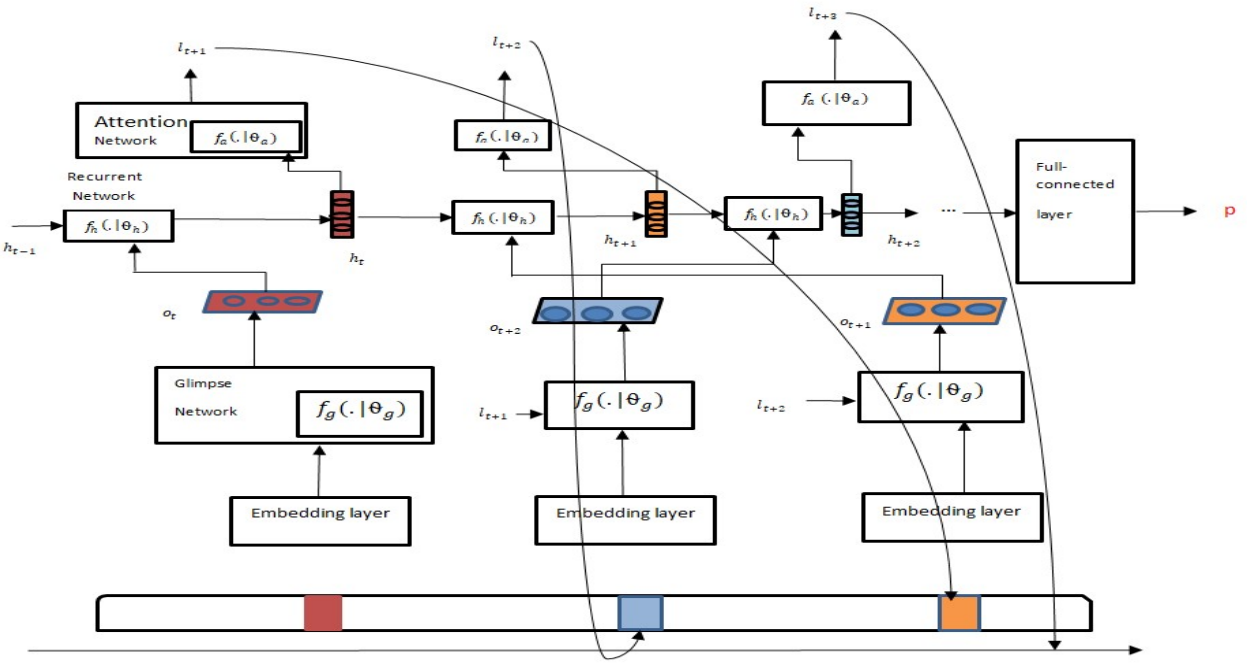

Figure 1: *Proposed framework based on hybrid deep learning models*

3.1 Our Framework

Our framework is the result of empirical study that combines deep learning models such as CNN, RNN and LSTM besides attention mechanism, glimpse network and our proposed hyper-parameter optimization method to classify long length documents in a scalable and efficient fashion. Our research assumes significance due to a smart agent based approach using Reinforcement Learning (RL), as illustrated in Figure 2, where agent picks best blocks of D. In other words, the agent is used to control the word blocks considered for processing. The RL has mechanism to choose best word blocks based on reward gained for its actions.

As presented in Figure 1, different kinds of networks such as CNN, RNN with attention

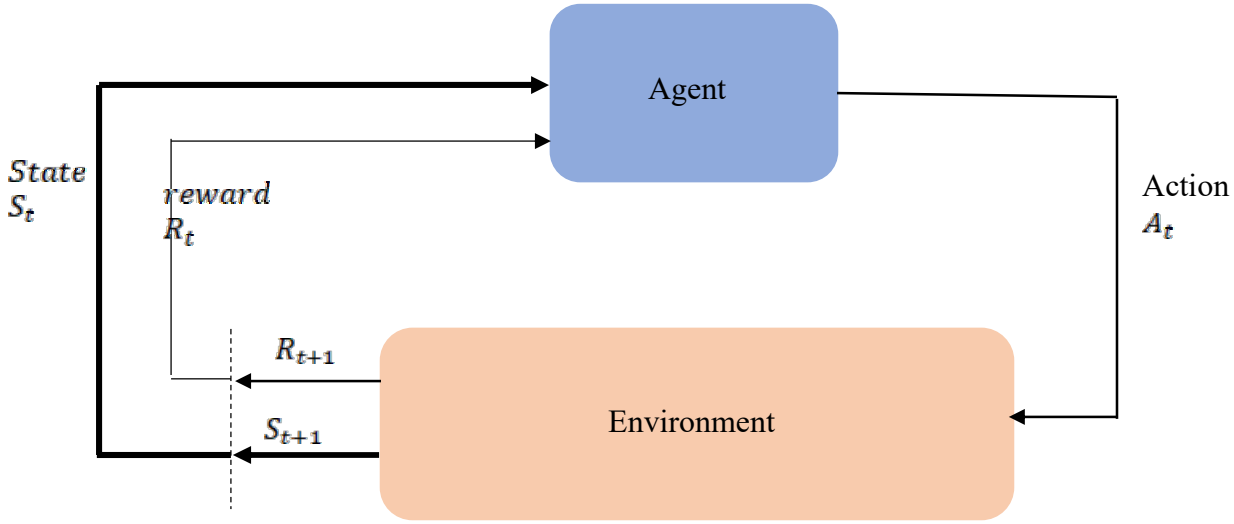

Figure 2: *Illustrates reinforcement learning process*

with RL. Instead of using simple random selection of word blocks from D, our method uses RL to determine best representative blocks from the document D. Thus the selected blocks reflect the characteristics of given document. The recurrent attention approach has its importance in improving classification performance in our framework. Local CNN feature extraction is done by glimpse network while the RNN module involved in our framework is responsible for feature aggregation. And the process of location predication for sampling is carried out by the attention network.

3.2 Reinforcement Learning

In the proposed framework, RL plays an important role in efficient sub-sampling of the documents. It is based on a smart agent that takes inputs from environment in the form of state and comes up with an action. Based on the action, the environment gives reward which is based on the correct representation of given word block from D. Environment in the proposed system reflects the mechanism that has intelligence to find how best a sub-sampled part of D can effectively represent D to some extent. With number of sub-sampled word blocks, it ensures that they provide best representation of D instead of taking entire content from D for processing. There is an action space and reward space. Reward is denoted as R_t, state is denoted as S_t and action is denoted as A_t in a given time step. There is an iterative process in which the agent eventually finds the best representative word blocks.

Table 1: Notations used in the proposed long document classification model

Notation	Meaning
$f_h(\cdot\|\theta_h)$	Aggregation of features extracted by CNN
$f_a(\cdot\|\theta_a)$	Predication associated with sampling location
$f_g(\cdot\|\theta_g)$	Denotes local feature extraction by CNN
l_0	Value associated with random location
r_t	Denotes reward in RL process
θ_h	Denotes recurrent network
θ_a	Denotes attention network
θ_g	Denotes glimpse network
D	Denotes a document
maxIter	Value reflecting maximum iterations
T	Denotes number of glimpse
R	Denotes value of cumulative return
$p(S_{1:T}\|\theta)$	Policy for optimization
γ	Denotes value for decaying coefficient

With the proposed framework, the given set of documents are subjected to efficient classification. The dataset obtained from [26] has ground truth in order to find the correctness of predictions made by our framework. Our framework is realized by proposing an algorithm known as Learning based Optimized Hybrid Model for Long Document Classification (LOHM-LDC).

3.3 Our Algorithm

We proposed an algorithm known as known as Learning based Optimized Hybrid Model for Long Document Classification (LOHM-LDC). It is designed to exploit the proposed framework and the underlying mechanisms such as RL and hyper-parameter optimization.

Algorithm: Optimized Hybrid Model for Long Document Classification (LOHM-LDC).

Inputs:
Document D
Attention network parameter θ_a
Recurrent network parameter θ_h
Glimpse network parameter θ_g
Maximum number of glimpse T
Maximum number of iterations maxIter
Initial hyper-parameters λ
Output:
Classification result

Begin
λ ← TuneWithRandomSearch(D)
For each iteration in maxIter
  For each glimpse t in T
    wordBlock←ExtractWords(l_t)
    wordVector←ObtainWordVector(wordBlock)
    o_t← GetFeatures($f_g(\cdot|\theta_g)$)
    h_t←GetNewHiddenState($f_h(\cdot|\theta_h), o_t$)
    l_{t+1}←PredictNxtLoc($f_a(\cdot|\theta_a), h_t$)
  End For
  label←PredictLabel($f_h(\cdot|\theta_h), h_{t+1}$)
  IF label is correct Then
    Return 1 as reward
  Else
    Return 0 as reward
  End If
  Update θ_a using RL
  Update θ_h and θ_g using back propagation
 End For
End

Algorithm 1: *known as Learning based Optimized Hybrid Model for Long Document Classification*

Algorithm 1 takes document D, attention network parameter θ_a, recurrent network parameter θ_h, glimpse network parameter θ_g, maximum number of glimpse T, maximum number of iterations maxIter and initial hyper-parameters λ as inputs and gives classification result as output. The algorithm has provision to search for best hyper-parameters and update λ. For each glimpse step t, $f_a(\cdot|\theta_a)$ predicts l_t. Based on the location word block is extracted. Then $f_g(\cdot|\theta_g)$ is used to obtain CNN features and such features are aggregated using prior hidden state associated with $f_h(\cdot|\theta_h)$. Then $f_a(\cdot|\theta_a)$ finds next location required by next glimpse. Once all glimpses are completed, the algorithm finally performs document classification. The $f_a(\cdot|\theta_a)$ is trained using RL as the l_t and θ_a do not have a differential relation. Based on the class label prediction, reward 1 is returned or else 0 is returned. Then $f_a(\cdot|\theta_a)$ gets optimized as expressed in Eq. 1.

$$J(\theta) = E_{p(S_{1:T}|\theta)}[\sum_{t=1}^{T} r_t] = E_{p(S_{1:T}|\theta)}[R] \quad (1)$$

Here the optimized policy is denoted as $p(S_{1:T}|\theta)$ which has potential to predict next position and the cumulative reward is computed as expressed in Eq. 2.

$$R = \sum_{t=1}^{t} \gamma^{t-1} r_t \quad (2)$$

Where r_t is the reward in each step while the cumulative return value is denoted as R reflecting sum of rewards. There is another value used in the computation which is known as decaying coefficient γ. Table 1 shows notations associated with the proposed framework and underlying algorithm.

3.4 Hyper-Parameter Optimization using Random Search

Any learning algorithm *A*, in AI domain, strives to optimize loss L(x; *f*) where *f* is the function to minimize loss while x denotes the samples for learning. It is associated with the distribution of ground truth G_x. In other words, a training set $x^{(train)}$ and function *f* are used to minimize loss. Often, *A* generates *f* by optimization of criterion linked to training in relation with parameters denoted as θ. Moreover, *A* can have its own hyper parameters denoted as λ that have influence on learning process. Given a training set $x^{(train)}$, f = $A_\lambda(x^{(train)})$ is the function reflecting the algorithm along with its parameters A_λ. By carefully choosing λ, it is possible to minimize, $E_{x\sim G_x}$[L(x; $A_\lambda(x^{(train)})$)] which denotes generalization error. Therefore, it is understood that A has computations for optimization of hyper-parameters. To state it differently identification of good value for λ (hyper-parameters) is essential for hyper-parameter optimization. This optimization problem is expressed as in Eq. 3. Table 2 shows notations used in the hyper-parameter optimization approach.

$$\lambda^{(*)} = \text{argmin}_{\lambda \in \Lambda} E_{x \sim G_x}[L(x; A_\lambda(X^{(train)}))] \quad (3)$$

Many real world algorithms generally do not have such optimization as indicted in Eq. 3. Though the process of hyper-parameter optimization relies on underlying dataset, striving for optimization has is influence on accuracy of A. Provided G_x, finding best values for optimization is a difficult problem. In this regard, grid search for hyper-parameter optimization is a poor choice. Therefore, we focus on random search which has potential to identify best values for λ. By incorporating cross validation technique, it is possible to realize hyper-parameter optimization.

$$\lambda^{(*)} \approx \text{argmin}_{\lambda \in \Lambda} \text{mean}_{x \in X^{(valid)}} L(x; A_\lambda(X^{(train)})) \quad (4)$$

$$\equiv \text{argmin}_{\lambda \in \Lambda} \Psi(\lambda) \quad (5)$$

$$\approx \text{argmin}_{\lambda \in \{\lambda^{(1)}, \ldots \ldots \lambda^{(S)}\}} \Psi(\lambda) \equiv \hat{\lambda} \quad (6)$$

The process of hyper-parameter optimization reflects in Eq. 4 to Eq. 6. In Eq. 5, the optimization problem is considered as response function Ψ. It is achieved by minimizing Ψ(λ) linked to λ ∈ Λ. The optimization strategy is finally made explicit as expressed in Eq. 6. In the optimization process, it is crucial to have set of trials denoted as $\{\lambda^{(1)}, \ldots \ldots \lambda^{(S)}\}$. Though there are many studies based on manual search and grid search, as explored in [21], [22], [23] and [24], our empirical study showed that random search is more efficient when there is high dimensional space.

Table 2: Notations used in hyper-parameter optimization

Notation	Description
$X^{(valid)}$	Denotes validation set
$Z^{(s)}$	Denotes value for validation score
w_s	Denotes value for estimated weight
$\mathbb{V}^{(valid)}(\lambda^{(s)})$	Denotes value for squared standard error
μ_z	Denotes mean value of random variable
σ_z^2	Denotes value for standard error
$\{\lambda^{(1)}, \ldots \ldots \lambda^{(S)}\}$	Denotes set of trials
f	Denotes optimization function
S	An integer value
z	Denotes a random variable
λ	Denotes optimized values for hyper-parameters
Ψ	Denotes a response function linked to hyper-parameter optimization
$\Psi(\lambda^{(i)})$	Denotes density value for given validation set score
$\mathbb{V}$	Variance estimated

It is often observed that with different λ, validation error and test error differ. The uncertainty associated with choice of λ can help in estimating accuracy of test set. In order to do so, it is important to find difference between estimates denoted as $\Psi^{(valid)} = \Psi$ and $\Psi^{(test)}$. It is based on validation and test sets as expressed in Eq. 7 and Eq. 8 respectively.

$$\Psi^{(valid)}(\lambda) = mean_{x \in X^{(valid)}} l(x; A_\lambda(X^{(train)})) \quad (7)$$

$$\Psi^{(test)}(\lambda) = mean_{x \in X^{(test)}} l(x; A_\lambda(X^{(train)})) \quad (8)$$

In the same fashion, it is desired to find Bernoulli variance on those sets of data as expressed in Eq. 9 and Eq. 10.

$$\mathbb{V}^{(valid)}(\lambda) = \frac{\Psi^{(valid)}(\lambda)(1-\Psi^{(valid)}(\lambda))}{|x^{(valid)}|-1} \quad (9)$$

$$\mathbb{V}^{(test)}(\lambda) = \frac{\Psi^{(test)}(\lambda)(1-\Psi^{(test)}(\lambda))}{|x^{(test)}|-1} \quad (10)$$

The variance estimation generally varies with different loss functions. In practice reporting $\lambda^{(s)}$ and $\Psi^{(test)}(\lambda^{(s)})$ that reduces $\Psi^{(valid)}(\lambda^{(s)})$ is preferred. Nevertheless, having many trials that reflect near optimal validation causes reporting test score issue. To address this problem, a weighted average of all scores is used. For specific $\lambda^{(s)}$ uncertainty associated with $X^{(valid)}$ which is a subset of G_x, a random variable z is used to get best test-set score among $\lambda^{(1)}, \ldots \ldots \lambda^{(S)}$. Gaussian mixture mode is employed to have z value associated with S components with variance $\sigma_s^2 = \mathbb{V}^{(test)}(\lambda^{(s)})$, means $\mu_s = \Psi^{(test)}(\lambda^{(s)})$ and weight $w_s = P(Z^{(s)} < Z^{(s')}, \forall s' \neq s)$ where $Z^{(i)}$ is computed as in Eq. 11.

$$Z^{(i)} \sim \mathcal{N}(\Psi^{(valid)}(\lambda^{(i)}), \mathbb{V}^{(valid)}(\lambda^{(i)})) \quad (11)$$

The mean and standard error computations are made as in Eq. 12 and Eq. 13.

$$\mu_z = \sum_{s=1}^{S} w_s \mu_s \quad (12)$$

$$\sigma_z^2 = \sum_{s=1}^{S} w_s(\mu_s^2 + \sigma_s^2) - \mu_z^2 \quad (13)$$

In practice, weights w_s is estimated through empirical study. When compared to traditional approach in hyper-parameter optimization, our method is found superior in finding best values that influence deep learning or neural network models. The proposed algorithm exploits this optimization technique. With hyper-parameter optimization, the values associated with λ are as follows. Learning rate is 0.001 and later it is adjusted to 0.0001. The batch size is optimized to 64 while convolutional

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

kernels used are 128 with sizes such as 3, 4, and 5. Dropout rate is optimized to 0.5.

4. RESULTS AND DISCUSSION

This section presents our experimental results reflecting the performance of the proposed algorithm. Dataset used for this study is collected from [26] which has volumes of data with long length documents. Accuracy is performance metric used to evaluate performance of the proposed model and compared with state of the art.

Window Size	Accuracy (%)			
	CNN Feature Aggregation	CNN with LSTM	CNN with Recurrent Attention Model	Proposed
10	86.37	83.42	89.84	90.15
20	86.53	86.05	90.15	92.36
50	86.98	86.71	91.67	93.25
100	87.03	87.23	91.89	93.87
200	87.21	88.06	92.23	94.39
400	87.56	88.86	93.45	95.23
500	87.95	89.11	93.87	95.65

Table 3: *Performance comparison with 200 words. As presented in Table 3, accuracy of different models for document classification are provided when number of words is 200.*

Window Size	Accuracy (%)			
	CNN Feature Aggregation	CNN with LSTM	CNN with Recurrent Attention Model	Proposed
10	87.48	86.28	90.05	92.36
20	90.11	88.41	92.22	93.63
50	90.33	89.08	93.08	94.32
100	90.60	89.08	93.56	94.68
200	92.08	90.08	93.67	94.98
400	92.12	90.12	93.53	95.02
500	92.78	91.35	93.95	95.23

Table 4: *Performance comparison with 400 words.As presented in Table 4, accuracy of different models for document classification are provided when number of words is 400.*

Window Size	Accuracy (%)			
	CNN Feature Aggregation	CNN with LSTM	CNN with Recurrent Attention Model	Proposed
10	88.36	86.66	90.11	91.26
20	91.21	89.59	92.38	93.25
50	91.88	89.92	93.68	94.36
100	92.16	90.23	94.24	95.24
200	93.35	90.51	94.11	95.55
400	93.68	90.78	94.54	95.78
500	93.89	91.05	94.89	96.01

Table 5: *Performance comparison with 600 words. As presented in Table 5, accuracy of different models for document classification are provided when number of words is 600.*

Window Size	Accuracy (%)			
	CNN Feature Aggregation	CNN with LSTM	CNN with Recurrent Attention Model	Proposed
10	89.14	89.44	92.36	93.15
20	92.13	90.28	92.53	93.55
50	92.32	90.37	93.86	94.34
100	92.63	90.56	94.36	95.28
200	93.38	90.82	94.41	95.45
400	93.41	90.94	93.87	95.68
500	93.56	91.21	94.65	95.95

Table 6: *Performance comparison with 800 words. As presented in Table 6, accuracy of different models for document classification are provided when number of words is 800.*

Window Size	Accuracy (%)			
	CNN Feature Aggregation	CNN with LSTM	CNN with Recurrent Attention Model	Proposed
10	90.26	89.48	93.17	94.24
20	92.21	90.36	93.21	94.69
50	92.59	90.61	94.56	95.24
100	93.78	91.77	94.68	95.67
200	93.89	91.96	95.48	96.12
400	93.97	92.15	95.60	96.45
500	94.02	92.34	94.73	96.67

Table 7: *Performance comparison with 1000 words. As presented in Table 7, accuracy of different models for document classification are provided when number of words is 1000.*

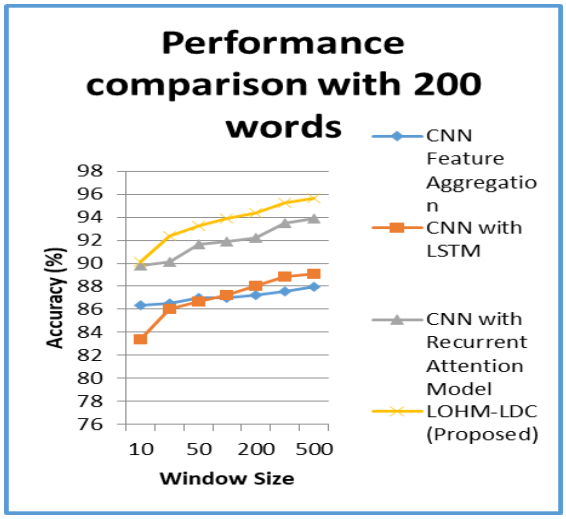

Figure 3: *Performance evaluation with 200 words.*

As presented in Figure 3, it is clearly observed that both window size and number of words play role in performance of the document classification models. When number of words is 200, the accuracy of the models is increased as the window size increases from 10 to 500 gradually. When the window size is 10 CNN feature aggregation showed 86.37% accuracy, CNN with LSTM model 83.42%, CNN with recurrent attention model 89.84% and the proposed model exhibits 90.15%. With window size set to highest value that is 500, CNN feature aggregation achieved 87.95% accuracy, CNN with LSTM 89.11%, CNN with recurrent attention model 93.87% and the proposed model exhibits highest performance with 95.65%.

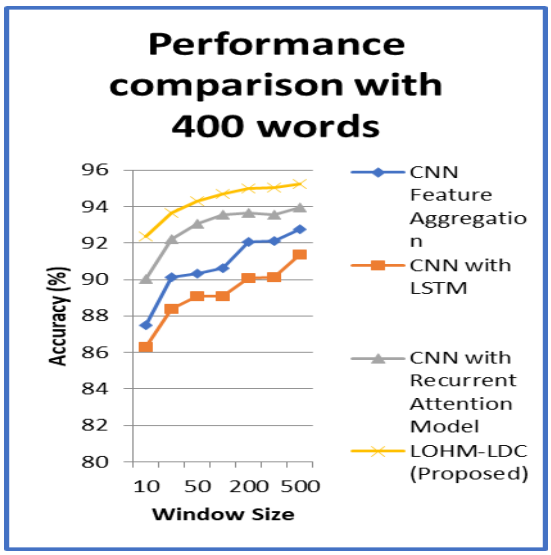

Figure 4: *Performance evaluation with 400 words.*

As presented in Figure 4, it is clearly observed that both window size and number of words play role in performance of the document classification models. When number of words is 400, the accuracy of the models is increased as the window size increases from 10 to 500 gradually. When the window size is 10 CNN feature aggregation showed 87.48% accuracy, CNN with LSTM model 86.28%, CNN with recurrent attention model 90.05% and the proposed model exhibits 92.36%. With window size set to highest value that is 500, CNN feature aggregation achieved 92.78% accuracy, CNN with LSTM 91.35%, CNN with recurrent attention el 93.95% and the proposed model exhibits highest performance with 95.23%.

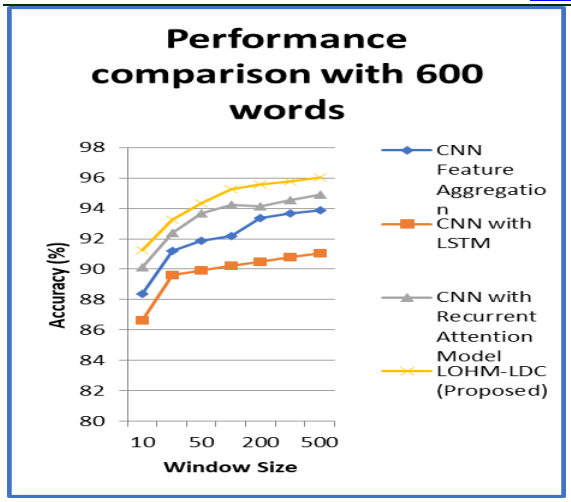

Figure 5: *Performance evaluation with 600 words.*

As presented in Figure 5, it is clearly observed that both window size and number of words play role in performance of the document classification models. When number of words is 600, the accuracy of the models is increased as the window size increases from 10 to 500 gradually. When the window size is 10 CNN feature aggregation showed 88.36% accuracy, CNN with LSTM model 86.66%, CNN with recurrent attention model 90.11% and the proposed model exhibits 91.26%. With window size set to highest value that is 500, CNN feature aggregation achieved 93.89% accuracy, CNN with LSTM 91.05%, CNN with recurrent attention model 94.89% and the proposed model exhibits highest performance with 96.01%.

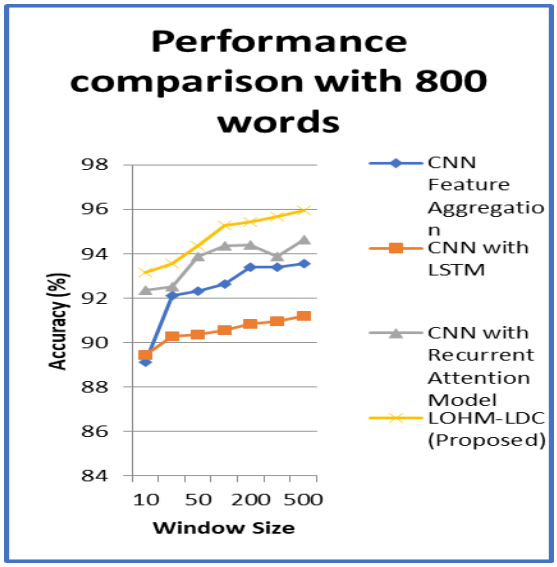

Figure 6: *Performance evaluation with 800 words.*

As presented in Figure 6, it is clearly observed that both window size and number of words play role in performance of the document classification models. When number of words is 800, the accuracy of the models is increased as the window size increases from 10 to 500 gradually. When the window size is 10 CNN feature aggregation showed 89.14% accuracy, CNN with LSTM model 89.66%, CNN with recurrent attention model 82.96% and the proposed model exhibits 93.15%. With window size set to highest value that is 500, CNN feature aggregation achieved 93.56% accuracy, CNN with LSTM 91.21%, CNN with recurrent attention model 94.65% and the proposed model exhibits highest performance with 95.95%.

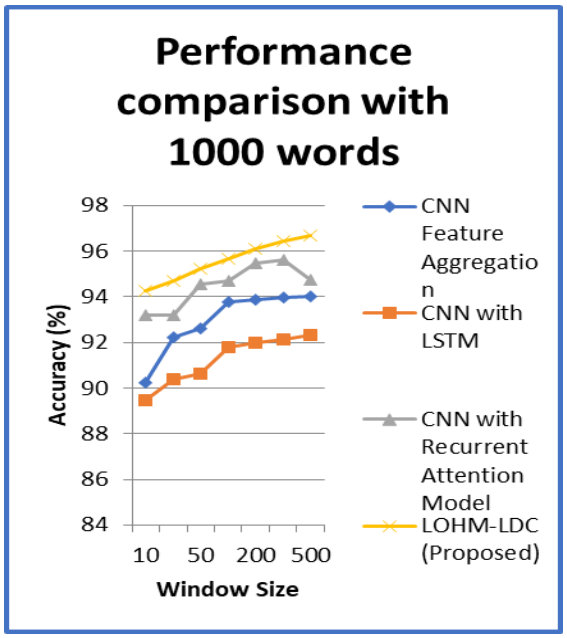

Figure 7: *Performance evaluation with 1000 words.*

As presented in Figure 7, it is clearly observed that both window size and number of words play role in performance of the document classification models. When number of words is 1000, the accuracy of the models is increased as the window size increases from 10 to 500 gradually. When the window size is 10 CNN feature aggregation showed 90.26% accuracy, CNN with LSTM model 89.48%, CNN with recurrent attention model 93.17% and the proposed model exhibits 94.24%. With window size set to highest value that is 500, CNN feature aggregation achieved 94.02% accuracy, CNN with LSTM 92.34%, CNN with recurrent attention model 94.73% and the proposed model exhibits highest performance with 96.67%.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a framework known as Long Document Classification Framework (LDCF) which exploits multiple deep learning models appropriately besides enhancing them with hyper-parameter optimization. It is an Artificial Intelligence (AI) enabled approach for scalable solution. Our framework exploits CNN, recurrent attention mechanism besides using RL agent to optimize sub-sampling leading to best characterization of long documents without considering content of entire document. Our empirical study has revealed that random search based approach for hyper-parameter optimization could improve the performance of the framework. Therefore, we could improve performance in obtaining better parameters with hyper-parameter optimization. We proposed an algorithm known as Learning based Optimized Hybrid Model for Long Document Classification (LOHM-LDC) to realize LDCF. It exploits Deep Reinforcement Learning (DRL) to pick text blocks intelligently. Our empirical study with ArXiv dataset, consisting of 1.7 million long documents, has revealed that our hybrid model has potential to outperform existing models with 96.67% accuracy. In future, we intend to experiment our framework with Bayesian optimization method, another parameter-optimization approach, and compare it with results of random search based optimization method incorporate in this paper.

REFERENCES:

[1] Banerjee, Imon; Ling, Yuan; Chen, Matthew C.; Hasan, Sadid A.; Langlotz, Curtis P.; Moradzadeh, Nathaniel; Chapman, Brian; Amrhein, Timothy; Mong, David; Rubin, Daniel L.; Farri, Oladimeji and Lungren, Matthew P. (2018). Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. Artificial Intelligence in Medicine, S0933365717306255

[2] Akhter, Muhammad Pervez; Jiangbin, Zheng; Naqvi, Irfan Raza; Abdelmajeed, Mohammed; Mehmood, Atif and Sadiq, Muhammad Tariq (2020). Document-level Text Classification using Single-layer Multisize Filters Convolutional Neural Network. IEEE Access, 1–1.

[3] He, Jun; Wang, Liqun; Liu, Liu; Feng, Jiao and Wu, Hao (2019). Long Document Classification From Local Word Glimpses via Recurrent Attention Learning. IEEE Access, 7, 40707–40718.

[4] Kamran Kowsari, Donald E. Brown§, Mojtaba Heidarysafa and Kiana Jafari Meimandi. (2017). HDLTex: Hierarchical Deep Learning for Text Classification. IEEE., pp.364-371.

[5] Ebrahimi, Mohammadreza; Suen, Ching Y. and Ormandjieva, Olga (2016). Detecting predatory conversations in social media by deep Convolutional Neural Networks. Digital Investigation, 18, 33–49.

[6] Kolsch, Andreas; Afzal, Muhammad Zeshan; Ebbecke, Markus and Liwicki, Marcus (2017). 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) - Real-Time Document Image Classification Using Deep CNN and Extreme Learning Machines. , 1318–1323.

[7] Tensmeyer, Chris; Saunders, Daniel and Martinez, Tony (2017). 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) - Convolutional Neural Networks for Font Classification. , 985–990.

[8] Li, Shaobo; Hu, Jie; Cui, Yuxin and Hu, Jianjun (2018). DeepPatent: patent classification with convolutional neural networks and word embedding. Scientometrics.

[9] Afzal, Muhammad Zeshan; Kolsch, Andreas; Ahmed, Sheraz and Liwicki, Marcus (2017). 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) - Cutting the Error by Half: Investigation of Very Deep CNN and Advanced Training Strategies for Document Image Classification,883–888.

[10] Robin Nix and Jian Zhang. (2017). Classification of Android Apps and Malware Using Deep Neural Networks. IEEE., pp.1871-1878.

[11] P. Mahalakshmi and N. Sabiyath Fatima; (2021). Ensembling of text and images using Deep Convolutional Neural Networks for Intelligent Information Retrieval . Wireless Personal Communications.

[12] Shabna Nasser, Sreejith C and Irshad M. (2018). Convolutional Neural Network with Word Embedding Based Approach for Resume Classification. IEEE., pp.1-6.

[13] Hassan, Abdalraouf; Mahmood and Ausif (2018). Convolutional Recurrent Deep Learning Model for Sentence Classification. IEEE Access, 1–1.

[14] Abdalraouf Hassan and Ausif Mahmood. (2017). Deep Learning Approach for Sentiment Analysis of Short Texts. IEEE., pp.705-710.
[15] Lenc, L., & Král, P. (2018). Deep Neural Networks for Czech Multi-label Document Classification. Lecture Notes in Computer Science, 460–471.
[16] Wei, Fusheng; Qin, Han; Ye, Shi and Zhao, Haozhen (2018). IEEE International Attention-based bidirectional gated recurrent Neural network and two-dimensional for document-level sentiment classification. Neurocomputing, S092523121931272X–.
[19] Lee, Gichag; Jeong, Jaey un; Seo, Seungwan; Kim, CzangYeob and Kan g, Pilsung (2018). Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network. Knowledge-Based Systems, S0950705118301710–.
[20] Zhong, Botao; Xing, Xuejiao; Love, Peter; Wang, Xu and Luo, Hanbin (2019). Convolutional neural network: Deep learning-based classification of building quality problems. Advanced Engineering Informatics, 40, 46–57.
[21] Nelder, J. A.; Mead, R. (1965). A Simplex Method for Function Minimization. The Computer Journal, 7(4), 308–313. http://doi:10.1093/comjnl/7.4.308.
[22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. Science, 220 (4598):671–680, 1983.
[23] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. Advances in Optimization and Numerical Analysis, pages 51– 67, 1994. http://doi:10.1007/978-94-015-8330-5_4.
[24] T. Weise. Global Optimization Algorithms - Theory and Application. Self-Published, second edition, 2009. http://www.it-weise.de/.
[25] Mnih, V.; Heess, N.; Graves, A. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2204–2212.
[26]Dulhare, U.N., Mubeen, A. ,"Detection and Classification of Rheumatoid Nodule using Deep Learning Models", Procedia Computer Science, 2022, 218,pp. 2401–2410
[27]ArXiv dataset. Collected from https://www.kaggle.com/datasets/1b6883fb66c5e7f67c697c2547022cc04c9ee98c3742f9a4d6c671b4f4eda591
Conference on Big Data (Big Data) - Empirical Study of Deep Learning for Text Classification in Legal Document Review, 3317–3320.
[17] Zheng, Jin and Zheng, Limin (2019). A Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model for Text Classification. IEEE Access, 1–1.
[18] Liu, Fagui; Zheng, Jingzhong; Zheng, Lailei and Chen, Cheng (2019). Combining