

## Article

# SHO-CNN: A Metaheuristic Optimization of a Convolutional Neural Network for Multi-Label News Classification

Muhammad Imran Nadeem <sup>1</sup>, Kanwal Ahmed <sup>1</sup>, Dun Li <sup>1,\*</sup>, Zhiyun Zheng <sup>1</sup>, Hafsa Naheed <sup>2</sup>, Abdullah Y. Muaad <sup>3,4</sup>, Abdulrahman Alqarafi <sup>5,\*</sup> and Hala Abdel Hameed <sup>6,7</sup>

<sup>1</sup> School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China

<sup>2</sup> Department of Computer Science, University of Science and Technology, Bannu 28100, Pakistan

<sup>3</sup> Sana'a Community College, Sana'a 5695, Yemen

<sup>4</sup> Department of Studies in Computer Science, University of Mysore, Mysuru 570005, India

<sup>5</sup> College of Computer Science and Engineering, Taibah University, Medina 42353, Saudi Arabia

<sup>6</sup> Faculty of Computer and Information Systems, Fayoum University, Fayoum 63514, Egypt

<sup>7</sup> Khaybar Applied College, Taibah University, Medina 42353, Saudi Arabia

\* Correspondence: ielidun@zzu.edu.cn (D.L.); asalqarafi@taibahu.edu.sa (A.A.)

**Abstract:** News media always pursue informing the public at large. It is impossible to overestimate the significance of understanding the semantics of news coverage. Traditionally, a news text is assigned to a single category; however, a piece of news may contain information from more than one domain. A multi-label text classification model for news is proposed in this paper. The proposed model is an automated expert system designed to optimize CNN's classification of multi-label news items. The performance of a CNN is highly dependent on its hyperparameters, and manually tweaking their values is a cumbersome and inefficient task. A high-level metaheuristic optimization algorithm, spotted hyena optimizer (SHO), has higher advanced exploration and exploitation capabilities. SHO generates a collection of solutions as a group of hyperparameters to be optimized, and the process is repeated until the desired optimal solution is achieved. SHO is integrated to automate the tuning of the hyperparameters of a CNN, including learning rate, momentum, number of epochs, batch size, dropout, number of nodes, and activation function. Four publicly available news datasets are used to evaluate the proposed model. The tuned hyperparameters and higher convergence rate of the proposed model result in higher performance for multi-label news classification compared to a baseline CNN and other optimizations of CNNs. The resulting accuracies are 93.6%, 90.8%, 68.7%, and 95.4% for RCV1-v2, Reuters-21578, Slashdot, and NELA-GT-2019, respectively.

**Keywords:** data mining; deep learning (DL); multi-label text classification (MLTC); spotted hyena optimizer (SHO); convolutional neural network (CNN); hyperparameter optimization



**Citation:** Nadeem, M.I.; Ahmed, K.; Li, D.; Zheng, Z.; Naheed, H.; Muaad, A.Y.; Alqarafi, A.; Abdel Hameed, H. SHO-CNN: A Metaheuristic Optimization of a Convolutional Neural Network for Multi-Label News Classification. *Electronics* **2023**, *12*, 113. <https://doi.org/10.3390/electronics12010113>

Academic Editor: Yue Wu

Received: 1 December 2022

Revised: 17 December 2022

Accepted: 21 December 2022

Published: 27 December 2022



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

News can be presented in the form of newspapers, periodicals, television, radio, blogs, news articles, etc. News media have been essential, critical, and informative from the beginning of time. News can be plain text and extremely complicated at the same time. This research accentuates news articles and their classification. News classification poses challenges, as news contains multiple semantics [1]. For instance, a single news story may contain information from multiple categories, such as computers, exports, and the economy. Legacy machine learning (ML) techniques classify news as a single-label classification task. Text classification is a primary task in natural language processing (NLP) [2]. It is especially useful for the classification of rich semantic text such as news articles. Multi-label text classification (MLTC) is the process of associating a text with numerous classification labels.

Over the years, researchers utilizing deep learning (DL) techniques have demonstrated its effectiveness in numerous domains [3–18]. Among various DL techniques, convolutional neural networks (CNNs) do not use matrix multiplication, rather they use a special

technique called ‘convolution’. The combination of convolution and pooling layers is used for the classification task [19,20]. Deep learning creates a learning model and forms learning patterns from a learning dataset and then efficiently predicts labels based on that learning [21]. CNNs have attained top performance in text categorization because of their outstanding capability of capturing local relations of temporal or hierarchical structures [22–24].

Kalchbrenner et al. [22] modeled words with stacking of numerous layers of convolution. Collobert et al. [23] and Yu et al. [24] both simulated text using a single convolutional layer. According to Kim et al. [25], the use of many convolutional layers is beneficial in the process of extracting high-level abstract characteristics. According to Yin et al. [26], the network can better handle sentences of varying lengths due to the use of a pooling operation. Furthermore, they investigated multichannel convolution architecture and variably sized feature detectors. Starting out, Conneau et al. [27] used a deep convolutional architecture consisting of 29 convolutional layers. To achieve superior results in detecting correlations among data, they increased the layer count to 49. However, in this context, they were unable to reach state-of-the-art results.

In order to significantly outperform long short-term memory (LSTM)-based approaches, Gu et al. [28] coupled a language CNN with recurrent highway networks. One study [29] employed LSTM to manage the flow of data throughout the network. The input window size in [28,29] was restricted due to the usage of a recurrent framework. There is also difficulty in representing hierarchies and identifying long-term dependencies [30]. Unlike other methods, CNN’s computational time scales linearly with the length of the sequences. CNNs have an advantage over recurrent neural networks since they not only obtain long-range information but also obtain a hierarchical representation of the input words [30].

The effectiveness of CNN is highly dependent on setting the external parameters, which vary based on data type, size, and origin, and therefore must be manually adjusted for each unique data set. These parameters are known as hyperparameters. The primary objective of this study is to obtain the ideal set of hyperparameters in the shortest possible time without any manual intervention, hence producing a classification model with improved performance and better time efficiency.

Recently, various metaheuristic algorithms have been proposed as effective methods for optimization [31–34]. Meta-heuristic algorithms used in such techniques have proven to be effective at reducing the search space needed to find an optimal solution. Unfortunately, once these algorithms are applied to huge search areas, they tend to use up a lot of time as well as computational resources and fall back on local optima. They also suffer from premature convergence and do not provide a middle ground between local and global searches. Thus, it is necessary to successfully fill the void left by metaheuristic optimization techniques when working with a wide search space to implement task-specific computational applications.

In order to get better outcomes, there is need for the development of more-comprehensive multi-text classification methods. In the disciplines of engineering and research, the utilization of the original heuristic algorithm is an extremely common practice [35]. Because of its excellent search performance, the spotted hyena optimizer, also known as the SHO algorithm, is utilized in the fields of engineering and science.

SHO is a metaheuristic optimization technique inspired by the principles of social and hunting behavior of spotted hyenas [36–41]. During the training process of the proposed model, SHO selects the optimal values of hyperparameters to tune the CNN. The best value-selection procedure involves turning the hyperparameters into vectors and then using advanced mathematical modeling to search for and assign values within the defined parameter range [38,39]. As this algorithm is inspired by spotted hyenas, it imitates the hunting behavior of hyenas, i.e., searching, surrounding, and attacking the target. The exploration and exploitation processes of SHO are highly precise and effective. The exploration and exploitation tasks are group-based, so it never descends into local optima. The fast convergence rate along with the robustness make SHO highly adaptable. These

characteristics of SHO make it highly suitable for exploring new domains [40,41]. SHO has already achieved global success in optimization and is regarded as one of the top exploration techniques [42–45]. Therefore, the purpose of this study is to utilize the highest strengths of metaheuristic techniques to overcome CNN's limitations in the field of news MLTC for improved classification accuracy and efficiency.

This research focuses on the use of CNN along with SHO to provide an expert system for automatically optimizing hyperparameters, including learning rate, momentum, number of epochs, batch size, dropout, number of nodes in hidden layers, and activation functions, to ensure the best possible accuracy and efficiency of an MLTC of news. Four publicly accessible news datasets, namely RCV1-v2 [46], Reuters-21578 [47], Slashdot [48], and NELA-GT-2019 [49], are used to evaluate and validate the proposed model. The results are extremely encouraging and reflect the model's high convergence rate, which result in improved CNN performance in terms of accuracy and efficiency. The salient contributions of this research are:

- i. A metaheuristic-optimized SHO-CNN model for multi-label news text classification;
- ii. Automation of hyperparameter optimization of a CNN using SHO;
- iii. Evaluation and comparison of SHO-CNN with other metaheuristic optimization algorithms

The contents of the paper are further arranged as follows: Section 2 represents related multi-label text classification work along with hyperparameter optimization of a CNN and related techniques are also reviewed. Section 3 illustrates the proposed methodology with the help of an algorithm and related steps. Section 4 represents the evaluations, results, and discussion. Finally, Section 5 concludes the research and provides recommendations for related future work.

## 2. Related Work

The literature review includes information along with a background study of the related models and techniques. The summary of the literature review and the limitations in the field of MLTC are also discussed in this section.

### 2.1. Multi-Label Text Classification

MLTC is a technique used to allocate multiple labels to a singular document in the field of NLP [50]. The primary objective of NLP is to utilize mathematical models and algorithms to enable computers to better comprehend and rationalize associations inside and amongst documents [51]. MLTC is primarily used for document classification in several applications [52]. Additionally, MLTC is commonly utilized for the classification of web content [53,54] and recommendation systems to better comprehend context [55,56]. MLTC is able to assign various labels to a single object. For better understanding, assume the list of labels as  $l = \{L_1, L_2, L_3, \dots, L_z\}$ , where  $l$  must assign  $n$  labels for  $L$  subsets to a related instance  $x$  [57]. In contrast to conventional text classification methods, MLTC allows several linked labels to be applied to each instance [58]. The primary goal of MLTC is to determine the maximum number of labels for an instance.

A hierarchical structure of classes has been proposed, where classes are inherited from other classes; if one object is assigned to a class, it may automatically relate to other classes as well. This technique is known as Hierarchical Multi-Label Text Classification (HMLTC) [59]. Another corresponding framework has been proposed, known as the Hierarchical Cognitive Structural Learning Model (HCSM). Furthermore, the study [59] has employed an Attentional Ordered Recurrent Neural Network (AORNN) for generating vector-based calculations to gather knowledge from data, along with a Hierarchical Bidirectional Capsule (HBiCaps) to utilize the said vectors in each iteration. One more related model is the Hierarchical-Attentional Graph Convolutional Recurrent Neural Network (H-AGCRNN), which modifies the vector formation algorithm for semantic extraction. A hierarchy-based update for knowledge extraction has been performed for the Hierarchical Attentional Graph Taxonomy Capsule-Graph Convolutional Network (HAGTP-GCN).

Experiments were performed on four datasets (RCV1, EUR-Lex, WOS-46985, and Patent); the results suggest that HCSM was more useful than H-AGCRNN and HAGTP-GCN.

Ibrahim et al. presented a Generic Hybridized Shallow Neural Network (GHS-NET) to employ CNN along with Bidirectional LSTM (bi-LSTM) for gathering data-related information [60]. The model was evaluated on three datasets related to the medical literature, i.e., Chemical Exposure Analysis, Cancer Hallmarks, and Medical Information Mart for Intensive Care (MIMIC-III). The experimentation process contained a variety of environment settings related to filter sizes, pooling, and activation functions.

Wang et al. proposed a model called Multi-Label Reasoner (ML-Reasoner) [50]; it utilizes a binary classifier to generate classes along with an argumentation process for the classification of data. The process consists of four steps; the first step is parsing all data for text embeddings. Converted vectors are used for the CNN Encoder. The label encoder assigns labels to documents. Later, a sigmoid function is used as an activation function to normalize the output. The said model was evaluated on two datasets (AAPD, RCV1V2). The outcomes demonstrated that the ML-Reasoner model outperformed CNN and LSTM.

Benites and Sapozhnikova modified a fuzzy Adaptive Resonance Associative Map with a hierarchy model known as the Hierarchical ARAM neural network (HARAM) [61]. This approach was designed to enhance the classification effectiveness of large datasets with intricate structures. Chen and Ren proposed Latent Wordwise Label information (MLC-LWL) for multi-label text classification [62]. This model captures information for each word to form a label and then classifies each label based on the matching structure between them. The said model was evaluated and had higher accuracy compared to primary machine learning techniques.

Wang, Tianshi, et al. merged a dynamic semantic representation model and a deep neural network as DSRM-DNN [16]. It made use of a word embedding model along with a clustering technique. DSRMDNN elements are measured and selected based on the word score. Considering backpropagation properties, a deep belief neural network is integrated for text classification. The classification process reconditions newly discovered words along with minimally occurring words throughout the cycle. The model was evaluated using three datasets, i.e., RCV1-v2, Reuters-21578, and EUR-Lex. The results suggest that the performance of the model under consideration is better than that of MultiLabel Decision Tree (ML-DT), Multi-Label k-Nearest Neighbor (ML-KNN), Binary Relevance (BR), Classifier Chains (CC), Multi-Label Neural Networks (MLNN), HARAM, Convolutional and Recurrent Neural Networks (CNNRNN), Hierarchical Label Set Expansion (HLSE), and Supervised Representation Learning (SERL) models.

## 2.2. Hyperparameter Optimization of CNN

Conventional AI-based neural networks use matrix multiplication, but CNN uses the convolution method for mathematical calculations and computing of the classification process [63–65]. There are also variable parameters that affect the performance of CNNs; they are known as hyperparameters. These parameters are external to the CNN and can highly influence the classification output. The determination of highly optimal hyperparameters guarantees a high outcome in terms of accuracy and efficiency, but it is a time-exhausting manual process, even for an expert with plenty of experience in the domain. In a complex CNN configuration, there are numerous hyperparameters along with distinct ranges. It is tedious to manually choose, evaluate, and tune each parameter and their combinations. This necessitates the development of an automated procedure to determine the appropriate hyperparameter values.

In order to tune hyperparameters of a simple CNN, Random Search (RS) and Grid Search (GS) models are widely used [66–68]. These methods are non-adaptive in nature since they rely on readily available results. GS is feasible for a CNN with a small number of hyperparameters, as it evaluates all the combinations of hyperparameters. Therefore, increasing the number of hyperparameters exponentially increases the number of possible combinations, resulting in poor performance and resource utilization [67]. In comparison

to the tuning of hyperparameters by using GS [68], Bergstra and Bengio have used RS for similar experiments [66]. Their results demonstrated that RS is time-efficient and generates more-precise models. The prevailing performance of nonadaptive techniques can be further improved by manually modifying the hyperparameter space.

Adaptive methods such as Bayesian optimization and metaheuristics-based models employ knowledge from data in the early phases [69–71]. Bayesian optimization models are considered to be more straightforward to implement than metaheuristics models. Metaheuristics-based models are best suited for progressive techniques, as they provide efficient handling for a solution space. Neuroevolution is an application of evolutionary computation related to neural networks. The prominent research for this domain includes EPNet [72] and NEAT [73]. Over the years, the evolution of high-speed memory management, especially Graphics Processing Units (GPUs), has enabled highly complex networks to perform efficiently.

In another work, Sun, Xue, Zhang, and Yen proposed an evolutionary CNN known as EvoCNN [74]. It automatically generates a CNN architecture along with initial weights. Ma et al. also proposed models based on evolutionary algorithms [75]. Both proposed models were evaluated on complex networks and resulted in high efficiency. Lecun et al. incorporated Genetic Algorithms (GA) along with grammatical evolution for automatic generation and tuning of hyperparameters for CNNs [64].

The effectiveness of metaheuristics-based algorithms for optimizing CNN hyperparameters has been demonstrated over the years. One such method is known as Particle Swarm Optimization (PSO). PSO-based models have been evaluated using publicly available standard datasets. The results show that PSO-based models have decreased classification error [76,77].

Due to the success of metaheuristic algorithms, several other metaheuristics-based approaches have been proposed, including Differential Evolution (DE) [78], Harmonic Search (HS) [79], and Reinforcement Learning (RL) [80,81]. The proposed models aim to tune hyperparameters and search for the optimal CNN architecture [82,83]. Hakim et al. [36] used the imperialist competitive algorithm (ICA) and Greywolf optimizer (GWO) to enhance CNN performance. The results demonstrated that the CNN optimized with metaheuristic algorithms outperformed the original CNN.

Dhiman and Kumar proposed SHO [38], which is an adaptation of metaheuristic algorithms. The authors evaluated the model on complex networks and standard datasets. Due to its advanced search ability, the results indicated a rapid convergence rate. The SHO algorithm was applied to image matching in the study [42], which resulted in improved accuracy. The modified SHO was utilized for PID parameter optimization in VAR in [44] and contributed to an increase in the diversity of search options.

Kadry et al. [37] have used SHO for CNN-based model optimization. The authors optimized the deep and handcrafted features using SHO and later concatenated them to create a feature vector with rich representation. In other research, Luo, Li, and Zhou employed SHO in the field of image processing [42]. Panda and Majhi [43] used SHO for the prediction of gene selection features using improved multi-objective SHO. Table 1 presents a summary of metaheuristic techniques and their limitations.

**Table 1.** Summary of metaheuristic optimization techniques in the literature.

Name	Contributions	Limitations
Random Search (RS) [66]	It is a direct search technique that does not maximize the problem's gradient. The program uses a random number generator to determine the values of the hyperparameters.	Methods such as this are not adaptive since they use predetermined outcomes. They are an inefficient means of going down, which only lowers the overall performance.
Grid Search (GS) [67,68]	A grid of values is formed by picking two or more parameter values and evaluating all possible permutations of those values.	Larger combinations tend to have a negative impact on efficiency and productivity.
Harmonic Search (HS) [79]	This method was developed specifically for the purpose of resolving difficulties involving optimization with a single target.	After increasing the parameters, the performance starts to suffer, which leads to less accurate results.
Particle Swarm Optimization (PSO) [76,77].	It optimizes a problem by repeatedly searching for better solutions until one is found that meets certain specified criteria.	Given the fixed length of the encoding string, it is required to manually adjust the dense layer parameters and their positions within the encoded text.
Imperialist Competitive Algorithm (ICA) [36]	It can distinguish between distinct sources from a mixed signal.	It does not adjust itself to the changing characteristics of a dynamic system.
Grey Wolf Optimizer (GWO) [36]	The algorithm models the social structure and hunting strategy of grey wolves in the wild. Since there are fewer control settings, it is simple to implement.	Its low accuracy is a result of its slow convergence rate.
Genetic Algorithm (GA) [64]	This technique, which is based on natural selection, is capable of solving both limited and unconstrained optimization problems.	Over time, it may settle on a solution that is less than ideal.
Differential Evolution (DE) [78]	In order to generate a new vector, this technique uses the dissimilarity between two or more existing vectors.	Finding the best way to start a population and set control settings can be difficult. Although the population may not have attained a local optimum, convergence can still take place.

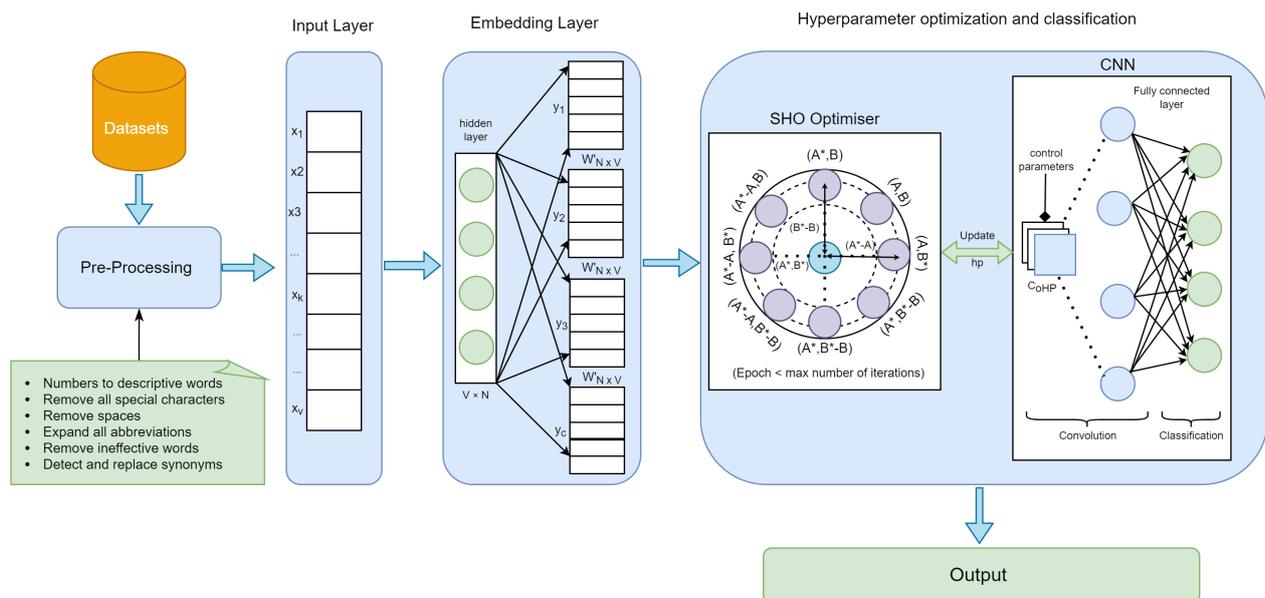
The current literature review confirms the necessity for MLTC of news. A primary CNN is inefficient for MLTC tasks [52]. The performance of a CNN is extremely dependent on the tuning of its hyperparameters, and automation of hyperparameters is crucial work in the field of CNN optimization. The performance of the model can suffer from an increase in complexity at times [67]. Metaheuristic-based optimizations are effective for automating CNN hyperparameter tuning [69,70,75]. Metaheuristic-based optimizations are effective for automating CNN hyperparameter tuning [36,64,76,78,79], but there is no single best-performing solution for all optimization problems [70]. Furthermore, increasing the dimensions greatly diminishes the effectiveness of approaches such as PSO, DE, and GA [84].

The proposed model employs SHO [38] because it holds superior exploration capabilities, rapid convergence rate, and extremely valuable qualities to prevent falling for local optima. In addition, it has been evaluated on complicated networks [42,44]. The optimization of CNN hyperparameters, particularly for an MLTC model, reflects an ongoing challenge. Considering the criticality, importance, and effectiveness of news media, this challenge poses a superior, purposeful, and significant research problem.

### 3. Methodology

The proposed model employs a CNN for the task of MLTC. The characteristics of CNN for the task of text classification are exceptional. The performance of CNN depends heavily on its hyperparameters. These hyperparameters are responsible for the learning

process and output of a model. The hyperparameters are always specified and updated manually to tune the model. To overcome this time-consuming and manual task of tuning a CNN, the proposed model employs SHO for the automated tuning of the CNN. SHO is based on a metaheuristic search algorithm; it fetches the set of hyperparameters and their values to establish the optimal values for each hyperparameter. In this research, the CNN is highly analyzed to optimize related hyperparameters, including batch size, learning rate, momentum, dropout, the number of epochs, the number of nodes, and the activation function. The proposed model for MLTC of news is shown in Figure 1.



**Figure 1.** Architecture of the proposed SHO-CNN for MLTC of news.

The operations of the proposed model can be divided into phases, beginning with preprocessing for cleaning datasets, word embeddings for vector representations, hyperparameter tuning by SHO, and classification output by CNN.

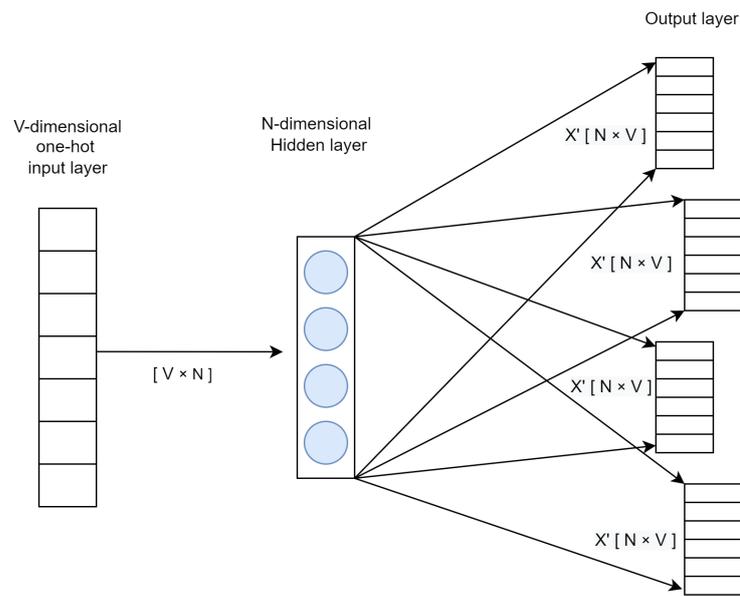
### 3.1. Preprocessing

Preprocessing is generally an initial step for every machine learning model. It involves cleaning and organizing raw data to make it suitable to develop and train a machine learning model. The data cleaning standard may vary as required by the model. Preprocessing text for the proposed model includes the following essential steps:

- Convert numbers to descriptive words;
- Remove all special characters;
- Remove spaces;
- Expand all abbreviations;
- Remove ineffective words;
- Detect and replace synonyms for words.

### 3.2. Word Embeddings

The function of word embedding is to convert textual words into their corresponding numerical vectors; this allows machine learning models to identify and learn hidden patterns among textual data for NLP. In the proposed model, the skip-gram model is used to generate vectors [85]. It involves semantics to create a vector for every vocabulary word, as represented in Figure 2. The related context properties are retrieved from vector documentation. Context properties are compared on the basis of similarity to the input. Input words are referred to as target words, and the output with expected context attributes is known as labels.



**Figure 2.** Skip-gram model.

The target vector and context labels are encoded using one of the primary methods, called ‘one-hot encoding’, which is also known as “1’-of-N” encoding. One-hot encoding stores a vector in the form of 0s and 1s. It is most appropriate for determining the size and orientation of words [86].

### 3.3. SHO-CNN

Considering the social and hunting behavior of a metaheuristic SHO algorithm, SHO is integrated with CNN to optimize the hyperparameters. The course of the current study includes implementation of domain standards and publicly available datasets. Secondly, we build and train the proposed model. Afterward, we evaluate the performance of the proposed model. Finally, we compare the performance of the proposed model with other baseline models using advanced performance measures, including accuracy, precision, recall, and F-measure.

Input and output parameters can be set during the development of machine learning models; however, the ideal values for hyperparameters cannot be determined. No expert is definite regarding the nature and relationship between training datasets and data processing. To circumvent this difficulty, specialists manually supply an extensive range of values for each hyperparameter, which is a time- and resource-intensive process. Ideally, a model ought to be capable of automatically searching for and obtaining the optimal values for each hyperparameter. In general, these hyperparameters differ between models and datasets and cannot be learned directly from data. For the tuning and optimization of hyperparameter values, external optimization techniques are applied.

Dataset  $DS$  used in this research is divided into training  $DS_{Tr}$  and testing  $DS_{Te}$  subsets ( $DS_{Te}, DS_{Tr} \in DS$ ). For every  $DS$ , there are various classification models  $C$  for a CNN. These models take hyperparameters  $HP$  for the CNN into consideration, including the learning rate, momentum, number of epochs, batch size, number of nodes, activation function, and dropout. Let us consider a classification model with optimized hyperparameters as  $C_{oHP}$ , where  $oHP$  is the desired optimized configuration of hyperparameters ( $oHP \in HP$ ). The optimized solution is calculated using SHO. The optimized solution minimizes the loss function at a rate of  $L$ . The hyperparameters of  $C_{oHP}$  can be validated via Equations (1) and (2).

$$oHP(DS_{Tr}) = \underset{C_{oHP} \in C}{\text{args min}} L(C_{oHP}, DS_{Tr}) + \rho(C_{oHP}) \quad (1)$$

$$\overline{oHP} = \operatorname{args\,min}_{oHP \in HP} L \frac{(C_{oHP}, DS_{Tr}), DS_{Te}}{\bar{C}(oHP)} \quad (2)$$

The population of spotted hyenas is initialized to begin the processing of the proposed model. Each hyena is defined as the combination of hyperparameters of CNN to be tuned. The fitness value of each hyena is calculated from the initialized population. The hyenas search for the best solution until the max number of iterations. The fitness of each hyena is checked after every iteration. The best value for a hyena is updated in the search space. A group of best solutions is maintained in the solution space. The procedure is terminated when the required condition is met or when the maximum number of repetitions has been reached. The algorithm for the proposed model is presented in Algorithm 1.

---

**Algorithm 1** SHO-CNN
 

---

**Require:** Size of population of Hyenas ( $H$ ),  $H(i = 1, 2, 3, \dots, n)$

**Ensure:** Fittest  $H_i$ , Every member in set  $H$  is a combination of hyperparameters  $\vec{hp}$  for the CNN, including learning rate, momentum, number of epochs, batch size, number of nodes, activation function, and dropout.

- 1: Initialize random  $H$  population of size  $n$
  - 2: Determine the fitness value of each search agent
  - 3: While (epoch < max number of iterations)
  - 4: For each  $H$
  - 5: Update the location of each search agent
  - 6: End for
  - 7: Update the control parameters  $\vec{X}$ ,  $\vec{Y}$ ,  $\vec{Distance}_{hp}$ , and  $\vec{hp}$
  - 8: Compute the fitness value of each  $H$  in a group
  - 9: If current solution  $\vec{L}_{hp}$  is better than the previous solution, update  $\vec{L}_{hp}$
  - 10: Update the group  $\vec{C}_{hp}$  with updated  $\vec{L}_{hp}$
  - 11: Epoch = Epoch + 1
  - 12: End while
- 

### 3.4. Tuning Hyperparameters

MLTC for news includes the classification of text data; therefore a one-dimensional CNN is utilized for experiments. The primary structure of the CNN is followed in the experiments, which contains convolutional layers at the start along with pooling layers, activation layers, and a fully connected layer at the end. A stack of five convolution layers is used in the experiments. A Rectified Linear Unit (ReLU) is the prevalent choice for the activation function in the case of multi-label text classification problems. The proposed model also utilizes the ReLU activation function. Max-pooling is applied as a pooling function in pooling layers. Adam is used as an optimizer. CNN has several hyperparameters that can be tuned to optimize the classification output. Before initiating the training process, the upper and lower limit value ranges of the parameters are assigned. The hyperparameters of the CNN and the parameter of SHO along with the selected range of values are represented in Table 2 and Table 3, respectively.

**Table 2.** Hyperparameters of CNN.

Hyperparameter	Value
Learning rate (LR)	Min. value 0.001, Max. value 0.1
Momentum (MOM)	[0.9 to 0.99]
Dropout	[0.1 to 0.9]
Epochs	[30,60,120,150,200,300]
Batch size (BS)	[8,16,32,64]
Number of nodes	(1,256)
Activation function	(ReLU, Sigmoid)

**Table 3.** Parameters of SHO.

Hyperparameter	Value
Size of population	30
Control parameter ( $\vec{h}_p$ )	(5,0)
Random vector $\vec{M}$	(0.5,1)
Number of iterations	200

#### 4. Evaluation and Results

This section describes the datasets, baseline models, and evaluation matrices used in the evaluation process along with a comparison of the results. The best values of the optimized hyperparameters for the CNN by the proposed model are also described in the section below.

##### 4.1. Experimental Setup

The development environment is based on Windows 10; system specifications are Intel(R) Core (TM) i5-1005G1 CPU @ 1.20 GHz–1.19 GHz with 16gb ram. A jupyter (<https://jupyter.org/>, accessed on 8 September 2022) notebook in VScode (<https://code.visualstudio.com/>, accessed on 8 September 2022) with Python (<https://www.python.org/>, accessed on 8 September 2022) programming is used to develop the proposed model. A Python package known as natural language toolkit NLTK (<https://www.nltk.org/>, accessed on 15 September 2022) is used for preprocessing data. The text is vectorized using the Keras (<https://keras.io/>, accessed on 15 September 2022) Pandas (<https://pandas.pydata.org/>, accessed on 15 September 2022) and NumPy (<https://numpy.org/>, accessed on 15 September 2022) are used for data manipulation and analysis.

A multi-label text classification model can be evaluated using a group of evaluation matrices to capture various related aspects. Accuracy, Precision, Recall, F-measure, MicroF1, and MacroF1 are used to evaluate and compare the proposed model [87].

##### 4.2. Datasets

Four standard publicly available datasets for the text-based news domain are used in this research. These datasets are a benchmark and have been used to evaluate several state-of-the-art machine learning models. Further details of each dataset are described below.

###### 4.2.1. RCV1-v2

RCV1-v2 is an updated version of Reuters corpus volume1. It has a collection of newswire articles produced by Reuters from 1996 to 1997. It contains 804,414 manually labeled news documents [46]. The news documents are categorized based on industries, topics, and regions.

###### 4.2.2. Reuters-21578

Reuters-21578 is the collection of news from one of the globally reputed news agencies, “Reuters”. The collection of news is from the Reuters newswire spanning 1987 [47]. We separate the dataset into 10,789 usable text articles and further divide it into a subset of 8631 data points for training and 2158 for testing.

###### 4.2.3. Slashdot

Slashdot is a dataset obtained from the Slashdot website, a technology-related news-sharing community platform [48]. The dataset contains a total of 291 label classes and 19,258 multi-labeled news articles for training and 4814 news articles for testing.

#### 4.2.4. NELA-GT-2019

NELA-GT-2019 is one of the latest datasets from the Harvard Dataverse. It is a large, multi-labeled news collection of around 1.12 million news articles [49]. It was obtained from 260 mainstream and alternative news sources from January 2019 to December 2019.

### 4.3. Baseline Models

The following baseline models are used for comparison with the proposed model.

#### 4.3.1. CNN

The results of the proposed model are compared with a CNN; experiments include a CNN network with five convolutional layers along with max pooling for the pooling layer functions.

#### 4.3.2. GA-CNN

The proposed model is also compared with a Genetic-Algorithm-optimized CNN (GA-CNN) [88]. GA is inspired by the theory of evolution and uses nature operations such as mutation, crossover, and selection. It determines and selects the fittest individual for the propagation of the search.

#### 4.3.3. PSO-CNN

PSO-CNN is a Particle swarm optimized CNN. PSO is a stochastic optimization technique; it employs the concept of social interaction to solve search problems. It iteratively uses several items to achieve the optimization goal [89].

#### 4.3.4. DE-CNN

DE-CNN is a CNN optimized by DE. DE is a population-based metaheuristics algorithm; it uses a large population of search agents to interact with each other and predict the best search in the solution space [90].

### 4.4. Results

In this paper, the SHO-CNN model for MLTC of news is proposed. SHO is utilized to optimize the CNN's hyperparameters. It is concluded that no single hyperparameter is responsible for CNN's optimization. A change in the value of one hyperparameter can have an effect on the values of other hyperparameters. Moreover, these optimal values are consistently dependent on the optimized values of other hyperparameters. SHO-CNN is evaluated using four benchmark datasets in the domain of MLTC of news: RCV1-v2, Reuters21578, Slashdot, and NELA-GT-2019. As this research aims to expand CNN's capabilities through the metaheuristic optimization of hyperparameters, the CNN is optimized using a variety of metaheuristic techniques to determine the optimal algorithm for optimization. The resultant best values for the hyperparameters of SHO-CNN are represented in Table 4.

**Table 4.** Best values of hyperparameters of SHO-CNN.

Dataset	Learning Rate	Momentum	Dropout	Epoch	Batch Size	Nodes	Activation Function
RCV1-v2	0.001	0.95	0.4	300	64	256	ReLU
Reuters21578	0.001	0.94	0.5	300	64	256	ReLU
Slashdot	0.001	0.95	0.4	300	128	256	ReLU
NELA-GT-2019	0.001	0.96	0.3	150	32	256	ReLU

The performance of the suggested SHO-CNN model is greater than that of the baseline approaches. It achieves accuracies of 93.65% and 90.81% for the RCV1-v2 and Reuters21578 datasets, respectively. The baseline model CNN achieves accuracies of 91.12%, 87.86%, 68.73%, and 95.45% for the RCV1-v2, Reuters21578, Slashdot, and NELA-GT-2019 datasets,

respectively. MLTC of news using the GA-CNN achieved accuracies of 92.32% and 88.34% for the RCV1-v2 and Reuters21578 datasets, respectively. The experiments with the PSO-CNN resulted in accuracies of 64.37% and 94.92% for the Slashdot and NELA-GT-2019 datasets, respectively. These results also confirm the study [79], which states that SHO optimization is more accurate than a basic CNN-, PSO-, or GA-optimized CNN. The results, along with models and evaluation matrices, are presented in Table 5.

**Table 5.** Multi-label text news dataset specifications.

Dataset	Model	Accuracy	Precision	Recall	f-Measure
RCV1-v2	CNN	0.911	0.902	0.799	0.851
	GA-CNN	0.923	0.921	0.905	0.918
	DE-CNN	0.925	0.92	0.921	0.928
	PSO-CNN	0.928	0.921	0.923	0.918
	SHO-CNN	0.936	0.930	0.935	0.931
Reuters-21578	CNN	0.878	0.875	0.866	0.871
	GA-CNN	0.883	0.881	0.885	0.888
	DE-CNN	0.889	0.888	0.884	0.886
	PSO-CNN	0.896	0.895	0.889	0.888
	SHO-CNN	0.908	0.902	0.909	0.901
Slashdot	CNN	0.571	0.501	0.564	0.530
	GA-CNN	0.592	0.591	0.590	0.590
	DE-CNN	0.627	0.620	0.615	0.617
	PSO-CNN	0.643	0.643	0.645	0.644
	SHO-CNN	0.687	0.683	0.682	0.682
NELA-GT-2019	CNN	0.917	0.914	0.903	0.908
	GA-CNN	0.923	0.922	0.914	0.918
	DE-CNN	0.939	0.931	0.939	0.935
	PSO-CNN	0.949	0.940	0.928	0.934
	SHO-CNN	0.954	0.950	0.951	0.951

A primary CNN is assessed together with an improved GA-CNN with a uniform mutation probability of 0.01, a crossover probability of 1, and a scaling factor of 0.5. PSO-CNN is also used in the evaluation process, with social, cognitive, and inertia constants of 1, 1, and 0.3, respectively. In addition, SHO-CNN with a control value of 1 and a random vector of 0.5 is also deployed. All optimization techniques use a population size of 30 with a maximum iteration count of 200.

As seen in Table 5, it is evident that SHO-CNN outperforms the other metaheuristic optimizations of CNNs for MLTC. Furthermore, SHO-CNN also achieves higher performance when compared to the baseline and other state-of-the-art approaches. LSTM, an upgraded version of CNN, is optimized using SHO [45], resulting in a higher hamming loss than SHO-CNN but superior micro-f1 and macro-f1 performance. Table 6 shows a comparison of the proposed SHO-CNN model with various state-of-the-art approaches on the RCV1-v2, Reuters21578, Slashdot, and NELA-GT-2019 datasets.

**Table 6.** Comparison between proposed model and similar research.

Dataset	Model	Hamming Loss	Micro-f1	Macro-f1	
RCV1-v2	BR [91]	0.864	0.852	0.662	
	CC [92]	0.872	0.855	0.682	
	LP [93]	0.874	0.858	0.691	
	CNN [25]	0.891	0.849	0.727	
	BERT [94]	0.733	0.877	0.667	
	CNN-RNN [95]	0.872	0.849	0.752	
	SGM [96]	0.814	0.869	0.712	
	SGM-GE [97]	0.753	0.878	0.762	
	Seq2Set [98]	0.736	0.879	0.751	
	ML-R [50]	0.793	0.898	0.695	
	Seq2Tree [99]	0.732	0.868	0.700	
	TextRCNN [100]	0.765	0.815	0.592	
	HiAGM [100]	0.768	0.839	0.633	
	HTCInfoMax [101]	0.748	0.835	0.627	
	HiMatch [102]	0.754	0.847	0.641	
SHO-LSTM [45]	0.737	0.913	0.781		
SHO-CNN (Proposed)	0.722	0.906	0.776		
Reuters21578	BR [91]	0.434	0.835	0.402	
	PCC [103]	0.442	0.834	0.398	
	seq2seq-RNN [104]	0.384	0.858	0.437	
	BERT [94]	0.261	0.905	0.640	
	LBA [105]	0.254	0.908	0.675	
	ML-KNN [106]	0.376	0.732	0.253	
	ML-ARAM [61]	0.363	0.626	0.163	
	ML-RNN [104]	0.332	0.858	0.457	
	LaMP [107]	0.291	0.889	0.560	
	MPVAE [108]	0.267	0.887	0.542	
	MrMP [109]	0.254	0.893	0.591	
	Class-wise cluster + FS [110]	-	0.800	0.725	
	CB-NTR [111]	0.247	0.907	0.633	
	SHO-LSTM [45]	0.240	0.910	0.690	
	SHO-CNN (Proposed)	0.237	0.908	0.665	
Slashdot	BR [91]	0.052	0.486	0.362	
	BR-Support [112]	0.055	0.516	0.357	
	PCC [103]	0.056	0.480	0.279	
	seq2seq-RNN [104]	0.058	0.528	0.270	
	set-RNN [112]	0.053	0.538	0.310	
	BERT [94]	0.037	0.583	0.380	
	LBA [105]	0.038	0.582	0.410	
	CC [92]	0.057	0.480	0.373	
	CNN [25]	0.049	0.512	0.412	
	CNN-RNN [95]	0.046	0.530	0.469	
	MAGNET [112]	0.039	0.568	0.475	
	SHO-LSTM [45]	0.038	0.650	0.492	
	SHO-CNN (Proposed)	0.037	0.632	0.453	
	NELA-GT-2019	BR [91]	0.033	0.870	0.792
		BR-Support [112]	0.040	0.846	0.783
PCC [103]		0.009	0.922	0.913	
LBA [105]		0.008	0.953	0.9420	
CNN [25]		0.008	0.934	0.926	
CNN-RNN [95]		0.008	0.935	0.944	
SHO-LSTM [45]		0.007	0.972	0.970	
SHO-CNN (Proposed)		0.006	0.963	0.956	

According to the mathematical model of the SHO algorithm, to find a suitable prey  $\vec{L}_p$ , the factor  $|Y| > 1$  facilitates the spotted hyenas to move away from the prey. This promotes exploration of the search space, which leads to finding diverse CNN structures during

optimization. The vector  $M$  provides random values for exploration; this mechanism is very beneficial for avoiding local optima problems even when the SHO algorithm is in the exploitation phase [88].

The results show that the proposed SHO-CNN model can perform better in real-time applications. The distinguishing factor in the proposed work is the introduction of SHO in a CNN model. As per our knowledge, this is the first case where an SHO-optimized CNN architecture is implemented for multi-label news classification. Furthermore, SHO-CNN achieves the best classification accuracies among all the baseline and modern techniques under consideration. The result of the proposed model demonstrates the significance of CNN in the domain of MLTC. This proposed algorithm has a very fast convergence rate and can successfully reduce the problem of being trapped in local minima while training the CNN. It also illustrates the capabilities of SHO for optimizing the hyperparameters of CNN.

Transformer encoders are best if computation is not a concern. When classes are fairly distributed, a CNN is a successful model due to its diligent performance and assessment efficiency. A CNN can perform better even without pre-trained word embeddings. The CNN model performed faster; on average, the CNN was 17.6% quicker than RNN, 91.3% faster than the Transformer encoder, and 94.7% faster than BERT-Base. BERT-Base is the slowest model due to the huge number of top-layer parameters [113].

More recently, CNNs have garnered attention for NLP tasks due to their higher performance, particularly on lengthier texts [114–117]. This is mostly due to the fact that CNNs are easier to train than other NLP methods. In NLP applications, CNNs use a convolutional layer with only one-dimension, which collects information from neighboring words. The number of filters in a one-dimensional convolutional layer considers  $n$  number of words next to one other as one ( $n$ -gram) [118–120]. Furthermore, adding a higher  $n$ -gram method to CNN algorithms does not significantly increase the computing cost [113].

#### 4.5. Ablation Study

We further classified CNN hyperparameters into two groups: group A includes critical hyperparameters such as LR, mom, number of nodes, and activation function. Group B comprises advanced hyperparameters such as dropout, epoch number, and batch size, which are thought to have a significant impact on performance. Initially, we evaluated the suggested SHO-CNN model with only Group-A hyperparameters, which resulted in a significant performance improvement compared to CNN's baseline performance. Secondly, we analyzed SHO-CNN with Group-B hyperparameters; the results show that optimizing only the advanced hyperparameters is insufficient for achieving high-performance results. The comparisons of hyperparameters from groups A and B are shown in Figure 3. According to the ablation study, the vital or advanced hyperparameters alone are not adequate for achieving high performance; rather, a mix of these hyperparameters is the optimum approach for improved performance.

#### 4.6. Discussion

The suggested SHO-CNN model outperforms the baseline and state-of-the-art approaches, as determined by experimentation. The selection of hyperparameter values plays an important part in the performance of the deep learning model. Selecting hyperparameters using a thorough hyperparameter search is computationally infeasible. Furthermore, selecting a random initialization for hyperparameters could result in subpar performance. Moreover, the choice of hyperparameter values is dependent on the dataset. This section compares SHO-CNN to the current state-of-the-art in terms of convergence rate, epoch, batch size, and learning rate.

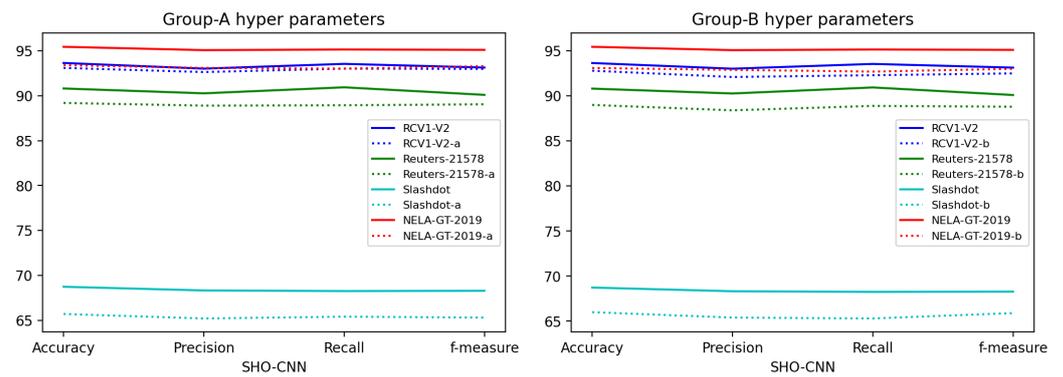


Figure 3. Comparisons based on ablation study.

### 4.6.1. Convergence Rate

The convergence rate for the proposed model is compared with the other state-of-the-art optimization techniques using RCV1-v2, Reuters-21578, Slashdot, and NELA-GT-2019 datasets. In Figure 4, the x-axis represents the number of iterations, and the y-axis represents a measure of fitness. The proposed model appears highly convergent and outperforms other techniques in vogue. The results also indicate that DE has a higher convergence rate when compared to GA and PSO for both datasets. The performance of DE and PSO are relatively similar, and they both performed better than GA and a simple CNN.

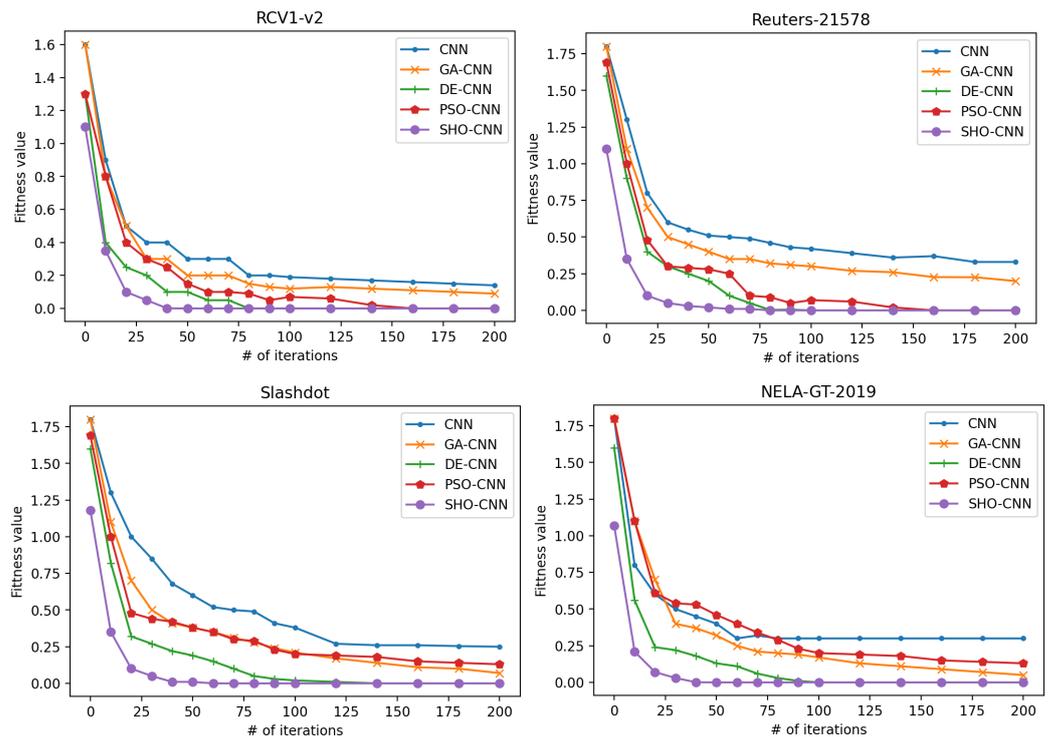
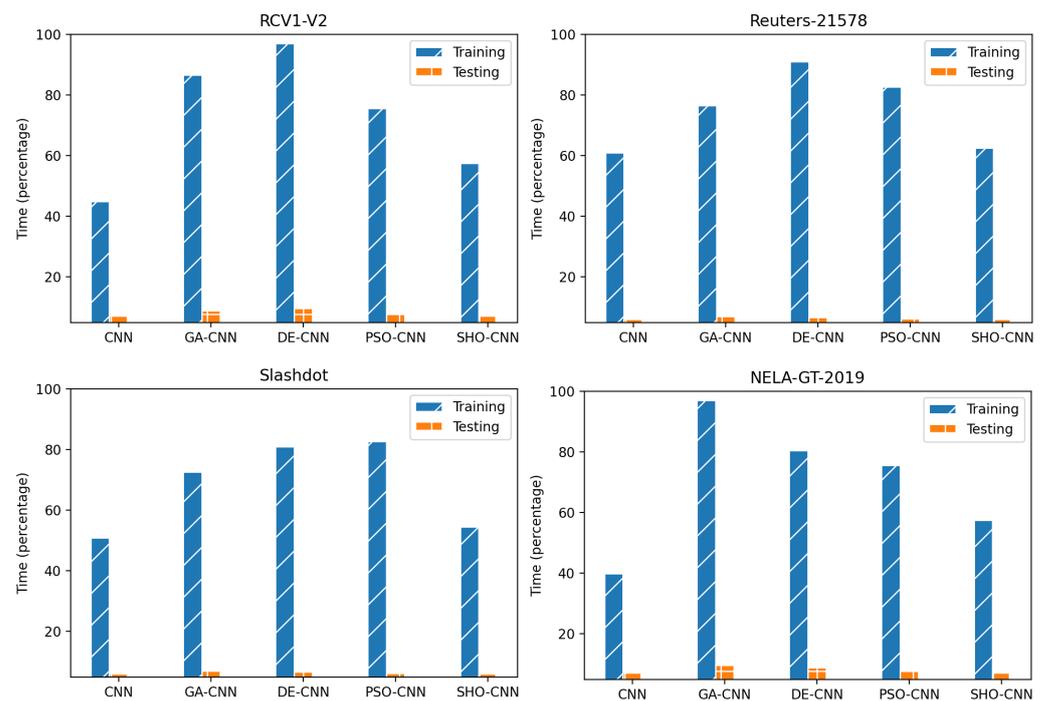


Figure 4. Evaluations based on convergence rate.

### 4.6.2. Execution Time

Using the RCV1-v2, Reuters-21578, Slashdot, and NELA-GT-2019 datasets, the execution time of the proposed model is compared to that of existing cutting-edge optimization strategies. The x-axis of Figure 5 reflects the optimization strategies used to optimize CNN alongside a primary CNN, while the y-axis represents the percentage of time. The findings reveal that the proposed model's efficiency is fairly similar to that of a standard CNN, but the proposed model has faster convergence and better performance than a standard CNN. The results also reveal that PSO is superior to GA and DE in terms of performance.



**Figure 5.** Evaluations based on execution time.

#### 4.6.3. Epoch

The evaluation of the proposed SHO-CNN model based on epoch is represented in Figure 6. The results reveal that SHO-CNN initiated at an accuracy of 87.5% and acquired a higher accuracy of 93.65% after 300 epochs for RCV1-v2. This optimal number of epochs provided by SHO resulted in encountering the underfitting problem of CNNs. It is evident from the results that SHO-CNN achieved the highest accuracy. A plain CNN, without any optimization, begins at 84.9% and 82.4% accuracy for RCV1-v2 and Reuters-21578, respectively, which is the lowest amongst the group. In the case of Reuters-21578, the proposed model also surpassed all the other mentioned optimization techniques. The proposed SHO-CNN model initiated at 86.5% and achieved 90.81% accuracy with 300 epochs for Reuters-21578. It also achieved an accuracy of 95.45% with 150 epochs for the NELA-GT-2019 dataset.

#### 4.6.4. Batch Size

Evaluation of the proposed SHO-CNN model based on the selection of batch size is represented in Figure 7. Optimal batch size decreases noise in the gradients, hence enhancing the classification accuracy. The evaluation process included batch sizes of 8, 16, 32, 64, and 128. The accuracies of the proposed model are 87.5%, 90.9%, 92.3%, and 93.65% for RCV1-v2 at batch sizes of 8, 16, 32, and 64, respectively. The proposed SHO-CNN model yields maximum accuracies for the RCV1-V2, Reuters-21578, Slashdot, and NELA-GT-2019 datasets at batch sizes of 64, 64, 128, and 32, respectively.

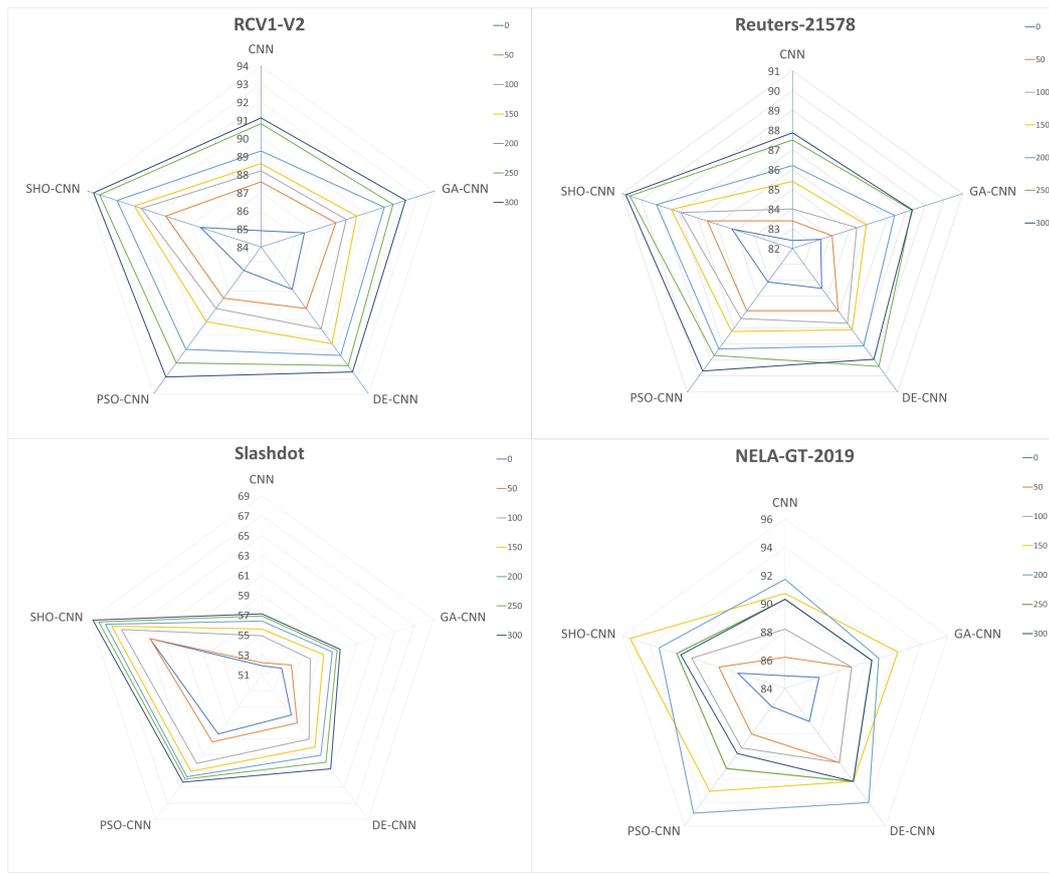


Figure 6. Evaluations based on epochs.

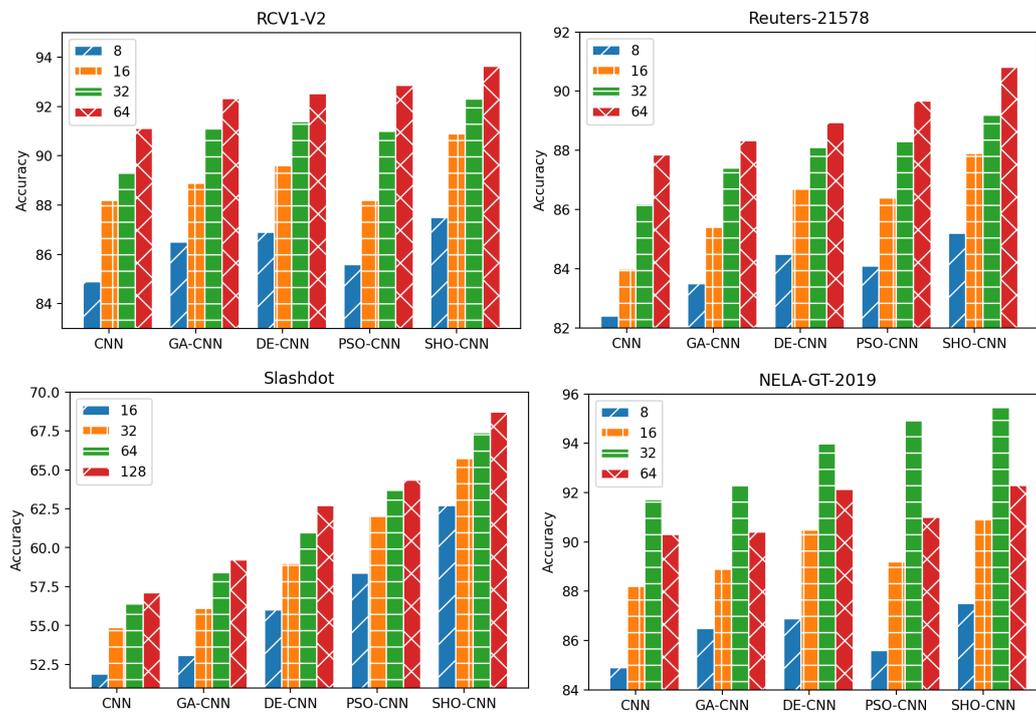
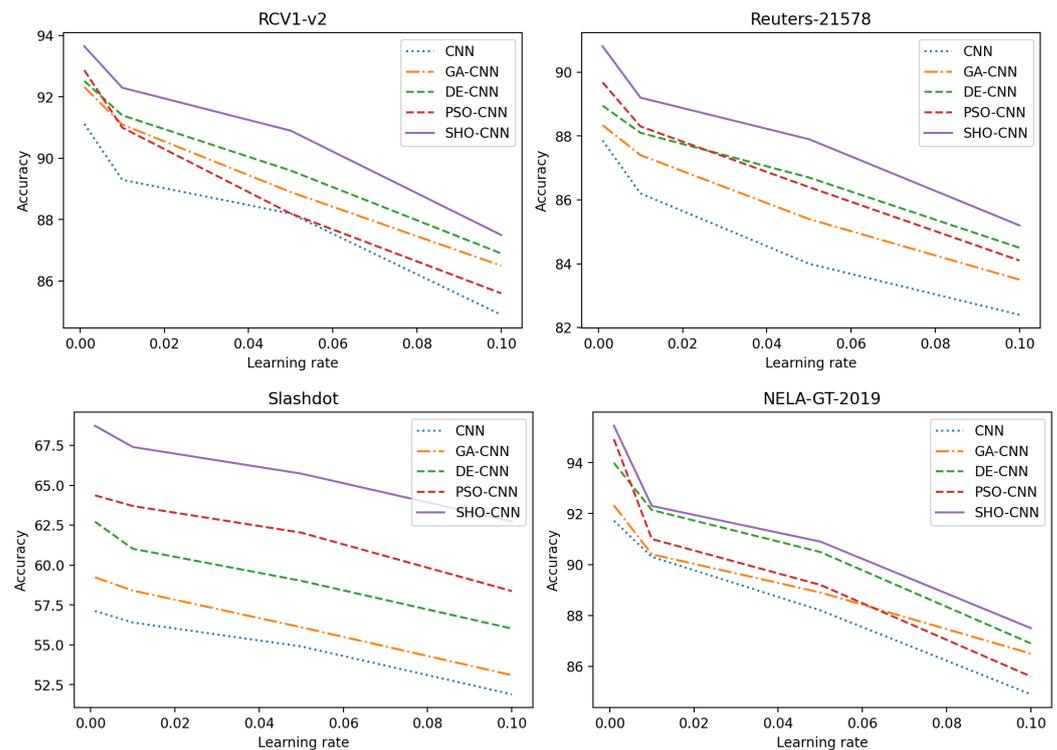


Figure 7. Evaluations based on batch size.

#### 4.6.5. Learning Rate

The proposed SHO-CNN model, along with CNN, GA-CNN, DE-CNN, and PSO-CNN are also evaluated based on varying learning rates (Figure 8). Exploring the optimal learning rate is the critical factor for a model's best performance. The learning rate is correlated with other hyperparameters of a CNN, including momentum and batch size. The evaluation range for the learning rate is set to 0.001, 0.01, 0.05, and 0.1. The accuracies of the proposed model were 93.65%, 92.3%, 90.9%, and 87.5% accuracy for RCV1-v2 at learning rates of 0.001, 0.01, 0.05, and 0.1, respectively. SHO-CNN achieved the highest accuracies for the RCV1-V2, Reuters-21578, Slashdot, and NELA-GT-2019 datasets at a learning rate of 0.001.



**Figure 8.** Evaluations based on learning rate.

In this research, a basic CNN is used in the proposed model to capture spatial correlations and hierarchical structures of text with exceptional accuracy without increasing the model's complexity [114,115,120]. The classification of text employs a one-dimensional CNN, resulting in the fastest execution and testing times [113,116,117]. Adding more filters does not significantly extend the execution time [27,119,120]. Five convolutional layers are utilized by SHO-CNN to extract high-level abstract properties [22]. Furthermore, the CNN is optimized through numerous optimization strategies; the inclusion of SHO enhances the variety of search solutions for optimizing hyperparameters while preventing the occurrence of local minima [44,45].

Although the SHO-optimized CNN performed well for the task of multi-label news text classification, there are some limitations of this research that need to be considered. Firstly, this study was conducted solely using the RCV1-V2, Reuters21578, Slashdot, and NELA-GT-2019 datasets, so it may not be generalized for other text classification tasks. Secondly, it is observed that SHO is occasionally susceptible to local optima when using a lesser number of repetitions. More recently proposed optimization algorithms (such as Political Optimizer [121], Heap-based Optimizer [122], Harris Hawks Optimizer [84]) can be tried for the optimization of CNNs. The classification of SHO-CNN is solely based on news text; combining text and images can generate good results.

## 5. Conclusions and Future Research

This research proposes an expert automated method based on SHO of CNNs for the purpose of MLTC of news. SHO is a metaheuristic-based approach that provides higher exploration of the solution space while avoiding the problem of becoming trapped in local optima. SHO is used in the proposed model to tune the CNN hyperparameters for improved classification performance. Furthermore, the CNN algorithm helps to improve the performance by spontaneously extracting the discriminant features from the text. As a result, the proposed model, SHO-CNN, is highly convergent and best-suited for multi-label classification. The proposed model is evaluated and appraised on four standard benchmark news datasets. SHO-CNN addresses the complexity of deep learning models by utilizing a primary CNN to achieve optimal performance compared to various baseline models, state-of-the-art models, and other optimization techniques for CNN. The results indicate that the proposed model has accuracies of 93.65%, 90.81%, 68.73%, and 95.45% for the RCV1-V2, Reuters-21578, Slashdot, and NELA-GT-2019 datasets, respectively. The results also reveal that SHO-CNN enhances a basic CNN's accuracies on the RCV1-V2, Reuters-21578, Slashdot, and NELAGT-2019 datasets by 2.5%, 2.95%, 11.6%, and 3.7%, respectively. Future studies should investigate optimization approaches for different DL models. A multimodal approach to the classification of news is also a viable future strategy, given the majority of news is now disseminated via the internet, particularly social media. In addition, a future study will incorporate a model with the capacity to classify multilingual news articles.

**Author Contributions:** Conceptualization, M.I.N. and K.A.; methodology, M.I.N. and K.A.; software, H.N.; validation, A.A.; A.Y.M. and Z.Z.; formal analysis, K.A. and H.A.H.; investigation, D.L. and A.Y.M.; resources, Z.Z.; data curation, H.A.H.; writing—original draft preparation, M.I.N.; writing—review and editing, K.A. and D.L.; visualization, M.I.N. and H.N.; supervision, D.L.; project administration, A.A.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
NLP	Natural Language Processing
ML	Machine Learning
DL	Deep Learning
Conv	Convolutional Layer
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
SHO	Spotted Hyena Optimizer
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
GS	Grid Search
DE	Differential Evolution
GWO	Grey Wolf Optimizer
ICA	Imperialist Competitive Algorithm

## References

1. Hu, T.; Wang, H.; Yin, W. Multi-label news classification algorithm based on deep bi-directional classifier chains. *J. Zhejiang Univ. (Eng. Sci.)* **2019**, *53*, 2110–2117.
2. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
3. Al-Sarem, M.; Alsaeedi, A.; Saeed, F.; Boulila, W.; AmeerBakhsh, O. A novel hybrid deep learning model for detecting COVID-19-related rumors on social media based on LSTM and concatenated parallel CNNs. *Appl. Sci.* **2021**, *11*, 7940. [[CrossRef](#)]
4. Moujahid, H.; Cherradi, B.; Al-Sarem, M.; Bahatti, L. Diagnosis of COVID-19 disease using convolutional neural network models based transfer learning. In Proceedings of the International Conference of Reliable Information and Communication Technology, Langkawi, Malaysia, 21–22 December 2020; Springer International Publishing: Cham, Switzerland, 2021; Volume 72, pp. 148–159.
5. Gannour, E.; Oussama, H.; Hamida, S.; Cherradi, B.; Al-Sarem, M.; Raihani, A.; Saeed, F.; Hadwan, M. Concatenation of Pre-Trained Convolutional Neural Networks for Enhanced COVID-19 Screening Using Transfer Learning Technique. *Electronics* **2021**, *11*, 103. [[CrossRef](#)]
6. Al-Sarem, M.; Saeed, F.; Boulila, W.; Emar, A.H.; Al-Mohaimed, M.; Errais, M. Feature selection and classification using CatBoost method for improving the performance of predicting Parkinson's disease. In *Advances on Smart and Soft Computing*; Springer: Singapore, 2021; pp. 189–199.
7. Li, D.; Ahmed, K.; Zheng, Z.; Mohsan, S.A.H.; Alsharif, M.H.; Hadjouni, M.; Jamjoom, M.M.; Mostafa, S.M. Roman Urdu Sentiment Analysis Using Transfer Learning. *Appl. Sci.* **2022**, *12*, 10344. [[CrossRef](#)]
8. Ahmed, K.; Nadeem, M.I.; Li, D.; Zheng, Z.; Ghadi, Y.Y.; Assam, M.; Mohamed, H.G. Exploiting Stacked Autoencoders for Improved Sentiment Analysis. *Appl. Sci.* **2022**, *12*, 12380. [[CrossRef](#)]
9. Mittal, V.; Gangodkar, D.; Pant, B. Deep Graph-Long Short-Term Memory: A Deep Learning Based Approach for Text Classification. *Wirel. Pers. Commun.* **2021**, *119*, 2287–2301. [[CrossRef](#)]
10. Liao, W.; Wang, Y.; Yin, Y.; Zhang, X.; Ma, P. Improved sequence generation model for multi-label classification via CNN and initialized fully connection. *Neurocomputing* **2020**, *382*, 188–195. [[CrossRef](#)]
11. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [[CrossRef](#)]
12. Zhan, H.; Lyu, S.; Lu, Y.; Pal, U. DenseNet-CTC: An end-to-end RNN-free architecture for context-free string recognition. *Comput. Vis. Image Underst.* **2021**, *204*, 103168. [[CrossRef](#)]
13. Pan, S.; Guan, H.; Chen, Y.; Yu, Y.; Gonçalves, W.N.; Junior, J.M.; Li, J. Land-cover classification of multispectral LiDAR data using CNN with optimized hyper-parameters. *ISPRS J. Photogramm. Remote. Sens.* **2020**, *166*, 241–254. [[CrossRef](#)]
14. Kousalya, K.; Saranya, T. Improved the detection and classification of breast cancer using hyper parameter tuning. *Mater. Today Proc.* **2021**, in press. [[CrossRef](#)]
15. Wang, Y.; Zhang, H.; Zhang, G. cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. *Swarm Evol. Comput.* **2019**, *49*, 114–123. [[CrossRef](#)]
16. Wang, T.; Liu, L.; Liu, N.; Zhang, H.; Zhang, L.; Feng, S. A multi-label text classification method via dynamic semantic representation model and deep neural network. *Appl. Intell.* **2020**, *50*, 2339–2351. [[CrossRef](#)]
17. Hijazi, S.; Kumar, R.; Rowen, C. *Using Convolutional Neural Networks for Image Recognition*; Cadence Design Systems Inc.: San Jose, CA, USA, 2015; pp. 1–12.
18. Li, Q.; Cai, W.; Wang, X.; Zhou, Y.; Feng, D.D.; Chen, M. Medical image classification with convolutional neural network. In Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 10–12 December 2014; pp. 844–848.
19. Fasihi, M.; Nadimi-Shahraki, M.H.; Jannesari, A. A Shallow 1-D Convolution Neural Network for Fetal State Assessment Based on Cardiotocogram. *SN Comput. Sci.* **2021**, *2*, 287. [[CrossRef](#)]
20. Fasihi, M.; Nadimi-Shahraki, M.H.; Jannesari, A. Multi-Class Cardiovascular Diseases Diagnosis from Electrocardiogram Signals using 1-D Convolution Neural Network. In Proceedings of the 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI), Las Vegas, NV, USA, 11–13 August 2020; pp. 372–378.
21. Lee, T. EMD and LSTM Hybrid Deep Learning Model for Predicting Sunspot Number Time Series with a Cyclic Pattern. *Sol. Phys.* **2020**, *295*, 82. [[CrossRef](#)]
22. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A Convolutional Neural Network for Modelling Sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 23–25 June 2014; pp. 655–665. [[CrossRef](#)]
23. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the International Conference on Machine Learning (ICML), Helsinki, Finland, 5–9 July 2008; pp. 160–167.
24. Yu, L.; Hermann, K.M.; Blunsom, P.; Pulman, S. Deep learning for answer sentence selection. In Proceedings of the Advances in Neural Information Processing Systems (NIPS) Workshop, Montreal, QC, Canada, 8–13 December 2014.
25. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
26. Yin, W.; Schütze, H. Multichannel variable-size convolution for sentence classification, In Proceedings of the Conference on Natural Language Learning (CoNLL), Beijing, China, 30–31 July 2015; pp. 204–214.

27. Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y. Very deep convolutional networks for natural language processing. *arXiv* **2016**, arXiv:1606.01781.
28. Gu, J.; Wang, G.; Cai, J.; Chen, T. An empirical study of language cnn for image captioning. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
29. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; pp. 933–941
30. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [[CrossRef](#)]
31. Divya, S.; Kiran, E.L.N.; Rao, M.S.; Vemulapati, P. Prediction of gene selection features using improved multi-objective spotted hyena optimization algorithm. In *Data Communication and Networks. Advances in Intelligent Systems and Computing*; Jain, L., Tsihrantzis, G., Balas, V., Sharma, D., Eds.; Springer: Singapore, 2020; Volume 1049.
32. Akyol, S.; Alatas, B. Plant intelligence based metaheuristic optimization algorithms. *Artif. Intell. Rev.* **2017**, *47*, 417–462. [[CrossRef](#)]
33. Alatas, B.; Bingol, H. Comparative Assessment of Light-Based Intelligent Search and Optimization Algorithms. *Light Eng.* **2020**, *28*, 51–59. [[CrossRef](#)]
34. Abdel-Basset, M.; Abdel-Fatah, L.; Sangaiah, A.K. Metaheuristic algorithms: A comprehensive review. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*; Academic Press: New York, NY, USA, 2018; pp. 185–231.
35. Wang, P.; Zhou, Y.; Luo, Q.; Han, C.; Lei, M. Complex-valued Encoding Metaheuristic Optimization Algorithm: A Comprehensive Survey. *Neurocomputing* **2020**, *407*, 313–342. [[CrossRef](#)]
36. Hakim, W.L.; Rezaie, F.; Nur, A.S.; Panahi, M.; Khosravi, K.; Lee, C.-W.; Lee, S. Convolutional neural network (CNN) with metaheuristic optimization algorithms for landslide susceptibility mapping in Icheon, South Korea. *J. Environ. Manag.* **2022**, *305*, 114367. [[CrossRef](#)] [[PubMed](#)]
37. Kadry, S.; Srivastava, G.; Rajinikanth, V.; Rho, S.; Kim, Y. Tuberculosis Detection in Chest Radiographs Using Spotted Hyena Algorithm Optimized Deep and Handcrafted Features. *Comput. Intell. Neurosci.* **2022**, *2022*, 9263379. [[CrossRef](#)] [[PubMed](#)]
38. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [[CrossRef](#)]
39. Pangle, W.M.; Holekamp, K.E. Functions of vigilance behaviour in a social carnivore, the spotted hyaena, *Crocuta crocuta*. *Anim. Behav.* **2010**, *80*, 257–267. [[CrossRef](#)]
40. Yirga, G.; Ersino, W.; De longh, H.H.; Leirs, H.; Gebrehiwot, K.; Deckers, J.; Bauer, H. Spotted hyena (*Crocuta crocuta*) coexisting at high density with people in Wukro district, Northern Ethiopia. *Mamm. Biol.* **2013**, *78*, 193–197. [[CrossRef](#)]
41. Dhiman, G.; Kaur, A. Spotted hyena optimizer for solving engineering design problems. In Proceedings of the 2017 International Conference on Machine Learning and Data Science (MLDS), Noida, India, 14–15 December 2017; pp. 114–118. [[CrossRef](#)]
42. Luo, Q.; Li, J.; Zhou, Y. Spotted hyena optimizer with lateral inhibition for image matching. *Multimed. Tools Appl.* **2019**, *78*, 34277–34296. [[CrossRef](#)]
43. Panda, N.; Majhi, S.K. Improved spotted hyena optimizer with space transformational search for training pi-sigma higher order neural network. *Comput. Intell.* **2020**, *36*, 320–350. [[CrossRef](#)]
44. Zhou, G.; Li, J.; Li, Z.; Tang, Q.; Luo, Y.; Zhou, Y. An improved spotted hyena optimizer for pid parameters in an avr system. *Math. Biosci. Eng.* **2020**, *17*, 3767–3783. [[CrossRef](#)]
45. Khataei Maragheh, H.; Gharehchopogh, F.S.; Majidzadeh, K.; Sangar, A.B. A New Hybrid Based on Long Short-Term Memory Network with Spotted Hyena Optimization Algorithm for Multi-Label Text Classification. *Mathematics* **2022**, *10*, 488. [[CrossRef](#)]
46. Lewis, D. D.; Yang, Y.; Rose, T.; Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.* **2004**, *5*, 361–397.
47. UCI Machine Learning Repository: Reuters-21578 Text Categorization Collection Data Set. Available online: <https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection> (accessed on 12 September 2022).
48. Leskovec, J.; Lang, K.; Dasgupta, A.; Mahoney, M. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *Internet Math.* **2009**, *6*, 29–123. [[CrossRef](#)]
49. Gruppi, M.; Horne, B.D.; Adali, S. NELA-GT-2019: A Large Multi-Labelled News Dataset for the Study of Misinformation in News Articles. *arXiv* **2020**, arXiv:2003.08444.
50. Wang, R.; Ridley, R.; Su, X.; Qu, W.; Dai, X. A novel reasoning mechanism for multi-label text classification. *Inf. Process. Manag.* **2021**, *58*, 102441. [[CrossRef](#)]
51. Omar, A.; Mahmoud, T.M.; Abd-El-Hafeez, T.; Mahfouz, A. Multi-label Arabic text classification in Online Social Networks. *Inf. Syst.* **2021**, *100*, 101785. [[CrossRef](#)]
52. Udandarao, V.; Agarwal, A.; Gupta, A.; Chakraborty, T. InPHYNet: Leveraging attention-based multitask recurrent networks for multi-label physics text classification. *Knowl.-Based Syst.* **2021**, *211*, 106487. [[CrossRef](#)]
53. Ciarelli, P.M.; Oliveira, E.; Salles, E.O.T. Multi-label incremental learning applied to web page categorization. *Neural Comput. Appl.* **2014**, *24*, 1403–1419. [[CrossRef](#)]
54. Yao, L.; Sheng, Q.Z.; Ngu, A.H.H.; Gao, B.J.; Li, X.; Wang, S. Multi-label classification via learning a unified object-label graph with sparse representation. *World Wide Web.* **2016**, *19*, 1125–1149. [[CrossRef](#)]
55. Ghiandoni, G.M.; Bodkin, M.J.; Chen, B.; Hristozov, D.; Wallace, J.E.A.; Webster, J.; Gillet, V.J. Enhancing reaction-based de novo design using a multi-label reaction class recommender. *J. Comput.-Aided Mol. Des.* **2020**, *34*, 783–803. [[CrossRef](#)]

56. Laghmari, K.; Marsala, C.; Ramdani, M. An adapted incremental graded multi-label classification model for recommendation systems. *Prog. Artif. Intell.* **2018**, *7*, 15–29. [CrossRef]
57. Zou, Y.-p.; Ouyang, J.-h.; Li, X.-m. Supervised topic models with weighted words: Multi-label document classification. *Front. Technol. Electron. Eng.* **2018**, *19*, 513–523. [CrossRef]
58. Li, X.; Ouyang, J.; Zhou, X. Labelset topic model for multi-label document classification. *J. Intell. Inf. Syst.* **2016**, *46*, 83–97. [CrossRef]
59. Wang, B.; Hu, X.; Li, P.; Yu, P.S. Cognitive structure learning model for hierarchical multi-label text classification. *Knowl.-Based Syst.* **2021**, *218*, 106876. [CrossRef]
60. Ibrahim, M.A.; Ghani Khan, M.U.; Mehmood, F.; Asim, M.N.; Mahmood, W. GHS-NET a generic hybridized shallow neural network for multi-label biomedical text classification. *J. Biomed. Inform.* **2021**, *116*, 103699. [CrossRef]
61. Benites, F.; Sapozhnikova, E. HARAM: A Hierarchical ARAM Neural Network for Large-Scale Text Classification. In Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA, 14–17 November 2015; pp. 847–854. [CrossRef]
62. Chen, Z.; Ren, J. Multi-label text classification with latent word-wise label information. *Appl. Intell.* **2021**, *51*, 966–979. [CrossRef]
63. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Handwritten digit recognition with a back-propagation network. In Proceedings of the Annual Conference on Advances in Neural Information Processing Systems, Denver, CO, USA, 27–30 November 1989; pp. 396–404.
64. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
65. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
66. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
67. Bellman, R.E. *Adaptive Control Processes: A Guided Tour*; Princeton University Press: Princeton, NJ, USA, 2015; Volume 2045.
68. Larochelle, H.; Erhan, D.; Courville, A.; Bergstra, J.; Bengio, Y. An empirical evaluation of deep architectures on problems with many factors of variation. In Proceedings of the 24th International Conference on Machine Learning (ICML 2007), Corvallis, OR, USA, 20–24 June 2007; pp. 473–480.
69. Bergstra, J.S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. In Proceedings of the NIPS'11: 24th International Conference on Neural Information Processing Systems, Granada, Spain, 12–15 December 2011; pp. 2546–2554.
70. Hutter, F.; Hoos, H.H.; Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In Proceedings of the International Conference on Learning and Intelligent Optimization, Rome, Italy, 17–21 January 2011; pp. 507–523.
71. Hoffman, M.W.; Shahriari, B. Modular Mechanisms for Bayesian Optimization. 2014; pp. 1–5. Available online: [https://www.mwhoffman.com/papers/hoffman\\_2014b.pdf](https://www.mwhoffman.com/papers/hoffman_2014b.pdf) (accessed on 12 September 2022).
72. Yao, X.; Liu, Y. A new evolutionary system for evolving artificial neural networks. *IEEE Trans. Neural Netw.* **1997**, *8*, 694–713. [CrossRef]
73. Stanley, K.O.; Miikkulainen, R. Evolving neural networks through augmenting topologies. *Evol. Comput.* **2002**, *10*, 99–127. [CrossRef]
74. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G. Evolving deep convolutional neural networks for image classification. *IEEE Trans. Evol. Comput.* **2020**, *24*, 394–407. [CrossRef]
75. Ma, B.; Li, X.; Xia, Y.; Zhang, Y. Autonomous deep learning: A genetic DCNN designer for image classification. *Neurocomputing* **2020**, *379*, 152–161. [CrossRef]
76. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G. A particle swarm optimization-based flexible convolutional autoencoder for image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2295–2309. [CrossRef] [PubMed]
77. Wang, B.; Sun, Y.; Xue, B.; Zhang, M. Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
78. Wang, B.; Sun, Y.; Xue, B.; Zhang, M. A hybrid differential evolution approach to designing deep convolutional neural networks for image classification. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Wellington, New Zealand, 11–14 December 2018; pp. 237–250.
79. Lee, W.-Y.; Park, S.-M.; Sim, K.-B. Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. *Optik* **2018**, *172*, 359–367. [CrossRef]
80. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press, Cambridge, MA, USA, 1998; Volume 135.
81. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
82. Baker, B.; Gupta, O.; Naik, N.; Raskar, R. Designing neural network architectures using reinforcement learning. *arXiv* **2016**, arXiv:1611.02167.
83. Neary, P. Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning. In Proceedings of the 2018 IEEE International Conference on Cognitive Computing (ICCC), San Francisco, CA, USA, 2–7 July 2018; pp. 73–77.
84. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]

85. Kuang, S.; Davison, B.D. Learning class-specific word embeddings. *J. Supercomput.* **2020**, *76*, 8265–8292. [[CrossRef](#)]
86. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. In Proceedings of the Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–10 December 2015, pp. 649–657.
87. Hossin, M.; Sulaiman, M.N. A review on evaluation metrics for data classification evaluations. *Int. J. Data Min. Knowl. Manag. Process.* **2015**, *5*, 1.
88. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
89. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 1994, pp. 1942–1948.
90. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
91. Boutell, M.R.; Luo, J.; Shen, X.; Brown, C.M. Learning multilabel scene classification. *Pattern Recognit.* **2004**, *37*, 1757–1771. [[CrossRef](#)]
92. Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier chains for multi-label classification. *Mach. Learn.* **2011**, *85*, 333. [[CrossRef](#)]
93. Tsoumakas, G.; Katakis, I. Multi-label classification: An overview. *Int. J. Data Warehous. Min.* **2007**, *3*, 1–13. [[CrossRef](#)]
94. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
95. Chen, G.; Ye, D.; Xing, Z.; Chen, J.; Cambria, E. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2377–2383.
96. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016.
97. Yang, P.; Sun, X.; Li, W.; Ma, S.; Wu, W.; Wang, H. Sgm: Sequence generation model for multi-label classification. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 21–25 August 2018; pp. 3915–3926.
98. Yang, P.; Luo, F.; Ma, S.; Lin, J.; Sun, X. A deep reinforced sequence-to-set model for multi-label classification. In Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5252–5258.
99. Yu, C.; Shen, Y.; Mao, Y.; Cai, L. Constrained Sequence-to-Tree Generation for Hierarchical Text Classification. *arXiv* **2022**, arXiv:2204.00811.
100. Zhou, J.; Ma, C.; Long, D.; Xu, G.; Ding, N.; Zhang, H.; Xie, P.; Liu, G. Hierarchy-aware global model for hierarchical text classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 1106–1117.
101. Deng, Z.; Peng, H.; He, D.; Li, J.; Yu, P.S. Htcinfomax: A global model for hierarchical text classification via information maximization. *arXiv* **2021**, arXiv:2104.05220.
102. Chen, H.; Ma, Q.; Lin, Z.; Yan, J. . Hierarchy-aware label semantics matching network for hierarchical text classification. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual, 1–6 August 2021; Volume 1, pp. 4370–4379.
103. Dembczynski, K.; Cheng, W.; Hüllermeier, E. Bayes optimal multilabel classification via probabilistic classifier chains. In Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; Volume 10, pp. 279–286.
104. Nam, J.; Mencia, E.L.; Kim, H.J.; Fürnkranz, J. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In Proceedings of the Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5413–5423.
105. Liu, N.; Wang, Q.; Ren, J. Label-Embedding Bi-directional Attentive Model for Multi-label Text Classification. *Neural Process. Lett.* **2021**, *53*, 375–389. [[CrossRef](#)]
106. Zhang, M.; Zhou, Z. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognit.* **2007**, *40*, 2038–2048. [[CrossRef](#)]
107. Lanchantin, J.; Sekhon, A.; Qi, Y. Neural message passing for multi-label classification. In Proceedings of the ECML-PKDD, Ghent, Belgium, 14–18 September 2020; Volume 11907, pp. 138–163.
108. Bai, J.; Kong, S.; Gomes, C. Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model. In Proceedings of the IJCAI International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 1–2 January 2020.
109. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
110. Swarnalatha, K.; Guru, D.S.; Anami, B.S.; Vinay, Kumar, N. A filter based feature selection for imbalanced text classification. In Proceedings of the International Conference on Recent Trends in Image Processing & Pattern Recognition (RTIP2R), Solapur, India, 21–22 December 2018.
111. Huang, Y.; Gilederehi, B.; Köksal, A.; Özgür, A.; Ozkirimli, E. Balancing Methods for Multi-label Text Classification with Long-Tailed Class Distribution. *arXiv* **2021**, arXiv:2109.04712.

112. Pal, A.; Selvakumar, M.; Sankarasubbu, M. Multi-label text classification using attention-based graph neural network. *arXiv* **2020**, arXiv:2003.11644.
113. Lu, H.; Ehwerhemuepha, L.; Rakovski, C. A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC Med. Res. Methodol.* **2022**, *22*, 181. [[CrossRef](#)]
114. Kim, H.; Jeong, Y.S. Sentiment classification using convolutional neural networks. *Appl Sci.* **2019**, *9*, 2347. [[CrossRef](#)]
115. Hughes, M.; Li, I.; Kotoulas, S.; Suzumura, T. Medical text classification using convolutional neural networks. In *Informatics for Health: Connected Citizen-Led Wellness and Population Health*; IOS Press: Amsterdam, The Netherlands, 2017; pp. 246–250.
116. Widiastuti, N.I. Convolution neural network for text mining and natural language processing. *Iop Conf. Ser. Mater. Sci. Eng.* **2019**, *662*, 052010. [[CrossRef](#)]
117. Banerjee, I.; Ling, Y.; Chen, M.C.; Hasan, S.A.; Langlotz, C.P.; Moradzadeh, N.; Chapman, B.; Amrhein, T.; Mong, D.; Rubin, D.L.; et al. Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. *Artif. Intell. Med.* **2019**, *97*, 79–88. [[CrossRef](#)] [[PubMed](#)]
118. Liu, Z.; Huang, H.; Lu, C.; Lyu, S. Multichannel cnn with attention for text classification. *arXiv* **2020**, arXiv:200616174.
119. Zhao, W.; Joshi, T.; Nair, V.N.; Sudjianto, A. Shap values for explaining cnn-based text classification models. *arXiv* **2020**, arXiv:200811825.
120. Cheng, H.; Yang, X.; Li, Z.; Xiao, Y.; Lin, Y. Interpretable text classification using CNN and max-pooling. *arXiv* **2019**, arXiv:1910.11236.
121. Askari, Q.; Younas, I.; Saeed, M. Political optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowl.-Based Syst.* **2020**, *195*, 105709. [[CrossRef](#)]
122. Askari, Q.; Saeed, M.; Younas, I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst. Appl.* **2020**, *161*, 113702. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.