Abstract: In recent times, global cities have been transforming from traditional cities to sustainable smart cities. In text sentiment analysis (SA), many people face critical issues namely urban traffic management, urban living quality, urban information security, urban energy usage, urban safety, etc. Artificial intelligence (AI)-based applications play important roles in dealing with these crucial challenges in text SA. In such scenarios, the classification of COVID-19-related tweets for text SA includes using natural language processing (NLP) and machine learning methodologies to classify tweet datasets based on their content. This assists in disseminating relevant information, understanding public sentiment, and promoting sustainable practices in urban areas during this pandemic. This article introduces a modified aquila optimizer with a stacked deep learning-based COVID-19 tweet Classification (MAOSDL-TC) technique for text SA. The presented MAOSDL-TC technique incorporates FastText, an effective and powerful text representation approach used for the generation of word embeddings. Furthermore, the MAOSDL-TC technique utilizes an attention-based stacked bidirectional long short-term memory (ASBiLSTM) model for the classification of sentiments that exist in tweets. To improve the detection results of the ASBiLSTM model, the MAO algorithm is applied for the hyperparameter tuning process. The presented MAOSDL-TC technique is validated on the benchmark tweets dataset. The experimental outcomes implied the promising results of the MAOSDL-TC technique compared to recent models in terms of different measures. This MAOSDL-TC technique improves accuracy and interpretability of sentiment prediction.

Keywords: artificial intelligence; COVID-19; emotions; Twitter data; sentiment analysis

1. Introduction

Social media platforms play an important part during extreme crises as individuals use these communications media to share feedback, sentiments, thoughts, and reactions with other people to manage and respond to crises [1]. Thus, this study focuses on explorative collective reactions to events expressed on social platforms [2]. Special consideration will be given to analyzing the public’s responses to worldwide medical-relevant events, particularly the pandemic, described through Twitter’s social network, due to its widespread reputation and ease of access utilizing the application programming interface (API) [3]. Sentiment analysis (SA) is a kind of technique employed to represent, separate, or define personal data like ideas communicated in a given content, depending on natural language processing (NCP) and computational methods [4]. The major goal of SA is to define the
author’s feelings as negative, positive, or neutral regarding different subjects [5]. To evaluate the effects of social media information relevant to the COVID-19 pandemic, research associated with people’s opinions on medical information and applications gained major significance [6]. In particular, text analysis of Twitter information has been the emphasis in several reviews, allowing researchers to analyze massive instances of user-defined content to find views, which can inform decision-making and earlier reaction mechanisms [7]. The Twitter platform has been undergoing a large infusion of data relevant to COVID-19 problems [8]. For SA, researchers have been using different kinds of textual documents such as Facebook posts and tweets [9].

Several research works on SA using social media data are available in the literature [10]. Identification of such sentiments from social media can support respondents in comprehending network dynamism, for example, panics, users’ important problems, and emotional impacts on members’ skills [11]. This study aims to examine the application of deep learning (DL) methods and natural language processing (NLP) approaches, namely SA, to support policymakers and communities to avoid the growth of misleading information, incitement of insurrection, and fake news [12]. SA or public view mining can be described as a way of employing machine learning (ML) and NLP for the classification of sentiments and subjective data [13]. SA is defined as the most common research field in the domain of NLP as it provides the ability to study and analyze sentiments that are expressed by various individuals [14].

This article introduces a modified aquila optimizer with stacked deep learning-based tweets classification (MAOSDL-TC) technique for text SA. The presented MAOSDL-TC technique incorporates FastText, an effective powerful text representation approach used for the generation of word embeddings. Furthermore, the MAOSDL-TC technique utilizes an attention-based stacked bidirectional long short-term memory (ASBiLSTM) model for the classification of sentiments that exist on Twitter. To improve the detection results of the ASBiLSTM model, the MAO algorithm is applied for the hyperparameter tuning process. The presented MAOSDL-TC technique is validated against the benchmark of a COVID-19 tweets dataset.

2. Related Works

Qorib et al. [15] downloaded public tweets day-to-day from Twitter using the Tweeter API and pre-processed and labelled them. Vocabulary normalization was based mainly on the stemming and lemmatization processes. The NRCLexicon method was used to transform tweets into 10 different classes. A T-test was deployed to check the statistical point of the relationship between the sentiments. Lastly, neural networks including the bidirectional encoder representations from transformers (BERT), 1-dimensional convolutional neural network (1DCNN), long short-term memory (LSTM), and multilayer perceptron (MLP) were tested and trained. In [16], an approach was introduced that was designed to provide an ensemble module where the advantages of automatic feature extraction and handcrafted features were linked through ML and DL algorithms. Before training ML techniques, unstructured information was attained, pre-processed, and annotated using VADER and TextBlob. Sunagar et al. [17] implemented the tweet classification of COVID-19 datasets via implementing DL approaches. The algorithm was executed using two-word embedding methods such as Global Vector for Word2Vec and Word Representation (GloVe).

In [18], the researchers presented an NLP technique based on the bidirectional LSTM (BiLSTM) method to implement sentiment classification and detect several problems related to public sentiment on COVID-19. BiLSTM is an enhanced version of classical LSTM to generate the outputs from right and left contexts at every time step. This enabled authorized institutions utilizing this model to alleviate the effect of negative messages and to understand the people’s concerns. Tatineni et al. [19] presented a technique to evaluate the emotion of live tweets. The technique comprised a dashboard with different functionalities. The central dashboard had a clickable map of India that illustrated state-wide data visualization and had country-wide data visualization of the emotion drawn from
Twitter. Live emotion prediction of tweets can be accomplished using the DL techniques. Tweet fetching is a dynamic to obtain new data automatically. Vaddadi et al. [20] developed a technique that used automated implementation to extract details regarding COVID-19 from the up-to-date tweet data. SA uses LSTM, which is a kind of recurrent neural network (RNN) employed by using Twitter’s COVID-19 hashtags to see people’s reactions to the outbreak. Then, the tweet datasets are categorized and labelled as positive, negative, and neutral and the results visualized.

Chakraborty et al. [21] presented SA on the amount of tweets gathered on COVID-19. In the beginning, they analyzed the trends of public sentiments related to COVID-19 using the n-gram analysis and evolutionary classification. Next, the sentiment rating was calculated on gathered tweets based on the class. Lastly, the LSTM model was trained through two classes of rated tweets to forecast sentiment on the COVID-19 dataset. Tawfik and Makhlouf [22] analyzed public opinions on the program of vaccination against COVID-19. To achieve this, an ensemble mechanism based on DL was established, which fused LSTM and bidirectional gated recurrent unit (BiGRU). The accuracy of the presented algorithm was compared with five different ML techniques, and two DL algorithms using advanced approaches.

Raheja and Asthana [23] implemented an SA of tweets in lockdown utilizing a multinomial LR approach. The presented methodology design followed the pre-processed, polarity and scoring, and extracting features before executing the ML approach. In [24], a novel algorithm was presented for automatic sentiment classification of COVID-19 tweets utilizing ANFIS approaches. Jain et al. [25] purposed to analyze the performance of many classification techniques that demand an input value and identified to which resultant classification they belong. Six ML approaches, two ensemble systems, and four DL methods were utilized for this work. In [26], the R programming language was used to conduct an investigation of Twitter data. During this case, the authors planned a method named Hybrid Heterogeneous SVM (H-SVM) and carried out the sentiment classification and categorized tweets as negative, neutral, and positive.

3. The Proposed Model

This article is concentrated on the improvement of the MAOSDL-TC technique for text SA. The MAOSDL-TC technique mainly concentrates on the recognition and categorization of different kinds of sentiments in COVID-19 tweets. In the presented MAOSDL-TC technique, the following set of processes are involved, namely pre-processing, FastText, ASBiLSTM-based classification, and MAO-based parameter selection. Figure 1 depicts the workflow of the MAOSDL-TC algorithm.

3.1. Data Pre-Processing and Word Embedding

Text preprocessing is the technique used to clean the original text data. A robust text preprocessing technique is crucial for applications of NLP tasks. After preprocessing, the attained text components act as key elements of input that are fed into the processing of textual data. Preprocessing consists of different approaches for translating the original texts using a well-defined method: special characters or symbols, lemmatization, elimination of stopwords, lexical analysis (ignore case sensitivity, word tokenization, and removal of punctuation). Afterwards, the FastText method was employed for the processing of word embedding. FastText is a widely used text representation method that generates word embeddings that are a dense vector representation of words. This embedding captures the semantic meaning of an individual word and its subword information and morphological structure. Particularly, this makes FastText more effective in handling out-of-vocabulary words and capturing the relationship between words with related prefixes or suffixes. FastText works by assuming a word is a mixture of subword units (character n-grams). This technique enables it to create embedding for known and unknown words by leveraging the subword component.
Figure 1. Workflow of MAOSDL-TC algorithm.

3.2. Tweet Data Classification Using ASBiLSTM Model

Once the tweets are preprocessed, classification takes place using the ASBiLSTM model. In this study, we used the ASBiLSTM model as an essential element of the presented method, which has the benefit of simultaneously extracting temporal features of time series [27]. The BILSTM is an augmentation of the LSTM. The LSTM is a kind of RNN, which overcomes the problems of vanishing gradient from RNN through the inclusion of a gating module. In comparison with RNN, LSTM is composed of memory cells, forget, input, and outputs, in which the cell memory is liable to store the overview of historical input series, and the gate modules control the flow of information between the input and output datasets. LSTM aids efficient learning of long-term temporal dependency relationships by taking their well-developed structure into account.
Consider $c_{t-1}$ as the memory cell state of the prior $t-1$ time step, an input vector $x_t$ at $t$ time steps, and $h_{t-1}$ indicates the hidden layer of the prior $t-1$ time step. $f_t, i_t,$ and $0_t$ show the gate vector that controls how much data is to be forgotten, updated, and output from the memory cell, correspondingly. The operation of LSTM was formulated by the following expression:

$$
\begin{align*}
    f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
    i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
    o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
    c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
    h_t &= o_t \odot \tanh(C)
\end{align*}
$$

From the expression, the Tanh function ensures that the value of HL remains in $[-1, 1]$ the interval. $\sigma(\bullet)$ indicates the sigmoid function; the symbol $\odot$ shows the pointwise multiplication. The learnable parameters $W$ and $b$ are weight and deviation during the training model, respectively.

BILSTM combines a bidirectional conceptualization into LSTM that exploits forward and backward LSTM for feature extraction and concatenates respective hidden features for extracting patterns or bidirectional features. Accordingly, BILSTM attains context data in the previous observation for the entire input. This bidirectional extraction on the time series simplifies the capture of backwards and forward temporal attributes in wind power-related data considering the variation patterns. With the context feature, BiLSTM allows a hybrid model for wind power-related data to attain feature extraction capabilities and better representation that enables more accurate and efficient prediction of future observation by leveraging past observation.

Particularly, BILSTM trains its parameters in backward and forward paths to realize the context. During the backward layer, the LSTM estimates the derivation of transmission errors in the forward layer. The LSTM updates the parameters from the conventional way in the forward layer. Considering an input of length $T$, the operational procedures are shown below:

$$
\begin{align*}
    \vec{h}_t &= \overrightarrow{\text{LSTM}}(h_{t-1}, x_t, c_{t-1}), t \in [1, T] \\
    \vec{h}_t &= \overleftarrow{\text{LSTM}}(h_{t+1}, x_t, c_{t+1}), t \in [T, 1] \\
    H_t &= [\overrightarrow{h}_t, \overleftarrow{h}_t]
\end{align*}
$$

where $H_t$ indicates the hidden layer (HL) of BILSTM at time step $t$, $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ signifies the HL in the forward and backward layers at timestep.

In ASBiLSTM, the attention module was used to optimize the prediction outcomes. Figure 2 signifies the framework of ASBiLSTM. The attention module is a weighting quantity of sequences that allocates great weight to targets with higher correlation. An attention module minimizes the loss of prior datasets and extracts relevant information by highlighting the contribution of the most powerful and useful parts of the input to the outputs. In the DL technique, the attention module allocates weight to the output of BiLSTM by mapping the weights and the learning parameter matrix can be focused on the input that contributes to the outputs.
Electronics 2023, 12, x FOR PEER REVIEW 6 of 16

In ASBiLSTM, the attention module was used to optimize the prediction outcomes. As shown in Equations (3) to (5), a series of outputs $H_1, H_2, \ldots, H_t$ through the HL of BILSTM are fed as input to the attention model, and the distribution of attention weights is attained. Equation (5) indicates the accomplishment of the last state of the attention mechanism. Equation (4) shows the computation of attention weight by standardizing the score. Equation (3) defines the computation of similarities or correlations between the input and output features.

$$e_i = V_t \tanh(WH_i + b) \quad (3)$$

$$a_i = \frac{\exp(e_i)}{\sum_i \exp(e_i)} \quad (4)$$

$$C = \sum_j a_i H_j \quad (5)$$

where $V_t$ and $W$ signify the weighted coefficient of the parameter learned in the training model, $e_i$ indicates the distribution probability at $i$th time steps. $b$ shows the bias.

3.3. Hyperparameter Tuning Using MAO Algorithm

The MAO algorithm can be applied in this work for the hyperparameter tuning of the ASBiLSTM module. The AO mainly depends upon the prey-grabbing nature of the Aquila. AO is a population-based algorithm which exhibits its effectiveness in the field of complex and nonlinear optimization in a short period of time. The classical AO principally focuses on five significant steps namely initialization, expanded exploration, narrowed exploration, expanded exploitation and narrowed exploitation.

An MAO was introduced in this study [28]. By modifying the SCF from IAO, MAO was inspired to make further amendments to the AO. However, the convergence properties of SCF decelerate the accuracy of the epochs in IAO. These properties may be responsible for certain challenges in searching for an optimum result. To overcome these challenges, a modified version of IAO was introduced that integrates a modified search control factor (MSCF) that is particularly adapted to the 2nd and 3rd search processes. The subsequent section provides a detailed description of the MAO technique, which highlights certain modifications that were made and their effects on the optimization technique. The MSCF is used to control the search range, which reduces movement of the Aquila in terms of epochs. Accordingly, compared to the prior SCF, the search space is considerably narrower.
Furthermore, the optimum solution is found considerably more quickly than in the prior technique. The modified MSCF is shown as follows:

\[ M - SCF(t) = 2 \times \exp \left( 1 - \left( \frac{t \times (t \times 0.1)}{T} \right) \right) \times \text{dir.} \]  

(6)

\[ \text{dir} = \begin{cases} 
1 & \text{if } r < 0.5, \\
-1 & \text{else}.
\end{cases} \]  

(7)

where \( t \) denotes the existing iteration and \( T \) shows the maximal iteration. The \( r \) parameter shows a random integer ranging from zero to one, where \( \text{dir} \) indicates the direction control factor. These factors play a major role in controlling the flight direction of the Aquila.

The MSCF function aims to attain fast convergence by restricting the movement of the Aquila. Furthermore, it decreases optimization latency. The modified technique needs less time to recognize the optimum solution set than the original AO. Both optimization approaches were performed with sizes of 250 and 250 epochs.

With the incorporation of the MSCF function, the presented technique includes four different search stages that are discussed in the following:

**Step 1: Vertical Dive Attack (S_1)**

The Aquila begins its hunting by identifying the target region and selecting the optimum hunting position by swooping high in the air. These attacks are called vertical dive attacks and are expressed as follows:

\[ S_1(t + 1) = S_{\text{best}}(t) \times \left( 1 - \frac{t}{T} \right) + (m(t) - S_{\text{best}}(t)) \times r \]  

(8)

In Equation (8), \( S_1(t + 1) \) denotes the solution candidate of \((t + 1)\) epochs, \( r \) shows the random integer in \([0, 1]\) interval, and \( S_{\text{best}}(t) \) shows the better solution attained to the \( t \)th generation. \((1 - Tt)\) is used for controlling the search region. Now, \( m(t) \) denotes the mean value of the existing solution to \( t \) epochs.

**Step 2: Modified Full Search with a Short Glide Attack (MS)**

Before attacking the prey, the Aquila comprehensively searches the solution space via different directions and speeds, in what is called a full search with shorter glide attacks that can be shown as follows:

\[ MS_2(t + 1) = S_R(t) + M - SCF(t) \times (S_{\text{best}}(t) - S(t)) \times r \times (y - x) \]  

(9)

In Equation (9), \( x \), and \( y \) correspond to the positions or coordinates of the point making the spiral shape during the search step, \( r \) indicates the random integer within \([0,1]\), and \( MCF(t) \) denotes the modified search control factor. Rather than applying the Levy flight (LF) distribution, we integrated MSCF to eliminate the problems of getting stuck in a locally optimal solution.

**Step 3: Modified Search Around Prey and Attack (MS)**

The prey's region is located accurately after the \( MS_2 \) search step. The Aquila thoroughly explores around the target, and with pseudo attacks, recognizes the prey’s reaction in what is called a search around prey and attack.

\[ MS_3(i, j) = lb_j + r \times (ub_j - lb_j) + r \times (S_R(j) - S_{\text{best}}(j)) \times M - SCF(t) \times \left( 1 - \frac{t}{T} \right) \]  

(10)

In Equation (10), \( S_R(j) \) denotes the random set of solutions and \( MS_3(i, j) \) indicates the existing solution for \( t \) epochs.

**Step 4: Walk and Grab Attack (S)**
Finally, the Aquila attacks from above based on the prey’s movement for the 4th search approach. This search process can be denoted as “Walk and Grab Prey”,

\[ S_4(t + 1) = Q_F \times S_{\text{best}}(t) - (G_1 \times S(t) \times r) - G_2 \times \text{lev}(D), \]  

(11)

\[ Q_F = \frac{2^{x-1}}{t^{(n-1)^2}}, \]  

(12)

\[ G_1 = 2 \times \text{random} - 1, \]  

(13)

\[ G_2 = 2 \times \left(1 - \frac{t}{T}\right). \]  

(14)

where \( S_4(t + 1) \) represents the solution attained so far, and \( \text{lev}(D) \) shows the Levy distribution for the \( D \) dimensional range. \( Q_F \) indicates the quality function for balancing the search process, \( G_1 \) denotes each kind of movement of Aquila during the hunt, and \( G_2 \) shows the fight slope of hunting.

The fitness choice is a key component of the MAO method. Encoder performance is applied measure a better solution candidate. Now, the performance value is the foremost condition applied to develop an FF.

\[ \text{Fitness} = \max(P) \]  

(15)

\[ P = \frac{TP}{TP + FP} \]  

(16)

where \( TP \) and \( FP \) indicate the true and false positive values.

4. Results and Discussion

The performance validation of the MAOSDL-TC method on the sentiment classification of COVID-19 tweets takes place using the Kaggle dataset [29], which holds 2750 samples with 11 classes, as portrayed in Table 1.

Table 1. Description of database.

<table>
<thead>
<tr>
<th>Class</th>
<th>No. of Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimistic</td>
<td>250</td>
</tr>
<tr>
<td>Thankful</td>
<td>250</td>
</tr>
<tr>
<td>Empathetic</td>
<td>250</td>
</tr>
<tr>
<td>Pessimistic</td>
<td>250</td>
</tr>
<tr>
<td>Anxious</td>
<td>250</td>
</tr>
<tr>
<td>Sad</td>
<td>250</td>
</tr>
<tr>
<td>Annoyed</td>
<td>250</td>
</tr>
<tr>
<td>Denial</td>
<td>250</td>
</tr>
<tr>
<td>Surprise</td>
<td>250</td>
</tr>
<tr>
<td>Official report</td>
<td>250</td>
</tr>
<tr>
<td>Joking</td>
<td>250</td>
</tr>
<tr>
<td><strong>Total Number of Instances</strong></td>
<td><strong>2750</strong></td>
</tr>
</tbody>
</table>

Figure 3 represents the classifier performances of the MAOSDL-TC technique under the test database. Figure 3a,b shows the confusion matrix achieved by the MAOSDL-TC.
approach at 70:30 of the TR set/TS set. The outcome value determined that the MAOSDL-TC algorithm has classified and detected all 11 classes accurately. Then, Figure 3c depicts the PR examination of the MAOSDL-TC approach. The outcome demonstrated that the MAOSDL-TC algorithm has achieved greater PR outcomes in 11 classes. But, Figure 3d displays the ROC outcome of the MAOSDL-TC algorithm. The simulation value exhibited that the MAOSDL-TC approach led to able performances with higher values of ROC on 11 classes.

Figure 3. Performance of (a,b) confusion matrices, (c) PR_curve, and (d) ROC.

4.1. Result Analysis

A brief result of using the MAOSDL-TC technique on COVID-19 tweet classification is illustrated in Table 2 and Figure 4. The obtained results state that the MAOSDL-TC technique properly recognized all classes. On 70% of the TR set, the MAOSDL-TC technique provided an average accy of 99.19%, precn of 95.63%, recal of 95.55%, Fscore of 95.54%, and JI of 91.49%. In addition, on 30% of the TS set, the MAOSDL-TC approach attained an average accy of 99.45%, precn of 97.15%, recal of 96.89%, Fscore of 96.99%, and JI of 94.18%.
Table 2. COVID-19 tweet classification outcome of the MAOSDL-TC method on 70:30 of TR set/TS set.

<table>
<thead>
<tr>
<th>Class</th>
<th>Acc$_y$</th>
<th>Prec$_n$</th>
<th>Reca$_l$</th>
<th>F$_{score}$</th>
<th>Jaccard Index</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TR set (70%)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimistic</td>
<td>99.17</td>
<td>96.51</td>
<td>94.32</td>
<td>95.40</td>
<td>91.21</td>
</tr>
<tr>
<td>Thankful</td>
<td>98.60</td>
<td>92.47</td>
<td>92.97</td>
<td>92.72</td>
<td>86.43</td>
</tr>
<tr>
<td>Empathetic</td>
<td>99.32</td>
<td>94.32</td>
<td>98.22</td>
<td>96.23</td>
<td>92.74</td>
</tr>
<tr>
<td>Pessimistic</td>
<td>99.22</td>
<td>97.11</td>
<td>94.38</td>
<td>95.73</td>
<td>91.80</td>
</tr>
<tr>
<td>Anxious</td>
<td>99.12</td>
<td>92.67</td>
<td>98.33</td>
<td>95.42</td>
<td>91.24</td>
</tr>
<tr>
<td>Sad</td>
<td>99.22</td>
<td>94.67</td>
<td>96.39</td>
<td>95.52</td>
<td>91.43</td>
</tr>
<tr>
<td>Annoyed</td>
<td>99.69</td>
<td>100.00</td>
<td>96.84</td>
<td>98.40</td>
<td>96.84</td>
</tr>
<tr>
<td>Denial</td>
<td>98.96</td>
<td>92.35</td>
<td>96.57</td>
<td>94.41</td>
<td>89.42</td>
</tr>
<tr>
<td>Surprise</td>
<td>99.43</td>
<td>100.00</td>
<td>93.45</td>
<td>96.62</td>
<td>93.45</td>
</tr>
<tr>
<td>Official report</td>
<td>99.12</td>
<td>92.44</td>
<td>97.55</td>
<td>94.93</td>
<td>90.34</td>
</tr>
<tr>
<td>Joking</td>
<td>99.22</td>
<td>99.38</td>
<td>92.00</td>
<td>95.55</td>
<td>91.48</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>99.19</strong></td>
<td><strong>95.63</strong></td>
<td><strong>95.55</strong></td>
<td><strong>95.54</strong></td>
<td><strong>91.49</strong></td>
</tr>
<tr>
<td><strong>TS set (30%)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimistic</td>
<td>99.52</td>
<td>97.30</td>
<td>97.30</td>
<td>97.30</td>
<td>94.74</td>
</tr>
<tr>
<td>Thankful</td>
<td>99.39</td>
<td>98.39</td>
<td>93.85</td>
<td>96.06</td>
<td>92.42</td>
</tr>
<tr>
<td>Empathetic</td>
<td>99.03</td>
<td>92.94</td>
<td>97.53</td>
<td>95.18</td>
<td>90.80</td>
</tr>
<tr>
<td>Pessimistic</td>
<td>99.27</td>
<td>98.53</td>
<td>93.06</td>
<td>95.71</td>
<td>91.78</td>
</tr>
<tr>
<td>Anxious</td>
<td>99.76</td>
<td>100.00</td>
<td>97.14</td>
<td>98.55</td>
<td>97.14</td>
</tr>
<tr>
<td>Sad</td>
<td>99.39</td>
<td>96.47</td>
<td>97.62</td>
<td>97.04</td>
<td>94.25</td>
</tr>
<tr>
<td>Annoyed</td>
<td>99.64</td>
<td>96.72</td>
<td>98.33</td>
<td>97.52</td>
<td>95.16</td>
</tr>
<tr>
<td>Denial</td>
<td>99.39</td>
<td>96.05</td>
<td>97.33</td>
<td>96.69</td>
<td>93.59</td>
</tr>
<tr>
<td>Surprise</td>
<td>99.76</td>
<td>98.78</td>
<td>98.78</td>
<td>98.78</td>
<td>97.59</td>
</tr>
<tr>
<td>Official report</td>
<td>99.15</td>
<td>93.48</td>
<td>98.85</td>
<td>96.09</td>
<td>92.47</td>
</tr>
<tr>
<td>Joking</td>
<td>99.64</td>
<td>100.00</td>
<td>96.00</td>
<td>97.96</td>
<td>96.00</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>99.45</strong></td>
<td><strong>97.15</strong></td>
<td><strong>96.89</strong></td>
<td><strong>96.99</strong></td>
<td><strong>94.18</strong></td>
</tr>
</tbody>
</table>

Figure 5 illustrates the training accuracy $TR\_accu_y$ and $VL\_accu_y$ of the MAOSDL-TC approach. The $TL\_accu_y$ is determined by the evaluation of the MAOSDL-TC technique on the TR dataset whereas the $VL\_accu_y$ is computed by evaluating performance on a separate testing dataset. The results exhibit that $TR\_accu_y$ and $VL\_accu_y$ upsurge with an increase in epochs. As a result, the outcome of the MAOSDL-TC technique increases on the TR and TS datasets with a rise in the number of epochs.
Figure 4. Average of MAOSDL-TC approach on 70:30 of the TR set/TS set.

Figure 5. Accuracy curve of the MAOSDL-TC approach.

In Figure 6, the $TR_{loss}$ and $VR_{loss}$ results of the MAOSDL-TC approach without optimization are revealed. The $TR_{loss}$ defines errors among the predictive performance
and original values on the TR data. The $VR_{loss}$ represents a measure of the performance of the MAOSDL-TC system on individual validation data. The outcome signifies that the $TR_{loss}$ and $VR_{loss}$ tend to decrease with rising epochs. It depicted the enhanced performance of the MAOSDL-TC method and its capability to make accurate classifications. The minimal value of $TR_{loss}$ and $VR_{loss}$ demonstrates the improved performance of the MAOSDL-TC technique in capturing patterns and relationships.

Figure 6. Loss curve of the MAOSDL-TC approach.

4.2. Discussion

In Table 3 and Figure 7, we offered an extensive comparative result of the MAOSDL-TC technique [30,31]. The results indicate that the MAOSDL-TC technique exhibits promising results over other models. Based on $accy$, the MAOSDL-TC technique reports an increasing $accy$ of 99.45% while the RF, XGBoost, SVM, ensemble, DT, SFODLDSAC, and MPONLP-TSA techniques accomplish decreasing values of 91.20%, 91.40%, 90.95%, 93.19%, 90.82%, 98.50%, and 99.10%, respectively. At the same time, based on $precn$, the MAOSDL-TC approach reports a higher $precn$ of 97.15% while the RF, XGBoost, SVM, ensemble, DT, SFODLDSAC, and MPONLP-TSA systems achieve limited values of 92.48%, 91.69%, 90.39%, 94.32%, 90.82%, 96.15%, and 96.74% correspondingly. Finally, based on $Fscore$, the MAOSDL-TC technique reports a maximal $Fscore$ of 96.99% while the RF, XGBoost, SVM, ensemble, DT, SFODLDSAC, and MPONLP-TSA algorithms realize minimal values of 91.49%, 91.20%, 90.25%, 93.96%, 90.46%, 95.15%, and 95.90% correspondingly.
Table 3. Comparative outcome of MAOSDL-TC algorithm with recent methodologies.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accu_y</th>
<th>Prec_y</th>
<th>Reca_l</th>
<th>F_Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>91.20</td>
<td>92.48</td>
<td>91.26</td>
<td>91.49</td>
</tr>
<tr>
<td>XGboost</td>
<td>91.40</td>
<td>91.69</td>
<td>91.82</td>
<td>91.20</td>
</tr>
<tr>
<td>SVM</td>
<td>90.95</td>
<td>90.39</td>
<td>90.32</td>
<td>90.25</td>
</tr>
<tr>
<td>Ensemble</td>
<td>93.19</td>
<td>94.32</td>
<td>93.15</td>
<td>93.96</td>
</tr>
<tr>
<td>DT</td>
<td>90.82</td>
<td>90.82</td>
<td>90.92</td>
<td>90.46</td>
</tr>
<tr>
<td>SFODLDSAC</td>
<td>98.50</td>
<td>96.15</td>
<td>95.13</td>
<td>95.15</td>
</tr>
<tr>
<td>MPONLP-TSA</td>
<td>99.10</td>
<td>96.74</td>
<td>95.88</td>
<td>95.90</td>
</tr>
<tr>
<td>MAOSDL-TC</td>
<td>99.45</td>
<td>97.15</td>
<td>96.89</td>
<td>96.99</td>
</tr>
</tbody>
</table>

Figure 7. Comparative outcome of MAOSDL-TC algorithm with recent methods.

These results confirmed that the MAOSDL-TC technique exhibits enhanced performance over recent models.

5. Conclusions

This article has concentrated on the improvement of the MAOSDL-TC method for classification of text sentiments in COVID-19 tweets. The MAOSDL-TC technique mainly concentrates on the recognition and categorization of different kinds of sentiments in COVID-19-related tweets. In the presented MAOSDL-TC technique, the following set of processes were involved, namely pre-processing, FastText, ASBiLSTM-based classification, and MAO-based parameter selection. In this work, the ASBiLSTM model for the classification of sentiments existing in the tweets. Lastly, the MAO system can be applied for the hyperparameter tuning process, which aids in improving the detection results of the ASBiLSTM model. The presented MAOSDL-TC method is validated on the bench-
mark tweets dataset. The experimental outcomes, with maximum accuracy of 99.45%, suggested the promising results of the MAOSDL-TC technique compared to recent models. This MAOSDL-TC technique not only improves accuracy but also enhances the better interpretability of sentiment prediction.


Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number (RGP2/185/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R281), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSPD2023R521), King Saud University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444). This study is partially funded by the Future University in Egypt (FUE).

Data Availability Statement: Data sharing does not apply to this article as no datasets were generated during the current study.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

8. Sak, S.; Yavuzyigit, B.B. Striving for wellbeing digitally in the city amidst the pandemic: Solidarity through Twitter in Ankara. Habitat Int. 2023, 137, 102846. [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.