International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

HMBO-LDC: A Hybrid Model Employing Reinforcement Learning with Bayesian Optimization for Long Document Classification

Ayesha Mariyam
Assistant Professor,Computer Science and Artificial Intelligence,
Muffakham Jah College of Engineering and Technology,
Hyderabad, India
ayesha.mariyam@mjcollege.ac.in

Sk. Altaf Hussain Basha
Professor and Head, Computer Science and Engineering,
Krishna Chaitanya Institute of Technology and Sciences,
Prakasam, India
althafbashacse@gmail.com

S Viswanadha Raju
Senior Professor, Computer Science and Engineering,
JNTUH University College of Engieering Jagtial,
Nachupally,India
svraju.jntu@gmail.com

Abstract— With the emergence of distributed computing platforms and cloud-big data eco-system, there has been increased growth of textual documents stored in cloud infrastructure. It is observed that most of the documents happened to be lengthy. Automatic classification of such documents is made possible with deep learning models. However, it is observed that deep learning models like CNN and its variants do have many hyper parameters that are to be optimized in order to leverage classification performance. The existing optimization methods based on random search are found to have suboptimal performance when compared with Bayesian Optimization (BO). However, BO has issues pertaining to choice of covariance function, time consumption and support for multi-core parallelism. To address these limitations, we proposed an algorithm named Enhanced Bayesian Optimization (EBO) designed to optimize hyper parameter tuning. We also proposed another algorithm known as Hybrid Model with Bayesian Optimization for Long Document Classification (HMBO-LDC). The latter invokes the former appropriately in order to improve parameter optimization of the proposed hybrid model prior to performing long document classification. HMBO-LDC is evaluated and compared against existing models such as CNN feature aggregation method, CNN with LSTM and CNN with recurrent attention model. Experimental results revealed that HMBO-LDC outperforms other methods with highest classification accuracy 98.76%.

Keywords- Long Document Classification, Enhanced Bayesian Optimization, Hybrid Deep Learning Model, Hyperparameter Tuning.

I. Introduction

Classification textual documents has its applications in the real world. The applications are useful in the domains such as healthcare, medicine, legal, manufacturing and distribution to mention few. Classification of documents are done traditionally based on the content comparison and similarity. However, traditional approaches are not adaptive to the exponential growth in text documents in cloud infrastructure. The existing heuristics based approaches are not efficient for large scale classification of text documents [1]. Moreover, the documents that are lengthy or long document classification has number of problems. For instance, it becomes very complex and time taking to classify large volume of long documents. Therefore, machine learning (ML) and deep learning based approaches are found to be scalable and suitable for classification of such documents. It is observed that learning based phenomena has ware withal to classify such documents more efficiently.

Literature has revealed the significance of long document classification and methods besides importance of BO. Otter et al. [6] investigated on NLP and deep learning models. With their empirical study, they found that deep learning models are feasible for processing textual documents using NLP. They used a transformer model towards processing text documents. Medvedeva et al. [9] proposed a methodology for analysing European court documents in order to predict court decisions based on the historical data using ML models. It performs classification of legal text documents using SVM. Khanday et al. [12] proposed a methodology for automatic classification of Covid-19 related text documents to classify patients into Covid-19 disease or no disease. Their methodology supports multi-class classification considering Covid-19 and other flu related diseases. Ngiam and Khor [15] investigated on the healthcare documents in order to discovery different interesting trends or patterns from such data. With regard to BO, there are important observations.

Victoria and Maragatham [22] explored BO approach for automatic optimization of parameters pertaining to deep learning models. They explored different values for parameters and then optimized to have best parameters. Their research revealed that BO has its impact on accuracy of underlying CNN model. Wang et al. [25] employed BO in deep learning model used for automatic crop disease prediction. Particularly, they explored rice diseases using BO along with deep learning. In future then intend to integrate soil, weather and location to improving smart farming possibilities. Joy et al. [29] introduction a concept known as batch BO for deep learning models towards realizing a multi-scale search. Their empirical study revealed that significance of batch BO for hyperparameter tuning. From the literature, it is observed that BO has provision for improving performance of ML and DL models. Based on this proposition, the investigation has revealed that CNN with BO with further improvement has potential to leverage performance of long document classification. Our contributions in this paper are as follows.

1. We proposed an algorithm named Enhanced Bayesian Optimization (EBO) designed to optimize hyperparameter tuning.
2. We also proposed another algorithm known as Hybrid Model with Bayesian Optimization for Long Document Classification (HMBO-LDC). The latter invokes the former appropriately in order to improve parameter optimization of the proposed hybrid model prior to performing long document classification.
3. HMBO-LDC is evaluated and compared against existing models such as CNN feature aggregation method, CNN with LSTM and CNN with recurrent attention model. Experimental results revealed that HMBO-LDC outperforms other methods with highest classification accuracy 98.76%.

The remainder of the paper is structured as follows. Section 2 reviews relevant prior works on long document classification and BO. Section 3 presents the proposed hybrid approach along with enhanced BO. Section 4 presents experimental results while Section 5 concludes out method and results besides giving scope for future work.

II. Related Work

This section reviews literature on existing models that were used for document classification. It also throws light on possible optimizations towards improving parameter selection.

A. *Deep Learning for Document Classification*

Chalkidis and Kampas [1] used deep learning techniques for analysing legal documents. Their methodology has provision for taking legal document corpora as input and perform learning based analysis for document classification and data retrieval. They exploited word embeddings named Law2Vec for improving text processing. In future they intend to combine data coming from multiple sources. Fawaz et al. [2] investigated on different Deep Neural Networks (DNNs) to classify time series data. They considered both open source and other publicly available deep learning models for automatic text classification. Banerjeea et al. [3] focused on radiology reports classification using deep learning models. They compared performance between Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) architectures. They found that both CNN and RNN are feasible deep learning models for document classification. Guo et al. [4] investigated on information retrieval (IR) and the ranking models that are useful in IR research. They considered different applications like retrieval of documents, question answering and automatic conversion. Their empirical study revealed that deep learning models are useful in document classification. They intend to improve it in future with computer vision and NLP enhancements. Gibert et al. [5] studied on the prospect of ML models in processing textual documents. Their research was on malware related documents that are subjected to classification.

Otter et al. [6] investigated on NLP and deep learning models. With their empirical study, they found that deep learning models are feasible for processing textual documents using NLP. They used a transformer model towards processing text documents. Alarsan and Younes [7] explored ML models in order to classify heart disease related documents pertaining to ECG. Their work has found that medical document classification can be done using deep learning models. In future, they intend to work with medical documents pertaining to stress and clinical information. Li et al. [8] proposed a deep learning based framework named HEMOS which automatically detects humour of humans through document media documents. The system is made up of bi-LSTM along with attention mechanism. In future then intend to consider emotion related content for efficient classification of documents. Medvedeva et al. [9] proposed a methodology for analysing European court documents in order to predict court decisions based on the historical data using ML models. It performs classification of legal text documents using SVM. In future they intend to improve their methodology considering more approaches in linguistic and legal analysis. Stein et al. [10] used ML and deep learning models to realize hierarchical text classification. They employed word embeddings in the process of classification. They named their method as FastText which could achieve better accuracy over other models. They intend to improve it with differential function to leverage multi-class classification.

Niu et al. [11] proposed a unified attention model for automatic detection of classes and assign class labels. They found that attention mechanism has its influence on the input representations, output representations, features of input and softness associated with attention technique. It has prediction process towards finding class labels. In future, they intend to improve attention mechanism along with deep learning models. Khanday et al. [12] proposed a methodology for automatic classification of Covid-19 related text documents to classify patients into Covid-19 disease or no disease. Their methodology supports multi-class classification considering Covid-19 and other flu related diseases. They intended to improve their method with better feature engineering approaches. Lavecchia [13] discussed about deep learning models that can be used drug discovery. In the process, they explored challenges, possibilities and future prospects. They intend to combine deep learning models in future for better classification of textual documents. Ngiam and Khor [15] investigated on the healthcare documents in order to discovery different interesting trends or patterns from such data. They intend to explore further on medical documents in future with document classification towards automatic follow up of patients.

Martınez-Arellano et al. [16] proposed a deep learning model for automatic classification of tool wear dynamics in manufacturing industry. It was done as part of monitoring tool conditions automatically. They used time-series imaging data or set of documents that provide rich information about tools are used for classification. In future, they intend to use GPUs for improving its classification performance. Mai et al. [17] focused on bankruptcy prediction from different textual disclosures and text corpora. It was done using deep learning models that are based on CNN. They also used average embedding model for better classification of documents. In future they intend to improve their method with LSTM technique. Flah et al. [18] investigated on different ML models by classification of documents pertaining to structures used in civil engineering. It is designed to have structural health monitoring that helps construction industry to make decisions. Cervantes et al. [19] explored SVM as one of the widely used classification method along with its suitability for document classification. Xu et al. [20] focused on ML methods on safety related applications in order to classify them towards reliability engineering. Different ML and DL models are also fund in the literature for reliability classification of safety applications. In future, they intend to improve it with risk analysis also.

B. Parameter Optimization

In deep learning parameter optimization plays crucial role. Cho et al. [21] discussed about Bayesian Optimization (BO) for automatic improvement of parameters in deep learning models. They used different deep learning models for the application of BO. However, their methodology could provide optimized parameters but they intended to improve it further to determine the range of value for each parameter in future. Victoria and Maragatham [22] explored BO approach for automatic optimization of parameters pertaining to deep learning models. They explored different values for parameters and then optimized to have best parameters. Their research revealed that BO has its impact on accuracy of underlying CNN model. Sameen et al. [23] considered the problem of prediction of landslide susceptibility. In the process they employed BO technique for CNN in order to improve its performance. Their method showed the importance of BO in presence of computer vision applications using CNN models. Ranjit et al. [24] considered cloud infrastructure and deep learning models towards their optimization. For parameter tuning, they explored BO as a suitable candidate for it has provision to know best parameters for given problem. In future then intend to improve it further towards reducing computational complexity. Wang et al. [25] employed BO in deep learning model used for automatic crop disease prediction. Particularly, they explored rice diseases using BO along with deep learning. In future then intend to integrate soil, weather and location to improving smart farming possibilities.

Cui and Bai [26] proposed a new optimization method for CNN. It combines optimization method and genetic algorithm towards realizing parameter optimization in CNN. The genetic algorithm involves multi-level and multi-scale approach. Ragab et al. [27] proposed multiple CNN variants and make them into ensemble for automatic classification of environmental sounds. Their work incudes feature selection and BO for multi-class classification. In the process of searching, they intend to incorporate adaptive approach for improving search operation in future. Borgli et al. [28] proposed an algorithm for BO based optimization that cloud improve medical image analysis using transfer learning along with CNN. They wanted to improve it further towards avoiding overfitting problem with model configurations and BO. Joy et al. [29] introduced a concept known as batch BO for deep learning models towards realizing a multi-scale search. Their empirical study revealed that significance of batch BO for hyperparameter tuning. Joy et al. [30] explored directional derivatives along with BO for parameter tuning of deep models. With their approach search space is improved in order to have better approach in BO for parameter tuning. In future, they want to improve their method to have fine grained control over it. From the literature, it is observed that BO has provision for improving performance of ML and DL models. Based on this proposition, the investigation has revealed that CNN with BO with further improvement has potential to leverage performance of long document classification.

III. PROPOSED FRAMEWORK

A long document classification framework is proposed, as presented in Figure 1, based on a hybrid deep learning approach and proposed enhanced Bayesian optimization method for hyperparameter optimization of the hybrid deep learning model.

A. Framework

This framework has provision for both hybrid deep learning based models that work together for long document classification and also enhanced BO. The proposed deep learning model with hybrid approach is presented in Figure 2. This model is part of the proposed long document classification framework. As presented in Figure 1, the hybrid model includes recurrent network, attention network and glimpse network. More information about the hybrid deep learning model is provided in section B.

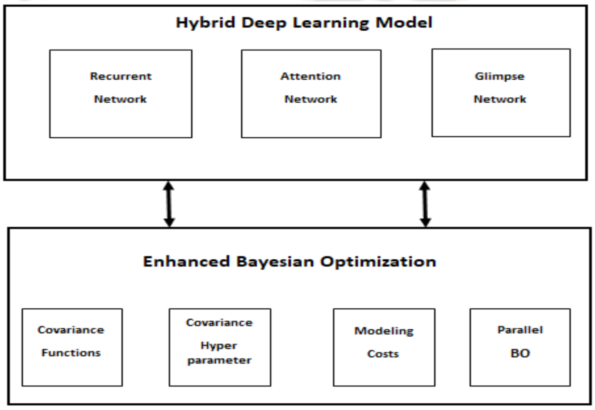

Figure 1: Proposed framework with enhanced BO for long document classification

The proposed enhanced BO is also part of the framework. Enhanced BO is based on consideration of covariance functions, covariance hyperparameters, modelling costs and parallel BO for optimizing the proposed hybrid model. More details on the enhanced BO are provided in Section 3.3.

B. Hybrid Deep Learning Model

A hybrid deep learning model is realized by combining deep learning models such as CNN, RNN and LSTM besides attention mechanism and glimpse network. This hybrid model is influenced by the ideas presented in [32]. Our model 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 model 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 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 2, different kinds of networks such as CNN, RNN with attention mechanism and glimpse network are used along 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.

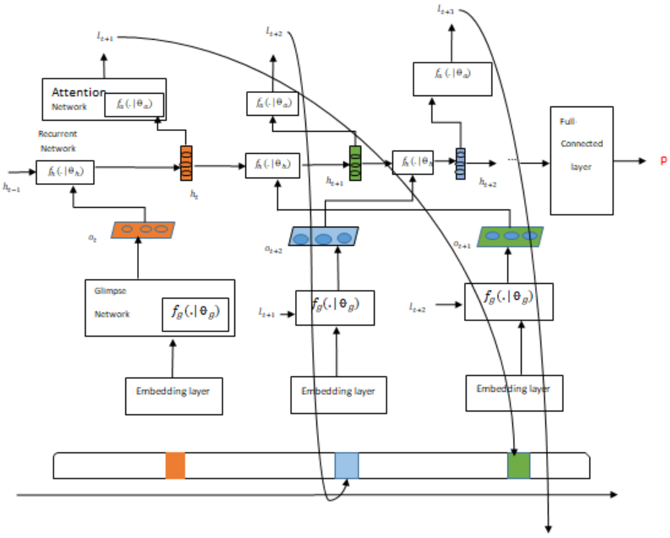

Figure 2: Proposed hybrid model for long document classification.

C. 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.

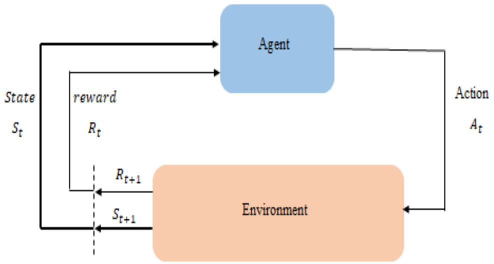

Figure 3: Illustrates reinforcement learning process

Environment in the proposed system, as presented in Figure 3, 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.

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

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

With the proposed framework, the given set of documents are subjected to efficient classification. The dataset obtained from [31] 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 Hybrid Model with Bayesian Optimization for Long Document Classification (HMBO-LDC).

D. Algorithm

An algorithm known as Hybrid Model with Bayesian Optimization for Long Document Classification (HMBO-LDC) is proposed. It is designed to exploit the proposed framework and the underlying mechanisms such as RL and enhanced BO based hyper-parameter optimization.

Algorithm: Hybrid Model with Bayesian Optimization for Long Document Classification (HMBO-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 ← EnhancedBO(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 $\theta_a$ using RL
  Update $\theta_h$ and $\theta_g$ 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.

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

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.

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

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.

E. Hyper-Parameter Optimization using Enhanced BO

In BO, it is important to find function $f(x)$ pertaining to a bounded set χ that we take on a subset of R^D. BO is different from other optimizations as it builds a probabilistic model for $f(x)$ besides exploiting the model for decision making. It does not simply relay on approximations of Hessian and local gradient but considers all prior evaluations of $f(x)$. Thus it can find non-convex functions minimally to avoid computational complexity. With $f(x)$ evaluations becoming expensive as it needs training using ML, this overhead can be justified as the model results in good decisions. Our work in this regard is motivated by the work of Brochu *et al.* [36]. With BO two important choices are made pertaining to choosing a prior over functions reflecting any assumptions and construction of utility function that determines next point for evaluation. We considered the Gaussian process (GP) as prior distribution on functions as it powerful. It takes the form such as $f: \chi \longrightarrow R$. GP is characterized by $\{X_n \in \chi\}_{n=1}^{N}$ inducing multivariate Gaussian distribution on R^N. From finite set of points (N), the nth point is taken as function value $f(X_n)$. Such distribution has elegant marginalization properties helping computation of conditions and marginal. On functions, the resulting distribution is determined by $m: \chi \rightarrow R$ which is a mean function and a covariance function $K: \chi \times \chi \rightarrow R$. More on Gaussian processes can be found in [35].

i. BO and its Acquisition Functions

As we assumed, $f(x)$ is from Gaussian process prior and our observations form $\{X_n, y_n\}_{n=1}^{N}$ where noise introduction is in the form $y_n \sim N(f(X_n), v)$. Then an acquisition function $a: \chi \to R^+$ determines the next point to be evaluation through proxy optimization. It is denoted as $X_{next} = argmax_X\, a(X)$ based on prior observations and hyper parameters of GP. This dependency is expressed as $a(X; \{X_n, y_n\}, \theta)$. Acquisition function has many choices. However, with respect to Gaussian process prior, those functions relay on the model via predictive mean function such as $\mu(X; \{X_n, y_n\}, \theta)$ and variance function denoted as $\sigma^2(X; \{X_n, y_n\}, \theta)$. In this regard, $X_{best} = argmin_{X_n} f(X_n)$ is used to compute current best value while $\Phi(\cdot)$ denotes cumulative distribution function. To maximize improvement probability analytical computation is carried out as expressed in Eq. 1.

$$a_{P1}(X; \{X_n, y_n\}, \theta) = \emptyset(\gamma(X)), \quad \gamma(X) = \frac{f(X_{best}) - \mu(X; \{X_n, y_n\}, \theta)}{\sigma(X; \{X_n, y_n\}, \theta)}. \qquad (1)$$

Alternatively, it is possible to compute expected improvement (EI) as in Eq. 2.

$$\alpha_{EI}(X; \{X_n, y_n\}, \theta) = \sigma(X, \{X_n, y_n\}, \theta)(\gamma(X)\emptyset(\gamma(X)) + N(\gamma(X); 0,1)) \qquad (2)$$

Upper confidence bound of GP is of late exploited to realize acquisition functions as expressed in Eq. 3.

$$\alpha_{LCB}(X; \{X_n, y_n\}, \theta) = \mu(X; \{X_n, y_n\}, \theta) \text{-} k\sigma(X; \{X_n, y_n\}, \theta) \qquad (3)$$

The balance between exploration and exploitation is achieved using a tunable parameter k. EI is considered in this work.

ii. Covariance Hyper parameters and Covariance Functions

GP supports rich distribution on functions and it is based on covariance function. Relevance determination is important in this context. This is computed as in Eq. 4.

$$K_{SE}(X, X^{`}) = \theta_0 exp\,\{-\tfrac{1}{2} r^2(X, X^{`})\}$$
$$r^2(X, X^{`}) = \sum_{d=1}^{D} \frac{(x_d - x_d^{`})^2}{\theta_d^2} \qquad (4)$$

To realize further optimizations 5/2 kernel is employed and this is as expressed in Eq. 5.

$$K_{M52}(X, X^{`}) = \theta_0 \left(1 + \sqrt{5r^2(X, X^{`})} + \tfrac{5}{3} r^2(X, X^{`})\right) exp\,\{-\sqrt{5}r^2(X, X^{`})\} \qquad (5)$$

It results in sample functions that are increasingly differentiable sans smoothness of the squared exponential. Once form of covariance is determined, it is important to manage hyper parameters. As we are using D + 3 Gaussian process, hyper parameters include constant mean m, observation noise v and covariance amplitude θ_0. Most common approach to optimize parameters linked to Gaussian process is expressed as $p(y|\{X_n\}_{n=1}^{N}, \theta, v, m) = N(y|m1, \sum_\theta \; + v1)$. Thus an integrated acquisition function is as expressed in Eq. 6.

$$\hat{\alpha}(X; \{X_n, y_n\}) = \int \; a(X; \{X_n, y_n\}, \theta) p(\{X_n, y_n\}_{n=1}^{N}) d\theta \qquad (6)$$

Both θ and $a(X)$ depend on all observations. For EI it is important to achieve generalization in hyper parameters. To achieve this Monte Carlo estimate is employed. Slice mapping discussed in [34] is used to acquire samples efficiently.

Table 2: Notations used in hyper-parameter optimization using Enhanced BO

Notation	**Meaning**
χ	Denotes bounded set
$y = [y_1, y_2, \dots, y_N]^T$	Covariance matrix
v	Denotes observation noise
m	Indicates a constant mean
k	Value used to balance exploitation and exploration
$f(x)$	Function
$a(X)$	Value acquired from different observations
$\Phi(\cdot)$	Indicates cumulative distribution function
GP	Prior distribution
$\theta_{1:D}$	Scales of D length
θ_0	Indicates covariance amplitude
$y_n \sim N(f(X_n), v)$ and v	Function observation with variance of noise
$(X): \chi \to R^+$	Indicates a duration function
R^D	Indicates a subset

iii. Considering Modelling Costs

BO needs to consider the cost of modeling with quick optimization. We considered EI per second for optimization which results in acquiring points without causing much overhead. As the true objective function *f*(X) and duration function $(X): \chi \to R^+$ are not known, we assume that they are independent but useful for capturing through GP variants for multi-task learning. With the independence assumption in place, it becomes easier to compute the inverse duration expected in order to compute EI per second.

iv. Parallelizing Bayesian Optimization Considering Modelling Costs

Due to the emergence of multi-core computing, it is possible to parallelize BO procedures. With batch parallelism, we determine the point to be evaluated next. As we cannot employ

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

the same function repeatedly, experiments are repeated. We also proposed a sequential strategy that benefits from tractable inference properties of GP to calculate Monte Carlo estimations associated with acquisition function. With N evaluations and $\{X_n, y_n\}_{n=1}^{N}$ as yielding data, it is possible that J evaluation are pending at different locations expressed as $\{X_j\}_{j=1}^{J}$. A new point is chosen depending on expected acquisition function considering all outcomes associated with pending evaluations.

$$\hat{a}(X;\{X_n, y_n\}, \theta, \{X_j\}) = \int_{R^J} \alpha(\mathrm{X};\{X_n, y_n\}, \theta, \{X_j, y_j\})\, p\left(\{X_j\}_{j=1}^{J}, \{X_n, y_n\}_{n=1}^{N}\right) \mathrm{dy1} \ldots \mathrm{dyj} \qquad (7)$$

This it is nothing but expectation of $a(x)$ in presence of a J-dimensional Gaussian distribution whose covariance and mean can be computed with ease. With regard to covariance hyper parameter it is easier to computed expected acquisition using samples. This kind of work is also found in [33] but we saw that Monte Carlo procedure for estimation is highly effective.

```
Algorithm:  Enhanced Bayesian Optimization (EBO)
Inputs:  Document D, initial hyper-parameters λ
Output: Updated hyper-parameters λ
1.  Begin
2.  Compute covariance function $K: \chi \times \chi \to R$
3.  Compute mean function $m: \chi \to R$
4.  Find probability of improvement
5.  Compute expected improvement
6.  Optimize marginal likelihood of parameters
7.  Update λ
8.  Return λ
9.  End
```

Algorithm 2: Enhanced Bayesian Optimization (EBO)

As presented in Algorithm 2, EBO takes the document D as input as the ML model needs to be subjected to hyperprameter optimization based on the data. Since data to be processed can help in setting appropriate hyperparameters, the algorithm strives to come up with tuned hyperparameters. The algorithm is generic in nature and works for different ML models. The process is based on Gaussian process priors. The algorithm computes covariance function followed by mean function. It gives clue pertaining to probability of improvement. Then it computes expected improvement. Afterwards the algorithm optimizes marginal likelihood of parameters prior to updating and returning them.

IV. Results and Discussion

This section presents result of the empirical study. Observations are made in terms of accuracy in long document classification. Since accuracy is the metric based on confusion matrix and dynamics of positive and negative predictions, this measure is widely used for performance comparison. The proposed method HMBO-LDC is compared against existing methods such as CNN with feature aggregation, CNN with LSTM and CNN with recurrent attention model. Dataset used for our empirical study is collected from [31].

Window Size	**Accuracy (%)**			
	CNN Feature Aggregation	**CNN with LSTM**	**CNN with Recurrent Attention Model**	HMBO-LDC **(Proposed)**
10	86.37	83.42	89.84	93.15
20	86.53	86.05	90.15	94.56
50	86.98	86.71	91.67	95.45
100	87.03	87.23	91.89	95.84
200	87.21	88.06	92.23	96.25
400	87.56	88.86	93.45	97.81
500	87.95	89.11	93.87	98.25

Table 3: Performance comparison with 200 words

Table 3 presents accuracy (%) of long document classification exhibited by different existing methods and proposed method against varied window size. These observations are when 200 words are used in experiments.

Window Size	**Accuracy (%)**			
	CNN Feature Aggregation	**CNN with LSTM**	**CNN with Recurrent Attention Model**	HMBO-LDC **(Proposed)**
10	87.48	86.28	90.05	94.58
20	90.11	88.41	92.22	95.65
50	90.33	89.08	93.08	96.24
100	90.60	89.08	93.56	96.69
200	92.08	90.08	93.67	96.94
400	92.12	90.12	93.53	97.25
500	92.78	91.35	93.95	97.32

Table 4: Performance comparison with 400 words

Table 4 presents accuracy (%) of long document classification exhibited by different existing methods and proposed method against varied window size. These observations are when 400 words are used in experiments.

Window Size	**Accuracy (%)**			
	CNN Feature Aggregation	**CNN with LSTM**	**CNN with Recurrent Attention Model**	HMBO-LDC **(Proposed)**
10	88.36	86.66	90.11	93.62
20	91.21	89.59	92.38	95.52
50	91.88	89.92	93.68	96.58
100	92.16	90.23	94.24	97.15
200	93.35	90.51	94.11	97.56
400	93.68	90.78	94.54	97.94
500	93.89	91.05	94.89	98.82

Table 5: Performance comparison with 600 words

Table 5 presents accuracy (%) of long document classification exhibited by different existing methods and proposed method against varied window size. These observations are when 600 words are used in experiments.

Window Size	Accuracy (%)			
	CNN Feature Aggregation	CNN with LSTM	CNN with Recurrent Attention Model	HMBO-LDC (Proposed)
10	89.14	89.44	92.36	96.10
20	92.13	90.28	92.53	96.56
50	92.32	90.37	93.86	97.45
100	92.63	90.56	94.36	97.89
200	93.38	90.82	94.41	98.15
400	93.41	90.94	93.87	98.45
500	93.56	91.21	94.65	98.68

Table 6: Performance comparison with 800 words

Table 6 presents accuracy (%) of long document classification exhibited by different existing methods and proposed method against varied window size. These observations are when 800 words are used in experiments.

Window Size	Accuracy (%)			
	CNN Feature Aggregation	CNN with LSTM	CNN with Recurrent Attention Model	HMBO-LDC (Proposed)
10	90.26	89.48	93.17	96.24
20	92.21	90.36	93.21	96.68
50	92.59	90.61	94.56	97.43
100	93.78	91.77	94.68	97.62
200	93.89	91.96	95.48	98.21
400	93.97	92.15	95.60	98.52
500	94.02	92.34	94.73	98.76

Table 7: Performance comparison with 1000 words

Table 7 presents accuracy (%) of long document classification exhibited by different existing methods and proposed method against varied window size. These observations are when 1000 words are used in experiments.

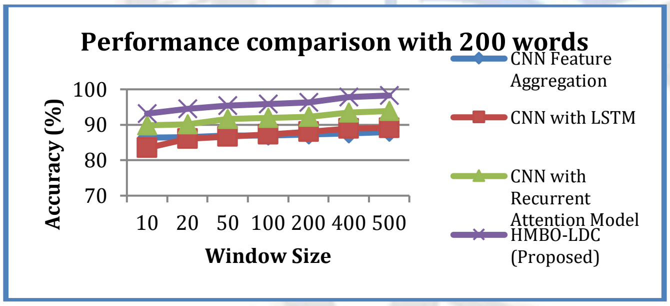

Figure 4: Performance comparison of against window size when number of words is 200

As presented in Figure 4, different deep learning models including the proposed model HMBO-LDC are evaluated for their performance in long document classification. Empirical study is made with different window size and number of words 200. Observations are made in terms of accuracy in classification. The predictions made by the models and ground truth values are used as per confusion matrix to arrive at accuracy details. Each model has shown different level of accuracy. When window size is 500, CNN feature aggregation showed 87.95% accuracy, CNN with LSTM model exhibited 89.11% accuracy, CNN with recurrent attention model showed 93.87% while the proposed model achieved 98.25% accuracy. From the results it is observed that the proposed model outperformed existing ones with highest accuracy.

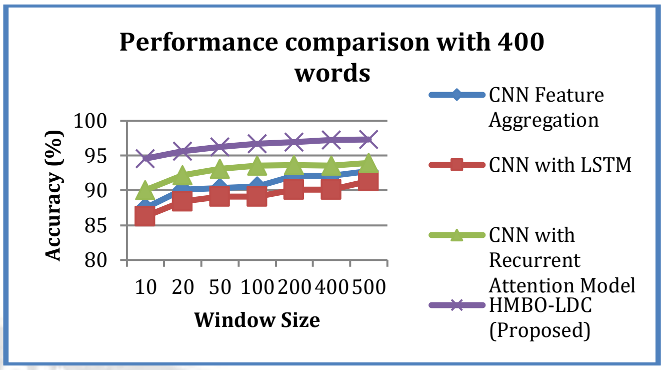

Figure 5: Performance comparison of against window size when number of words is 400

As presented in Figure 4, different deep learning models including the proposed model HMBO-LDC are evaluated for their performance in long document classification. Empirical study is made with different window size and number of words 400. Observations are made in terms of accuracy in classification. The predictions made by the models and ground truth values are used as per confusion matrix to arrive at accuracy details. Each model has shown different level of accuracy. When window size is 500, CNN feature aggregation showed 87.95% accuracy, CNN with LSTM model exhibited 89.11% accuracy, CNN with recurrent attention model showed 93.87% while the proposed model achieved 98.25% accuracy. From the results it is observed that the proposed model outperformed existing ones with highest accuracy.

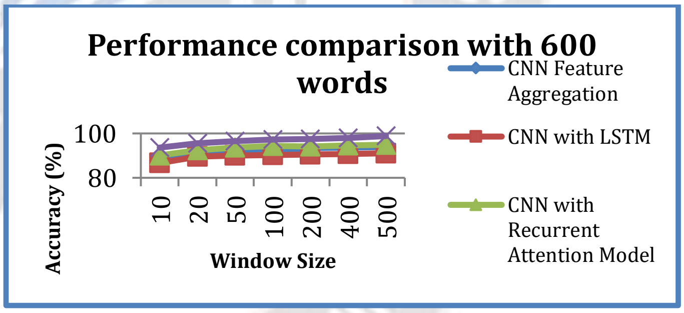

Figure 6: Performance comparison of against window size when number of words is 600

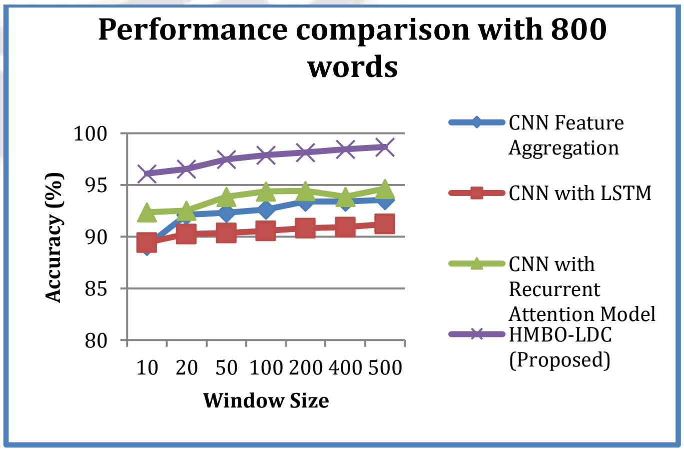

Figure 7: Performance comparison of against window size when number of words is 800

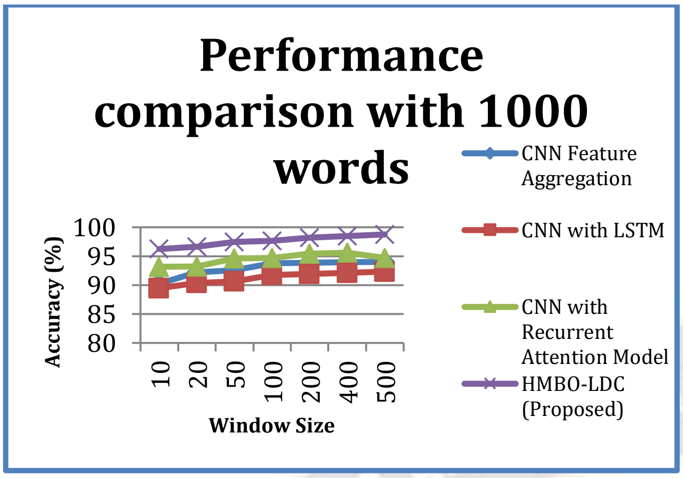

Figure 8: Performance comparison of against window size when number of words is 1000.

V. Conclusion And Future Work

In this paper, we proposed an algorithm named Enhanced Bayesian Optimization (EBO) designed to optimize hyperparameter tuning. This algorithm is designed and implemented to have better approach in BO overcoming its inherent limitations. Such algorithm is crucial for hyperparameter tuning of deep learning models. We also proposed another algorithm known as Hybrid Model with Bayesian Optimization for Long Document Classification (HMBO-LDC). The latter invokes the former appropriately in order to improve parameter optimization of the proposed hybrid model prior to performing long document classification. HMBO-LDC is evaluated and compared against existing models such as CNN feature aggregation method, CNN with LSTM and CNN with recurrent attention model. Experimental results revealed that HMBO-LDC outperforms other methods with highest classification accuracy 98.76%. Our work has revealed significance of EBO. However, we intend to consider an ensemble deep learning model along with EBO for leveraging classification performance further in future.

References

[1] Chalkidis, Ilias and Kampas, Dimitrios (2018). Deep learning in law: early adaptation and legal word embeddings trained on large corpora. Artificial Intelligence and Law. http://doi:10.1007/s10506-018-9238-9

[2] Ismail Fawaz, Hassan; Forestier, Germain; Weber, Jonathan; Idoumghar, Lhassaneand Muller, Pierre-Alain (2019). Deep learning for time series classification: a review. Data Mining and Knowledge Discovery. http://doi:10.1007/s10618-019-00619-1

[3] 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 andLungren, 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–. http://doi:10.1016/j.artmed.2018.11.004

[4] Guo, Jiafeng; Fan, Yixing; Pang, Liang; Yang, Liu; Ai, Qingyao; Zamani, Hamed; Wu, Chen; Croft, W. Bruce and Cheng, Xueqi (2019). A Deep Look into neural ranking models for information retrieval. Information Processing & Management, 102067–. http://doi:10.1016/j.ipm.2019.102067

[5] Gibert, Daniel; Mateu, Carles and Planes, Jordi (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. Journal of Network and Computer Applications, 102526. http://doi:10.1016/j.jnca.2019.102526

[6] Otter, Daniel W.; Medina, Julian R. and Kalita, Jugal K. (2020). A Survey of the Usages of Deep Learning for Natural Language Processing. IEEE Transactions on Neural Networks and Learning Systems, 1–21. http://doi:10.1109/TNNLS.2020.2979670

[7] Alarsan, FajrIbrahem and Younes, Mamoon (2019). Analysis and classification of heart diseases using heartbeat features and machine learning algorithms. Journal of Big Data, 6(1), 81–. http://doi:10.1186/s40537-019-0244-x

[8] Li, Da; Rzepka, Rafal; Ptaszynski, Michal and Araki, Kenji (2020). HEMOS: A novel deep learning-based fine-grained humor detecting method for sentiment analysis of social media. Information Processing & Management, 57(6), 102290. http://doi:10.1016/j.ipm.2020.102290

[9] Medvedeva, Masha; Vols, Michel and Wieling, Martijn (2019). Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law. http://doi:10.1007/s10506-019-09255-y

[10] Stein, Roger Alan; Jaques, Patrícia A. and Francisco Valiati, João (2018). An Analysis of Hierarchical Text Classification Using Word Embeddings. Information Sciences, S0020025518306935. http://doi:10.1016/j.ins.2018.09.001

[11] ZhaoyangNiu; GuoqiangZhong and Hui Yu; (2021). A review on the attention mechanism of deep learning .Neurocomputing. http://doi:10.1016/j.neucom.2021.03.091

[12] Khanday, AkibMohiUd Din; Rabani, Syed Tanzeel; Khan, QamarRayees; Rouf, Nusrat and MohiUd Din, Masarat (2020). Machine learning based approaches for detecting COVID-19 using clinical text data. International Journal of Information Technology. http://doi:10.1007/s41870-020-00495-9

[13] Lavecchia, Antonio (2019). Deep learning in drug discovery: opportunities, challenges and future prospects. Drug Discovery Today, S135964461930282X–. http://doi:10.1016/j.drudis.2019.07.006

[14] Zhang, Shen; Zhang, Shibo; Wang, Bingnan and Habetler, Thomas G. (2020). Deep Learning Algorithms for Bearing Fault Diagnostics â“ A Comprehensive Review. IEEE

Access, 1–1. http://doi:10.1109/ACCESS.2020.2972859

[15] Ngiam, Kee Yuan and Khor, IngWei (2019). Big data and machine learning algorithms for health-care delivery. The Lancet Oncology, 20(5), e262–e273. http://doi:10.1016/S1470-2045(19)30149-4

[16] Martínez-Arellano, Giovanna; Terrazas, German and Ratchev, Svetan (2019). Tool wear classification using time series imaging and deep learning. The International Journal of Advanced Manufacturing Technology. http://doi:10.1007/s00170-019-04090-6

[17] Mai, Feng; Tian, Shaonan; Lee,Chihoon and Ma, Ling (2018). Deep Learning Models for Bankruptcy Prediction using Textual Disclosures. European Journal of Operational Research, S0377221718308774–. http://doi:10.1016/j.ejor.2018.10.024

[18] Flah, Majdi; Nunez, Itzel; Ben Chaabene, Wassim and Nehdi, Moncef L. (2020). Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review. Archives of Computational Methods in Engineering. http://doi:10.1007/s11831-020-09471-9

[19] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). *A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing.* http://doi:10.1016/j.neucom.2019.10.118

[20] Zhaoyi Xu and Joseph Homer Saleh; (2021). Machine learning for reliability engineering and safety applications: Review of current status and future opportunities . Reliability Engineering &amp; System Safety. http://doi:10.1016/j.ress.2021.107530

[21] Cho, Hyunghun; Kim, Yongjin; Lee, Eunjung; Choi, Daeyoung; Lee, Yongjae and Rhee, Wonjong (2020). Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks. IEEE Access, 8, 52588–52608.

[22] Victoria, A. Helen and Maragatham, G. (2020). Automatic tuning of hyperparameters using Bayesian optimization. Evolving Systems.

[23] Sameen, Maher Ibrahim; Pradhan, Biswajeet and Lee, Saro (2019). Application of convolutional neural networks featuring Bayesian optimization for landslide susceptibility assessment. CATENA, 104249.

[24] Ranjit, Mercy Prasanna; Ganapathy, Gopinath; Sridhar, Kalaivani and Arumugham, Vikram (2019). IEEE 12th International Conference on Cloud Computing (CLOUD) - Efficient Deep Learning Hyperparameter Tuning Using Cloud Infrastructure: Intelligent Distributed Hyperparameter Tuning with Bayesian Optimization in the Cloud. , 520–522.

[25] YibinWang;Haifeng Wang and Zhaohua Peng; (2021). Rice diseases detection and classification using attention based neural network and bayesianoptimization . Expert Systems with Applications,.

[26] Cui, Hua and Bai, Jie (2019). A new hyperparameters optimization method for convolutional neural networks. Pattern Recognition Letters, S0167865519300327.

[27] Mohammed Gamal Ragab; SaidJadidAbdulkadir; NorshakirahAziz; HithamAlhussian; AbubakarBala and Alawi Alqushaibi; (2021). An Ensemble One Dimensional Convolutional Neural Network with Bayesian Optimization for Environmental Sound Classification . Applied Science,.

[28] Borgli, Rune Johan; KvaleStensland, Hakon; Riegler, Michael Alexander and Halvorsen, Pal (2019). 13th International Symposium on Medical Information and Communication Technology (ISMICT) - Automatic Hyperparameter Optimization for Transfer Learning on Medical Image Datasets Using Bayesian Optimization. , 1–6.

[29] Joy, TinuTheckel; Rana, Santu; Gupta, Sunil and Venkatesh, Svetha (2019). Batch Bayesian optimization using multi-scale search. Knowledge-Based Systems, S095070511930293X–.

[30] Joy, TinuTheckel; Rana, Santu; Gupta, Sunil and Venkatesh, Svetha (2020). Fast hyperparameter tuning using Bayesian optimization with directional derivatives. Knowledge-Based Systems, 106247.

[31] ArXiv dataset. Collected from https://www.kaggle.com/datasets/1b6883fb66c5e7f67c697c2547022cc04c9ee98c3742f9a4d6c671b4f4eda591

[32] 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.

[33] David Ginsbourger and Rodolphe Le Riche. Dealing with asynchronicity in parallel Gaussian process based global optimization. http://hal.archives-ouvertes.fr/hal-00507632, 2010.

[34] Iain Murray and Ryan P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In Advances in Neural Information Processing Systems 24, pages 1723–1731. 2010.

[35] Carl E. Rasmussen and Christopher Williams. Gaussian Processes for Machine Learning. MIT Press, 2006.

[36] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. pre-print, 2010. arXiv:1012.2599