mathematics

MDPI

Article

Differentiating Chat Generative Pretrained Transformer from Humans: Detecting ChatGPT-Generated Text and Human Text Using Machine Learning

Iyad Katib [1,*], Fatmah Y. Assiri [2], Hesham A. Abdushkour [3], Diaa Hamed [4] and Mahmoud Ragab [5,6,7]

[1] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[2] Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia; fyassiri@uj.edu.sa
[3] Nautical Science Department, Faculty of Maritime Studies, King Abdulaziz University, Jeddah 21589, Saudi Arabia; habdushakour@kau.edu.sa
[4] Faculty of Earth Sciences, King Abdulaziz University, Jeddah 21589, Saudi Arabia; dzeinalabedein@kau.edu.sa
[5] Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; mragab@kau.edu.sa
[6] Department of Mathematics, Faculty of Science, Al-Azhar University, Naser City, Cairo 11884, Egypt
[7] Centre for Artificial Intelligence in Precision Medicines, King Abdulaziz University, Jeddah 21589, Saudi Arabia
* Correspondence: iakatib@kau.edu.sa

Abstract: Recently, the identification of human text and ChatGPT-generated text has become a hot research topic. The current study presents a Tunicate Swarm Algorithm with Long Short-Term Memory Recurrent Neural Network (TSA-LSTMRNN) model to detect both human as well as ChatGPT-generated text. The purpose of the proposed TSA-LSTMRNN method is to investigate the model's decision and detect the presence of any particular pattern. In addition to this, the TSA-LSTMRNN technique focuses on designing Term Frequency–Inverse Document Frequency (TF-IDF), word embedding, and count vectorizers for the feature extraction process. For the detection and classification processes, the LSTMRNN model is used. Finally, the TSA is employed for selecting the parameters for the LSTMRNN approach, which enables improved detection performance. The simulation performance of the proposed TSA-LSTMRNN technique was investigated on benchmark databases, and the outcome demonstrated the advantage of the TSA-LSTMRNN system over other recent methods with a maximum accuracy of 93.17% and 93.83% on human- and ChatGPT-generated datasets, respectively.

Keywords: ChatGPT; artificial intelligence; feature extraction; human-generated text; tunicate swarm algorithm

MSC: 68-11

Citation: Katib, I.; Assiri, F.Y.; Abdushkour, H.A.; Hamed, D.; Ragab, M. Differentiating Chat Generative Pretrained Transformer from Humans: Detecting ChatGPT-Generated Text and Human Text Using Machine Learning. *Mathematics* **2023**, *11*, 3400. https://doi.org/10.3390/math11153400

Academic Editors: Faheim Sufi, Huawen Liu and Georgios Tsekouras

Received: 12 June 2023
Revised: 24 July 2023
Accepted: 31 July 2023
Published: 3 August 2023

1. Introduction

Over the decades, Natural Language Processing (NLP) has become an important field of research with an aim to improve the ability of computer systems to generate and understand human language [1]. The recent advancements in this field have led to the development of a large language model that exploits Machine Learning (ML) algorithms to generate humanlike language and learn from a humongous volume of textual data [2]. The Generative Pretrained Transformer (GPT) series, introduced by OpenAI, has received considerable attention in natural language processing [3]. The advancement of ChatGPT marks a crucial milestone in the field of NLP as it signifies a considerable step toward

the construction of sophisticated and state-of-the-art computer systems and it also has the potential to understand and generate natural languages [4]. Furthermore, it generates prompt responses that are not only contextually appropriate but also coherent in nature, and it is also trained using massive amounts of text data. The capacity of a model to generate text that is similar to human language has a major impact on communication, education, and language learning areas. The large language model that is currently popular, i.e., the ChatGPT system, represents considerable progress in the domain of NLP. BERT, coined by Google, is another significant example of a large language model [5]. Similar to ChatGPT, BERT can also be finetuned for NLP tasks that involve language translation, Sentimental Analysis (SA), and question–answer tasks and is pretrained using an enormous amount of text data. Though the systems vary in their model architecture and pretraining model, the core functionality of ChatGPT and BERT remains the same [6]. The advancement of such large language models helps in revolutionizing several industries that involve communication, education, and healthcare by allowing natural and more sophisticated interactions between machines and humans [7].

However, the highly prevalent adoption of revolutionary AI-based chatbots including ChatGPT highlights the importance of the capability to recognize whether a text has been written by human being or by AI [8]. This may have serious implications in different fields that involve digital forensics and information security [9]. For example, in the information security domain, the capability of identifying AI-generated text is essential both to detect and to protect against the malicious usage of AI, namely social engineering attacks or the spread of misinformation and disinformation. To ensure the accuracy and trustworthiness of the data [10], it is crucial to develop techniques to identify AI-generated texts. This is crucial because such techniques must be utilized in sensitive fields such as finance and banking, political campaigns, customer reviews, and legal documents (customer reviews of movies, restaurants, or products).

In this background, the current study presents the Tunicate Swarm Algorithm with Long Short-Term Memory Recurrent Neural Network (TSA-LSTMRNN) model to detect both human- and ChatGPT-generated text. The purpose of the proposed TSA-LSTMRNN method is to investigate the model's decision and identify the presence of any particular pattern. In addition to these, the TSA-LSTMRNN technique focuses on designing TF-IDF, word embedding, and count vectorizers for the purpose of feature extraction. For the detection and classification processes, the LSTMRNN model is used. Finally, the TSA is exploited for selecting the parameters for the LSTMRNN approach, which helps in attaining improved detection performance. The simulation performance of the proposed TSA-LSTMRNN technique was investigated using the benchmark datasets.

The rest of the paper is organized as follows. Section 2 provides information regarding related works, and Section 3 provides details about the proposed model. Then, Section 4 presents the analytical results, and Section 5 concludes the paper.

2. Related Works

Yu et al. [11] presented a large-scale CHatGPT-writtEn AbsTract dataset (CHEAT) to assist in the advancement of the recognition methods and inspect the possible negative effect of ChatGPT on academia. To be specific, the ChatGPT-written abstract data had a total of 35,304 synthetic abstracts, with Mix, Generation, and Polish as eminent representatives. Liao et al. [12] presented an ethical AIGC (Artificial Intelligence Generated Content) system in the healthcare sector. In this study, the authors mainly focused on examining the variances between medical texts generated by ChatGPT and those written by human experts. Further, the study also devised ML workflows to potentially distinguish and find medical texts generated by ChatGPT. At first, the authors built a set of datasets with one containing medical texts generated by ChatGPT and another one written by human experts. Eventually, the authors applied and devised ML approaches to ascertain the origin of the generated medical text.

Alamleh et al. [13] evaluated the performance of ML methods in distinguishing AI-generated text from human-written text. To achieve this objective, the authors gathered responses from computer science students to essay and programming assignments. Then, based on the data, the authors evaluated and trained numerous ML methods such as SVM, LR, NN, RF, and DT. Chen et al. [14] introduced an innovative method to differentiate human-written and ChatGPT-generated texts with the help of language methods. The authors gathered and released the preprocessed data called OpenGPTText, which contained rephrased content generated utilizing ChatGPT. Pardos and Bhandari [15] conducted an initial learning gain assessment of ChatGPT by comparing the efficiency of its hints to hints presented by human tutors on two different algebra topics such as intermediate and elementary algebra. Hamed and Yu [16] displayed how to differentiate ChatGPT-generated publications from their counterparts written by the researchers. By devising a supervised ML approach, the authors demonstrated how to differentiate machine-generated articles from scientist-created articles. The authors developed an algorithmic method that identified the ChatGPT-generated publication with a high precision.

Perkins et al. [17] explored the academic integrity considerations of students using AI tools, in addition to the Large Language Model (LLM), namely ChatGPT, in formal assessment. The authors evaluated the development of these tools and highlighted the possible ways in which the LLM supports the education of students in digital writing including composition and teaching of writing, the possibilities of co-creation between AI and the humans, enhancing Automated Writing Evaluation (AWE), and supporting EFL learners. Maddigan and Susnjak [18] presented a new system called Chat2VIS, which makes the maximum use of LLMs and illustrates how the complicated issue of language understanding can be resolved with high potential for prompt engineering, and it results in simple and precise end-wise solutions compared to the existing methods. Based on the prompts presented, Chat2VIS displays that the LLM presents a dependable method for rendering visualizations from natural language questions, even if the queries were underspecified and highly misspecified.

Only a limited number of studies are available in the literature related to the ChatGPT-generated text detection process. Therefore, there exists a need to improve the detection results for ChatGPT-generated texts. Due to a continuous deepening of the model, the number of parameters involved in DL models also increases quickly, which results in model overfitting. At the same time, different hyperparameters have significant impact on the efficiency of the CNN model. Particularly, hyperparameters such as epoch count, batch size, and learning rate selection are essential to attain effectual outcomes. Since the trial-and-error method for hyperparameter tuning is a tedious and an erroneous process, metaheuristic algorithms are applied. Therefore, in this work, the authors employed the TSA for parameter selection of the LSTMRNN model.

3. The Proposed Model

In the current research paper, the authors established an automated human-generated text and ChatGPT-generated text detection model, named the TSA-LSTMRNN model. The objective of the proposed TSA-LSTMRNN technique is to investigate the model's decision and compute whether any particular pattern can be detected. The model has three stages, namely feature extraction, LSTMRNN classification, and TSA-based parameter tuning. Figure 1 represents the overall flow of the TSA-LSTMRNN approach.

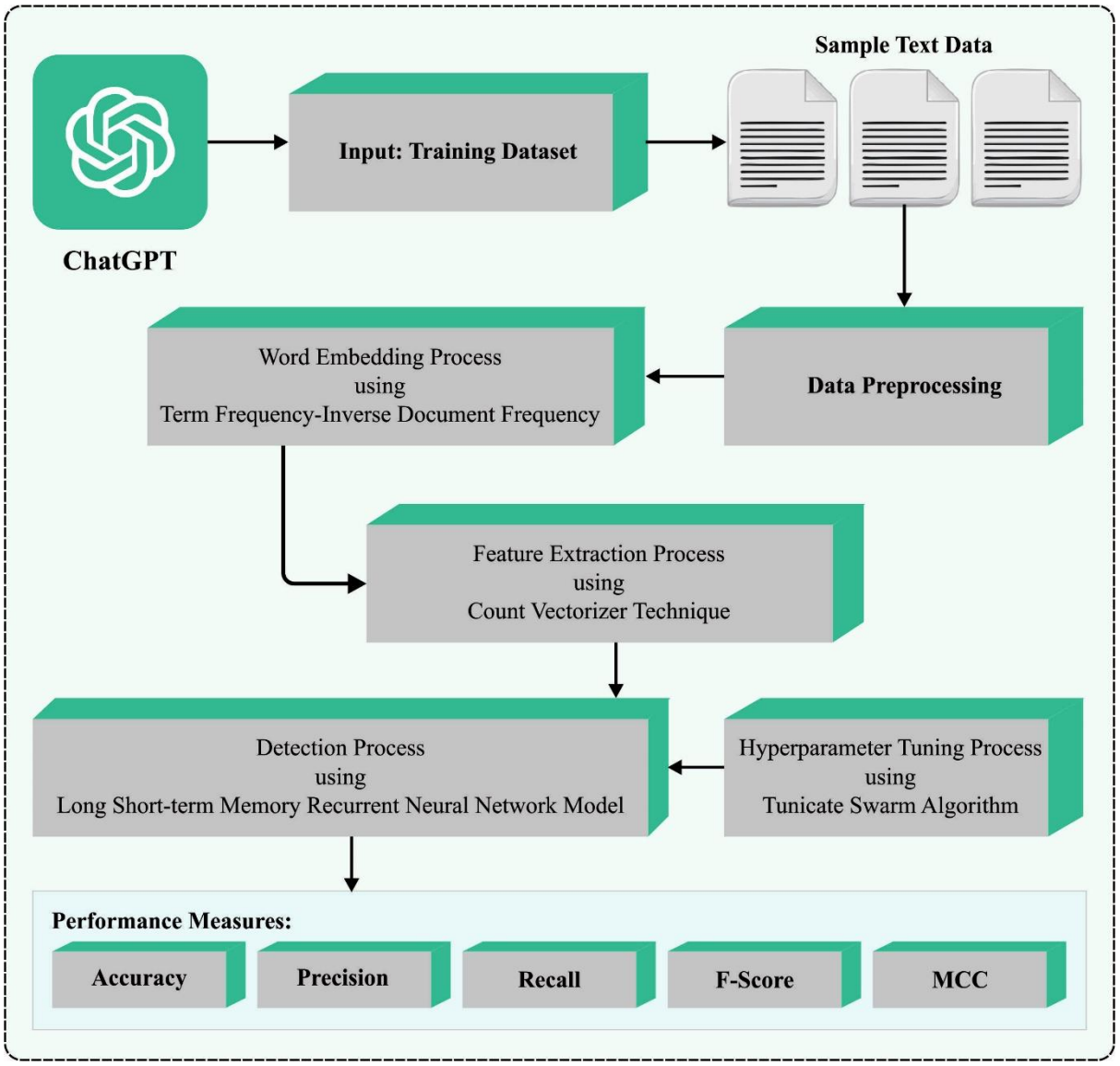

Figure 1. Overall flow of the TSA-LSTMRNN approach.

3.1. Feature Extraction

In this study, TF-IDF is used for the word embedding process, whereas the count vectorizer algorithm is utilized for feature extraction. In data mining, feature extraction is a procedure that contains steps to reduce the data count and make it accessible so as to define huge databases. When it comes to analyzing the mood of a difficult text, the main problem arises from the presence of numerous variables. Generally, to analyze a difficult or huge text, large amounts of memory and processing power are required. This results in the use of the classifier technique, which is more appropriate for trained instances and leads to worse generalizations for novel instances. The researchers mentioned that, in applications containing several features, extraction is the same process as dimensionality reduction. When feature extraction systems are applied to the input data before they are passed onto the classifier system, it is possible to achieve refined outcomes and high classification method accuracy.

In the literature, the count vectorizer feature extraction method has been utilized to cover tweets based on frequent words (count) that occurs in the tweets. Such tweets are then converted into a vector space [19]. A column matrix denotes the count vectorizer that generates a word matrix, whereas a row matrix corresponds to the text selected from the

document. Thereby, the word in that specific text instance is counted. The TF-IDF has a weighted feature for execution boosting and was used for tweet analysis in that study. Here, the length defines the TF of a feature in a single document.

$$TF = \frac{count_{t,d}}{totalcount_d} \tag{1}$$

Here, $count_{y,d}$ represents the amount of TF t in document d and $total\ count_d$ denotes the total amount of terms present in the document. IDF considers that when the text increases in terms of t, it would be highly informative for training the model.

$$idf' = \frac{i'}{df'_t} \tag{2}$$

In Equation (2), i' denotes the overall number of documents and df'_t shows the number of documents including the t phrase. Once the term t often appears in various documents, the IDF calculates the weight of the phrase t as low. For instance, a stop word has lower IDF values. Lastly, the TF-IDF is determined as follows:

$$tf - idf = tf_{t,d} * \log(idf). \tag{3}$$

Also, the current study used the word embedding method for feature extraction, namely Glove, pretrained model, FastText, and Word2Vec embedding with 300-D vectors.

3.2. Detection Model Using LSTMRNN

This stage aims to search a DL structure that contains attention layers so as to understand the detection process of human- and ChatGPT-generated texts. Therefore, in the presented LSTMRNN structure, many other processing stages are also involved. The primary one is the convolution of features [20]. The initial step is used for the extraction of high-level semantic features in word sequences. The LSTMRNN approach also determines the temporal connection among the features and creates the feature vectors.

Furthermore, the semantic meaning of the input text is assumed, and the secondary label sets are created with values allocated to individual data. The RNN approach proceeds with a series of pixels such as $x = x_1, x_2, \ldots\ldots, x_n$ creating Hidden Layers (HLs) $H = H_1,\ H_2, \ldots\ldots,\ H_n$ and outcome layers $O = O_1, O_2,\ \ldots\ldots, O_n$ in a subsequent manner.

$$O_t = \sigma(W_{H_tO_t} + b_t) \tag{4}$$

$$H_t = \sigma\left(W_{H_{t-1}H_t}H_{t-1} + W_{x_tH_t}x_t + b_{H_t}\right) \tag{5}$$

At this point, $W_{H_tO_t}$ signifies the vector in the hidden unit H_t and the output unit O_t, H_{t-1} stands for the hidden unit for the $t-1$ pixel series, $W_{H_{t-1}H_t}$ refers to the weighted vector in the hidden unit H_{t-1} to H_t for the sequence time t, and b_{H_t} and b_t refer to the biases.

Additionally, the LSTM stack is utilized to learn the time series features. On the other hand, the model acquires the issues contained in a single sequence of observation. The method must learn the sequences of past observations to predict the next value in order as given below.

$$ig_t = \tanh\left(W_{x_tig_t}x_t + W_{H_{t-1}ig_t}H_{t-1} + b_{ig_t}\right) \tag{6}$$

$$p_t = \sigma\left(W_{x_tp_t}x_t + W_{H_{t-1}p_t}H_{t-1} + b_{p_t}\right) \tag{7}$$

$$fg_t = \sigma\Big(W_{x_tfg_t}x_t + W_{H_{t-1}fg_t}H_{t-1} + b_{fg_t}\Big) \tag{8}$$

$$op_t = \sigma\left(W_{x_top_t}x_t + W_{H_{t-1}op_t}H_{t-1} + b_{op_t}\right) \tag{9}$$

$$Ce_t = ce_{t-1} \odot fg_t + ig_t \odot p_t \tag{10}$$

$$H_t = tanh(Ce_t) \odot op_t \tag{11}$$

Here, ig_i stands for the input gate; p_i implies the forecast from the initial layers; fg_t exemplifies the forget gate; H_t offers the data on output; b_{ig}, b_p, b_{fg}, b_{op} denote the bias vectors; Ce_t shows the cell state; and W_{xx} implies the weighted matrix. Either the RNN or the LSTM approach is integrated to extract the semantic features in the input tweets.

To be specific, it executes the attention layer so as to enhance the learning of the features as well as the feature weights. The LSTMRNN approach has been utilized to learn a series of sentences and create the weighted features through the attention procedure. Additionally, it also utilizes the secondary labels integrated with the LSTMRNN approach to assist in enhancing the areas of interest from the learning procedure. Therefore, X_i denotes the input, $f(X_i, X_{i+1})$ denotes the features created in the second layer, and $f(X_i, X_{i+1}, \cdots, X_i + L - 1)$ corresponds to the L^{th} layer. The feature value refers to the responses of multi-scale n-grams, for example, unigram X_i, bigram $X_i X_{i+1}$, and L-gram $X_i X_{i+1} \cdots X_i + L - 1$. Moreover, the scale reweighting is utilized to compute the SoftMax distribution of attention weights in which the descriptor is utilized as the dataset and the output weighted element weights are to be reweighed.

$$S_l^i = FL_{ensm}\left(X_l^i\right) \tag{12}$$

$$X_{atten}^i = \sum_{j=1}^{L} \alpha_L^i X_L^i \tag{13}$$

$$\partial_i^L = Softmax\left(MLP\left(X_{atten}^i\right)\right) \tag{14}$$

In the presented method, the novelty lies in the development of the weighted features by utilizing the attention layer procedure. The presented method recovers the text information in a series mapped by LSTMRNN, whereas the LSTM creates a series of annotations for every input. The vectors utilized are the concatenation of the HLs from the encoded layers. Afterward, the features can be developed using the attention layer model. At the time of the training process, every sample from the training data is passed on to the LSTMRNN approach one time step at a time. The LSTM units process the input, update their internal states, and produce an output at each time step. Then, the outputs are typically used to make the prediction process.

3.3. Hyperparameter Tuning Using TSA

In the current study, the TSA is used to improve the parameter selection process for the LSTMRNN model. The TSA is an optimization technique that is based on the biological behavior of animals, i.e., the foraging behavior of tunicates, marine invertebrates that radiate brighter bioluminescence [21]. In particular, the TSA draws its motivation from a peculiar behavior of the tunicates in the ocean, i.e., their jet drive and the swarm intelligence of the foraging processes. Under three major constraints, the mathematical modeling of jet propulsion is proposed: following the position of the best agent, preventing conflicts amongst the exploration agents, and remaining near the optimum agents. Figure 2 depicts the flowchart of the TSA.

To prevent conflicts when finding the best location, the new location of the search agent is evaluated as given below.

$$\vec{A} = \frac{\vec{G}}{\vec{M}} \tag{15}$$

$$\vec{G} = c_2 + c_3 - \vec{F} \tag{16}$$

$$\vec{F} = c_1 \cdot \vec{F}. \tag{17}$$

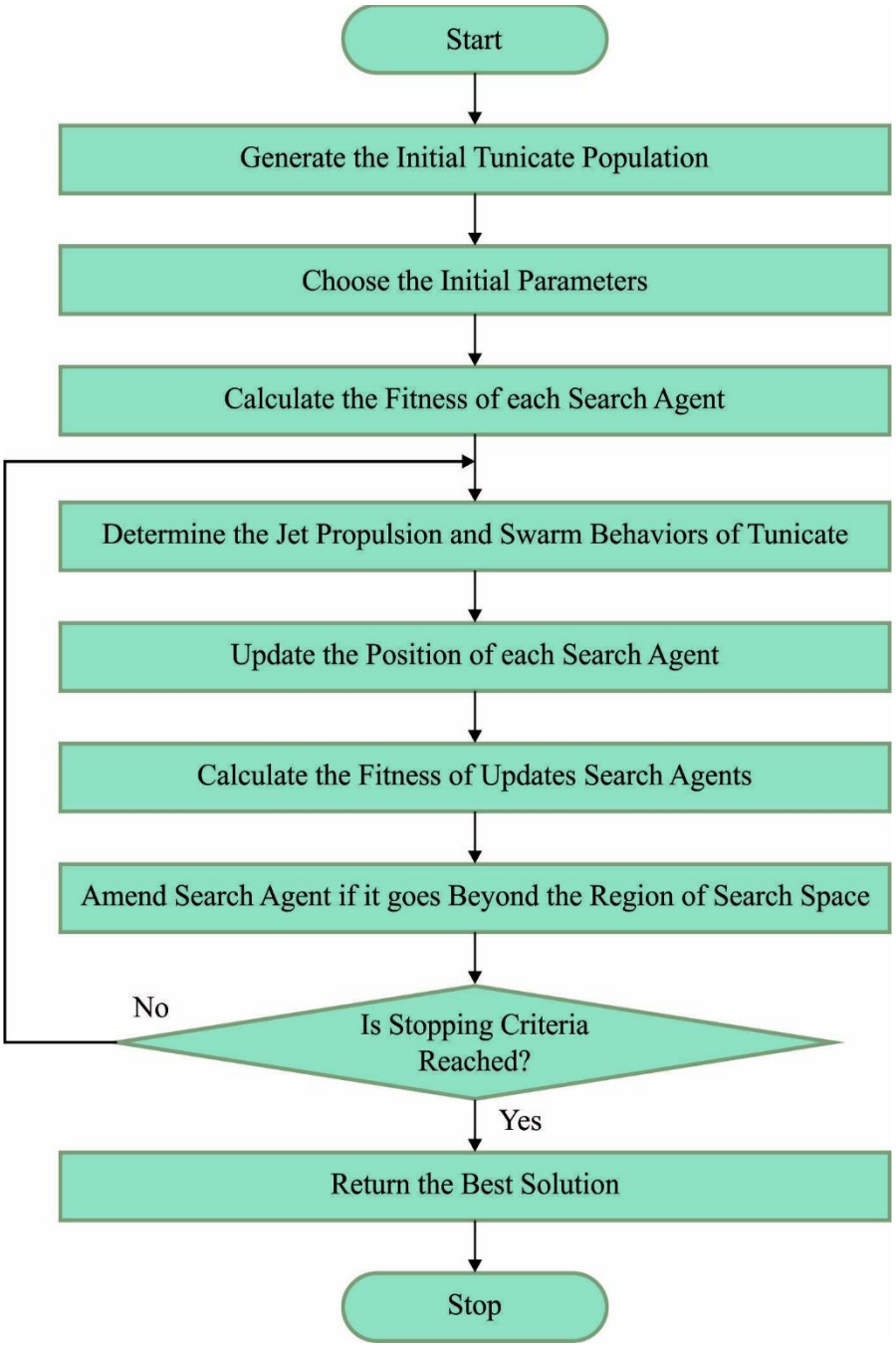

Figure 2. Flowchart of TSA.

Let $\vec{A}$ be the vector of a new position of the search agent; $\vec{F}$ refers to the water flow in the ocean; $\vec{G}$ indicates the gravity force; and $c1$, $c2$, and $c3$ denote three randomly generated numbers. The social forces between the agents are kept in a new vector $\vec{M}$ as given below.

$$\vec{M} = [P_{\min} + c_1 \cdot P_{\max} - P_{\min}]. \tag{18}$$

In Equation (18), $P_{\min} = 1$ and $P_{\max} = 4$ correspondingly describe the first and second subordinates, which demonstrate the speed of establishing a social interaction.

It is crucial to follow the present optimum agent in order to gain a better solution. Therefore, after ensuring that no conflicts are present between the neighboring agents in a swarm, the fittest location of the optimum agent is measured as given below.

$$\vec{P}\mathrm{D} = \left| X_{best} - r_{rand} \cdot \vec{\mathrm{p}}_p(x) \right| \tag{19}$$

In Equation (19), $\vec{PD}$ denotes the length between a better agent and the food origin, r_{rand} denotes a stochastic value within $[0, 1]$, X_{best} shows the optimal location, and the vector $\vec{P_p}(x)$ has the position of the tunicates at iteration x.

In order to ensure that the search agent is still closer to the optimal agent, their locations are calculated using Equation (20):

$$\vec{\mathrm{p}}_p(x) = \begin{cases} X_{best} + A \cdot \vec{P}D, \; if \; r_{rand} \geq 0.5 \\ X_{best} - A \cdot \vec{P}D, \; if \; r_{rand} < 0.5 \end{cases} \tag{20}$$

In Equation (20), $\vec{\mathrm{p}}_p(x)$ indicates the upgraded position of the agents at iteration x with respect to the better-scored location X_{best}.

The location of the existing agent is updated based on the position of two agents so as to model the swarming behaviors of the tunicates.

$$P_p\left(\vec{x} + 1\right) = \frac{\vec{\mathrm{p}}_p(x) + P_p\left(\vec{x} + 1\right)}{2 + c1} \tag{21}$$

The following steps demonstrate the flow of the original TSA approach:

Step 1: Initialize the population of tunicates $\vec{P}_p$.
Step 2: Set the maximum number of iterations and the original value for the parameter.
Step 3: Evaluate the fitness value of the exploration agents
Step 4: Explore a better agent in the search space.
Step 5: Upgrade the position of the exploration agent based on Equation (21).
Step 6: Return the newly updated agent to its boundaries.
Step 7: Measure the fitness cost of the updated search agents. In case of a solution superior to the prior solution, update $\vec{P}_p$ and keep the better solution in X_{best}.
Step 8: End the process if the terminating condition is satisfied. Or else, return to steps 5–8.
Step 9: Return the better solution (X_{best}).

Fitness choice is a key aspect of the TSA algorithm. An encoded outcome is employed herewith to determine the goodness of a candidate's performance. At present, the accuracy value is the major condition employed to plan an FF.

$$Fitness = \max(P) \tag{22}$$

$$P = \frac{TP}{TP + FP} \tag{23}$$

Here, TP and FP denote true and false positive values, respectively.

4. Results and Discussion

In this section, the results attained from the experimental investigation of the proposed TSA-LSTMRNN approach on human-generated text dataset and ChatGPT-generated text dataset are discussed. The approach was analyzed experimentally under a set of five experiments. Each experiment included a sample set comprising human-generated text and ChatGPT-generated text. The sample texts are given below.

Sample 1:

- Human-generated text: The selection on the menu was great and so were the prices.
- ChatGPT-generated text: The menu had a great selection and the prices were good.

Sample 2:

- Human-generated text: Point your finger at any item on the menu, order it and you won't be disappointed.
- ChatGPT-generated text: No matter what you order from the menu, you won't be disappointed.

Sample 3:

- Human-generated text: The one down note is the ventilation could use some upgrading.
- ChatGPT-generated text: The only drawback at this restaurant is that the ventilation could be improved.

Sample 4:

- Human-generated text: I believe that this place is a great stop for those with a huge belly and hankering for sushi.
- ChatGPT-generated text: I believe this place is a great stop for those with a big appetite and a desire for sushi.

Sample 5:

- Human-generated text: It was a truly special dining experience that exceeded all of my expectations.
- ChatGPT-generated text: I had a great experience at this restaurant and the services are beyond expectations.

The current study used a set of measures to examine the classification outcomes such as accuracy ($accu_y$), precision ($prec_n$), recall ($reca_l$), Mathew Correlation Coefficient (MCC), and F-score (F_{score}).

Precision measures the proportion of the correctly predicted positive instances of the overall instances that were predicted as positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{24}$$

Recall measures the proportion of the positive samples that were correctly classified.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{25}$$

Accuracy measures the proportion of the correctly classified samples (positives and negatives) against the total number of samples (number of samples that were classified).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{26}$$

F-score is a measure that combines the harmonic mean of the precision and recall measures.

$$\text{F} - \text{score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{27}$$

In Table 1 and Figure 3, the detection outcomes of the proposed TSA-LSTMRNN technique on the human-generated text dataset are provided. The results indicate that the proposed approach appropriately recognized the human-generated text dataset as positive and negative samples. For instance, in experiment 1, the TSA-LSTMRNN technique attained average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 93.17%, 92.56%, 93.17%, 92.77%, and 85.72%, respectively. Furthermore, in experiment 2, the TSA-LSTMRNN method reached average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 93.17%, 90.55%, 91.17%, 90.75%, and 87.71%, respectively. Moreover, in experiment 3, the proposed TSA-LSTMRNN algorithm

accomplished average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 87.83%, 87.71%, 87.83%, 87.77%, and 75.55%, correspondingly. Meanwhile, in experiment 4, the TSA-LSTMRNN system yielded average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 91.33%, 90.83%, 91.33%, 91.02%, and 82.16%, respectively. Finally, in experiment 5, the TSA-LSTMRNN approach reached average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 88.75%, 88.57%, 88.75%, 88.65%, and 77.32%, correspondingly.

Table 1. Detection outcomes of the proposed TSA-LSTMRNN approach on human-generated text dataset.

Human Dataset					
Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	**MCC**
Exp. 1					
Positive	91.00	96.30	91.00	93.57	85.72
Negative	95.33	88.82	95.33	91.96	85.72
Average	93.17	92.56	93.17	92.77	85.72
Exp. 2					
Positive	89.00	94.68	89.00	91.75	81.71
Negative	93.33	86.42	93.33	89.74	81.71
Average	91.17	90.55	91.17	90.75	81.71
Exp. 3					
Positive	89.00	89.90	89.00	89.45	75.55
Negative	86.67	85.53	86.67	86.09	75.55
Average	87.83	87.71	87.83	87.77	75.55
Exp. 4					
Positive	90.00	94.24	90.00	92.07	82.16
Negative	92.67	87.42	92.67	89.97	82.16
Average	91.33	90.83	91.33	91.02	82.16
Exp. 5					
Positive	89.50	90.86	89.50	90.18	77.32
Negative	88.00	86.27	88.00	87.13	77.32
Average	88.75	88.57	88.75	88.65	77.32

Figure 4 showcases the $accu_y$ values accomplished by the proposed TSA-LSTMRNN technique in training and validation methods on human-generated text database. The results indicate that the TSA-LSTMRNN technique achieved high $accu_y$ values over higher epochs. Also, a maximum validation $accu_y$ that was greater than the training $accu_y$ was achieved, which exhibits that the TSA-LSTMRNN technique establishes its capability on human-generated text databases.

A loss study was conducted for the proposed TSA-LSTMRNN system at the time of training and validation upon the human-generated text database, and the results are revealed in Figure 5. The results point out that the TSA-LSTMRNN method achieved similar training and validation loss values. It can be understood that the TSA-LSTMRNN technique operates effectively on human-generated text databases.

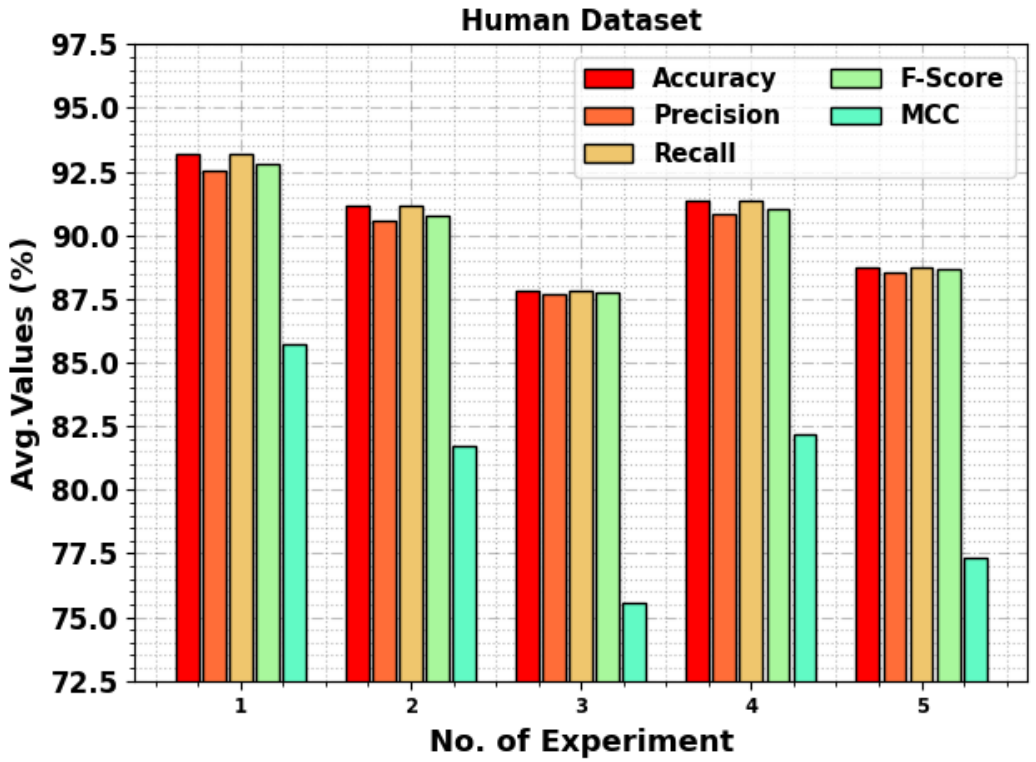

Figure 3. Average outcomes of the TSA-LSTMRNN approach on human-generated text dataset.

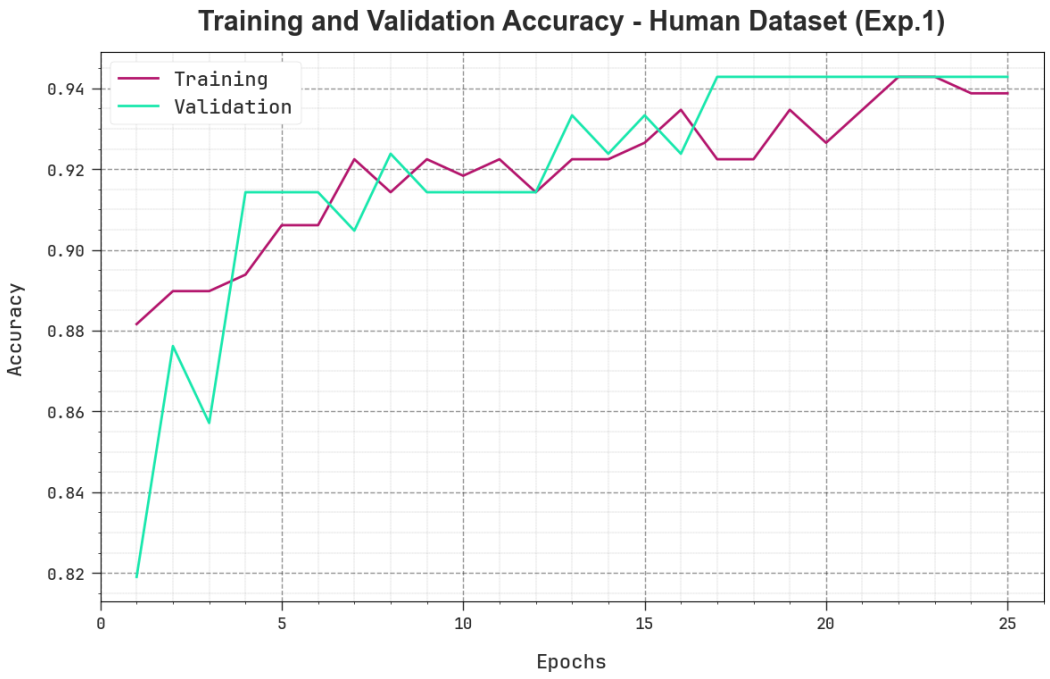

Figure 4. Accuracy curve of the TSA-LSTMRNN approach on human-generated text dataset.

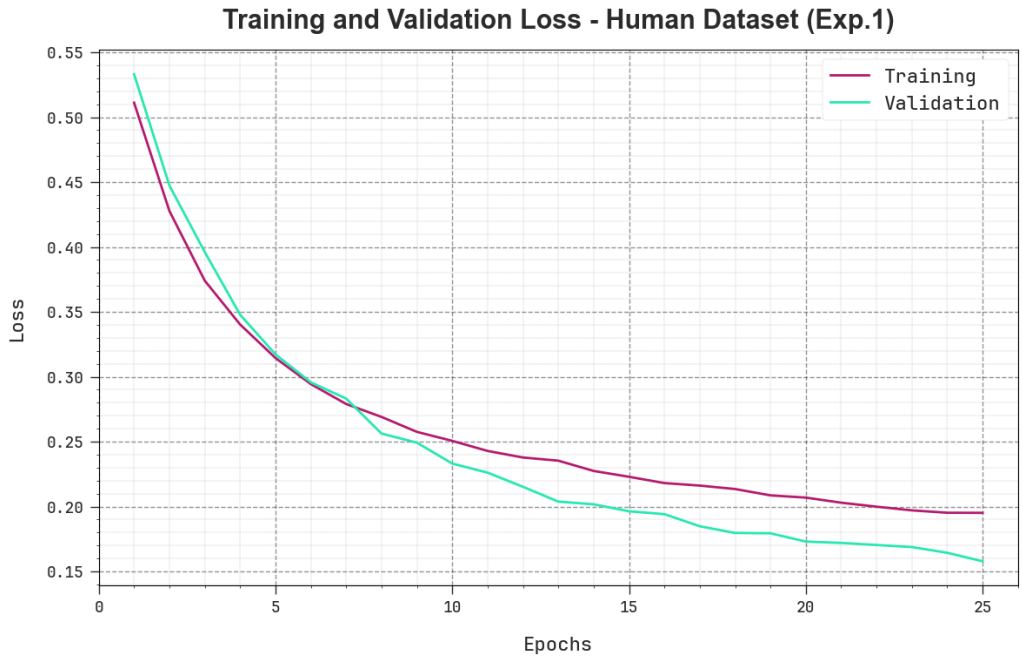

Figure 5. Loss curve of the TSA-LSTMRNN approach on human-generated text dataset.

A brief PR study was conducted on the TSA-LSTMRNN approach using the human-generated text database, and the results are shown in Figure 6. The outcomes indicate that the TSA-LSTMRNN algorithm enhanced the PR values. In addition to this, it is noticeable that the TSA-LSTMRNN technique can gain superior PR values in both the classes.

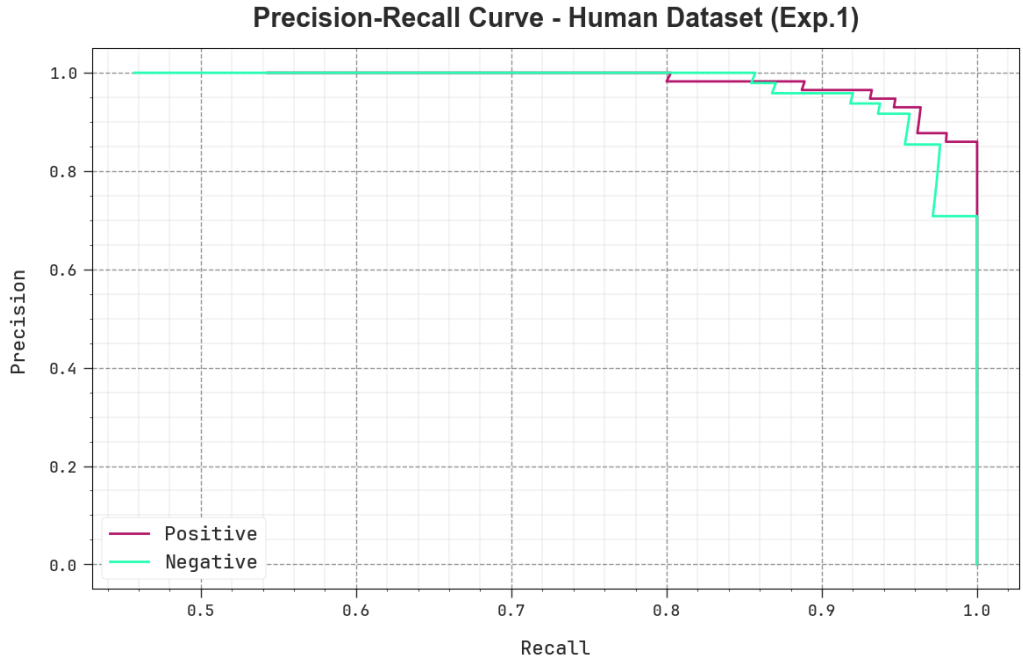

Figure 6. PR curve of the TSA-LSTMRNN approach on human-generated text dataset.

Figure 7 shows the results of the ROC analysis accomplished using the TSA-LSTMRNN technique on the human-generated text database. The results describe that the TSA-LSTMRNN approach enhanced the ROC values. Moreover, the TSA-LSTMRNN system extended the ROC values in both the class labels.

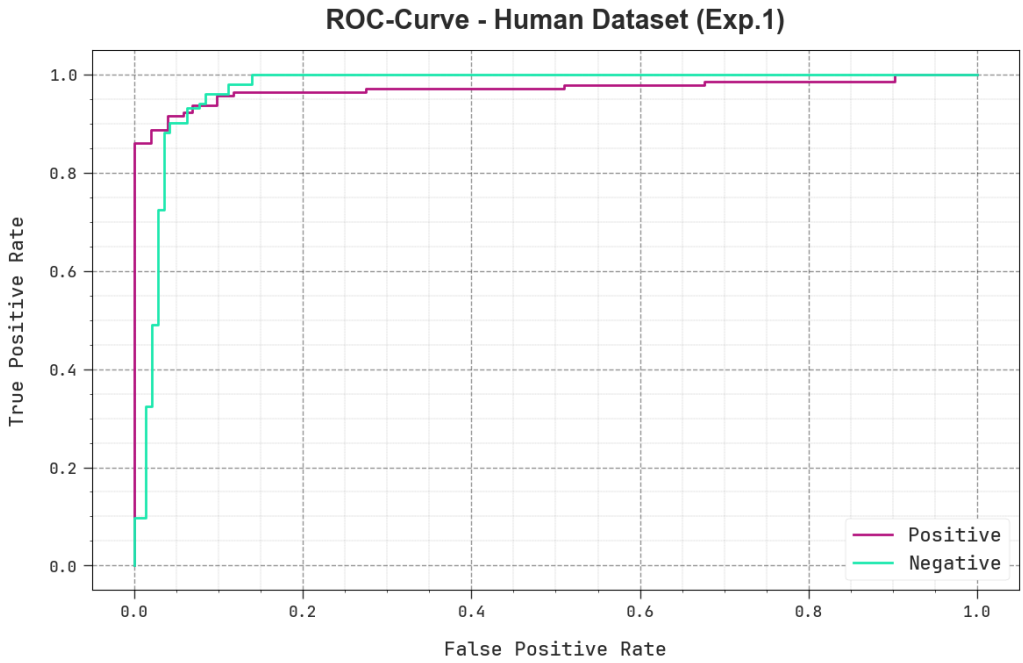

Figure 7. ROC curve of the TSA-LSTMRNN approach on human-generated text dataset.

In Table 2 and Figure 8, the detection outcomes of the TSA-LSTMRNN method on ChatGPT-generated text dataset are provided. The outcomes point out that the proposed TSA-LSTMRNN method properly recognized both positive and negative instances in the ChatGPT-generated text dataset. For instance, in experiment 1, the TSA-LSTMRNN method accomplished average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 78%, 81.51%, 78%, 78.65%, and 59.41%, correspondingly. Moreover, in experiment 2, the TSA-LSTMRNN technique attained average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 82.17%, 83.96%, 82.17%, 82.70%, and 66.10%, respectively. Additionally, in experiment 3, the proposed TSA-LSTMRNN methodology achieved average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 82.42%, 84.34%, 82.42%, 82.97%, and 66.73%, respectively. In the meantime, in experiment 4, the TSA-LSTMRNN system attained average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 81.58%, 84.19%, 81.58%, 82.24%, and 65.73%, correspondingly. Lastly, in experiment 5, the TSA-LSTMRNN technique yielded average $accu_y$, $prec_n$, $reca_l$, F_{score}, and MCC values of 93.83%, 94.52%, 93.83%, 94.12%, and 88.35%, correspondingly.

Table 2. Detection outcomes of the TSA-LSTMRNN approach on ChatGPT-generated text dataset.

ChatGPT Dataset					
Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	**MCC**
Exp. 1					
Positive	92.00	77.31	92.00	84.02	59.41
Negative	64.00	85.71	64.00	73.28	59.41
Average	78.00	81.51	78.00	78.65	59.41
Exp. 2					
Positive	91.00	81.98	91.00	86.26	66.10
Negative	73.33	85.94	73.33	79.14	66.10
Average	82.17	83.96	82.17	82.70	66.10

Table 2. *Cont.*

ChatGPT Dataset					
Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	**MCC**
Exp. 3					
Positive	91.50	82.06	91.50	86.52	66.73
Negative	73.33	86.61	73.33	79.42	66.73
Average	82.42	84.34	82.42	82.97	66.73
Exp. 4					
Positive	92.50	80.79	92.50	86.25	65.73
Negative	70.67	87.60	70.67	78.23	65.73
Average	81.58	84.19	81.58	82.24	65.73
Exp. 5					
Positive	97.00	93.27	97.00	95.10	88.35
Negative	90.67	95.77	90.67	93.15	88.35
Average	93.83	94.52	93.83	94.12	88.35

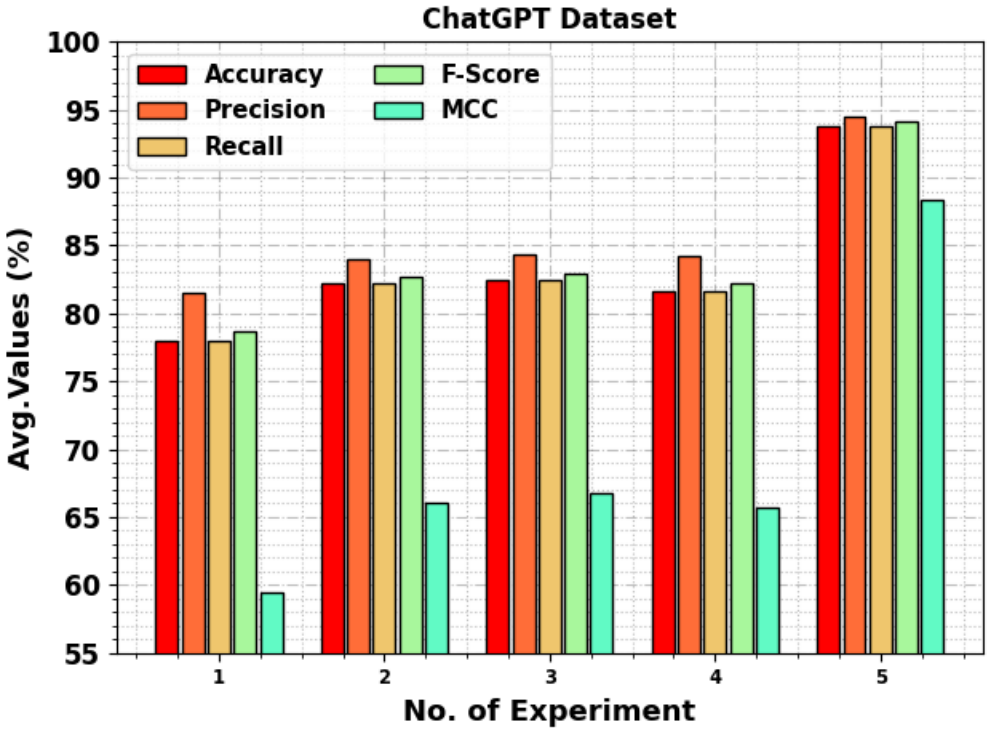

Figure 8. Average outcomes of the TSA-LSTMRNN approach on ChatGPT-generated text dataset.

Figure 9 shows the $accu_y$ curve plotted on the values achieved by the TSA-LSTMRNN technique in both training and validation models using the ChatGPT-generated text database. The outcomes imply that the TSA-LSTMRNN technique attained improved $accu_y$ values over superior epochs. Additionally, the enhanced validation $accu_y$ value exceeded the training $accu_y$ value, indicating that the TSA-LSTMRNN methodology works effectively on the ChatGPT-generated text database.

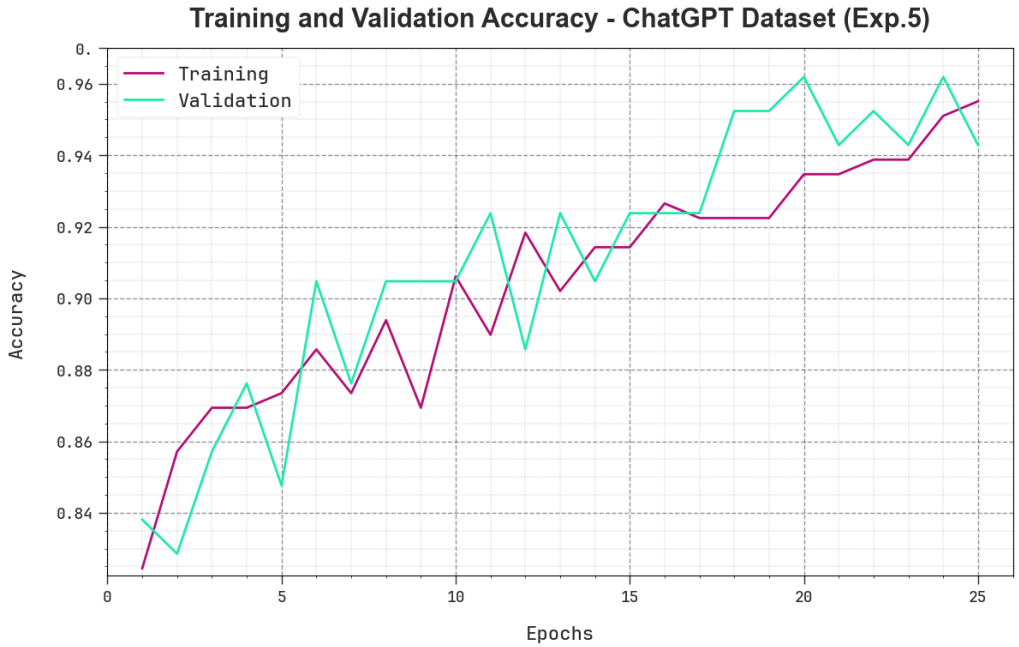

Figure 9. Accuracy curve of the TSA-LSTMRNN approach on ChatGPT-generated text dataset.

The loss curve of the TSA-LSTMRNN system, at the time of training and validation, on the ChatGPT-generated text database, is shown in Figure 10. The outcomes indicate that the TSA-LSTMRNN system realized adjacent training and validation loss values. Thus, it is obvious that the TSA-LSTMRNN system achieves capably on the ChatGPT-generated text database.

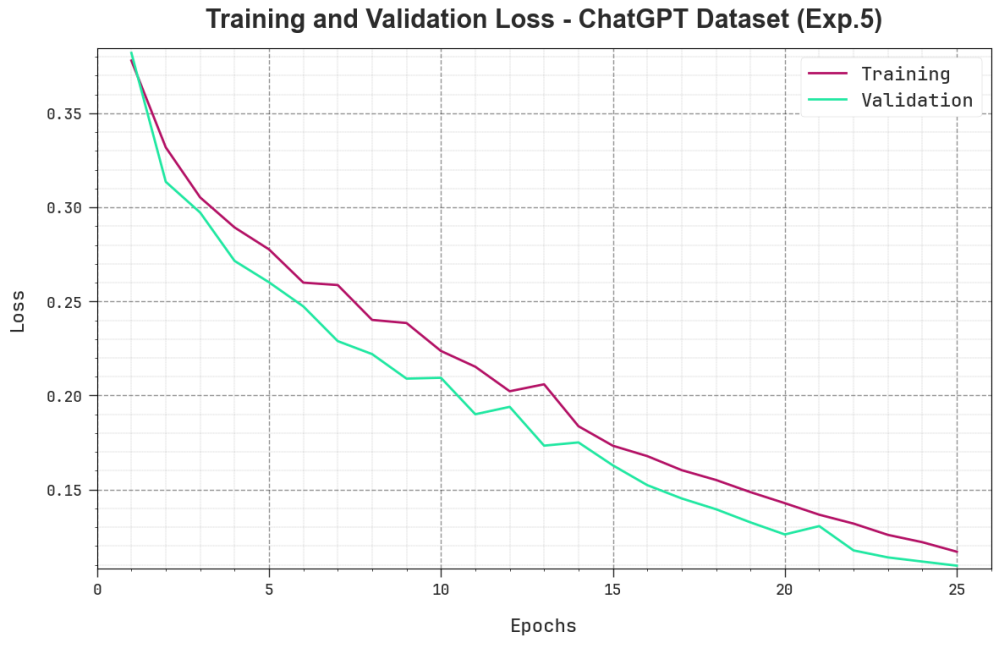

Figure 10. Loss curve of the TSA-LSTMRNN approach on ChatGPT-generated text dataset.

Figure 11 shows a brief PR curve of the TSA-LSTMRNN methodology upon the ChatGPT-generated text database. The results indicate that the TSA-LSTMRNN system affects superior values of PR. Afterward, it can be observed that the TSA-LSTMRNN technique obtained enhanced the PR values in both the class labels.

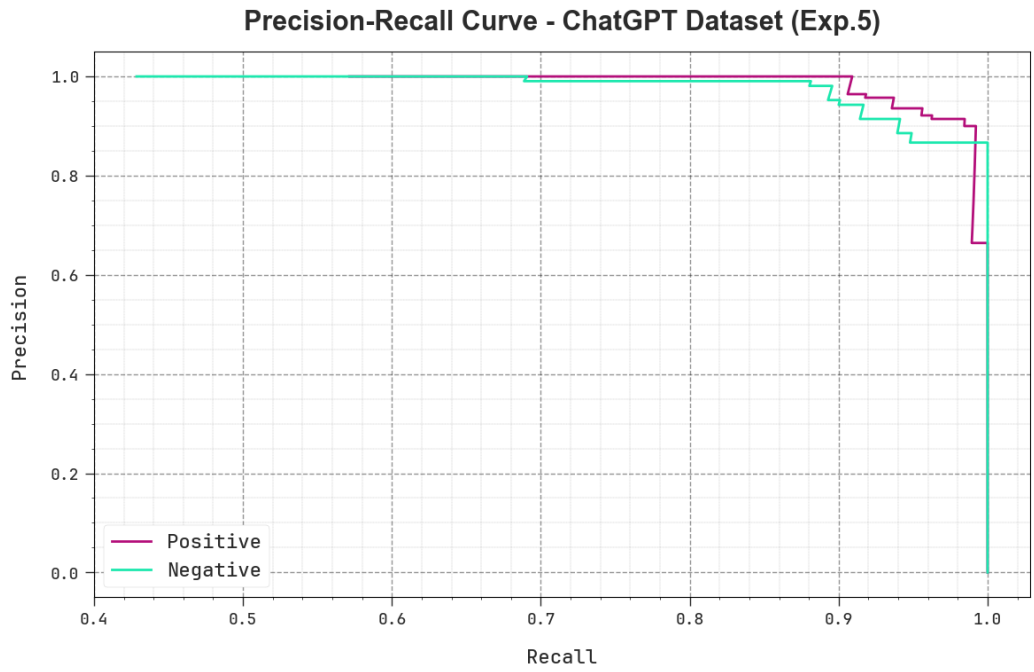

Figure 11. PR curve of the TSA-LSTMRNN approach on ChatGPT-generated text dataset.

In Figure 12, an ROC curve of the TSA-LSTMRNN technique is plotted for the ChatGPT-generated text database. The outcomes imply that the TSA-LSTMRNN approach accomplished higher ROC values. Moreover, it is clear that the TSA-LSTMRNN algorithm can yield higher ROC values in both the classes.

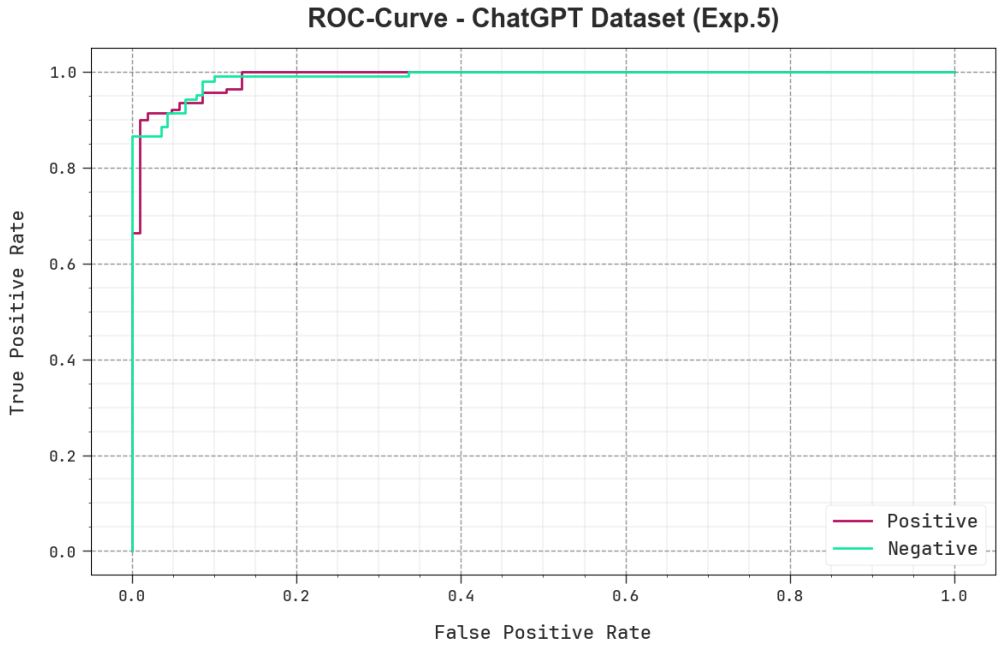

Figure 12. ROC curve of the TSA-LSTMRNN approach on ChatGPT generated text dataset.

In Table 3, the comprehensive comparison outcomes of the TSA-LSTMRNN system on both human-generated text dataset and the ChatGPT-generated text datasets are provided.

Table 3. Comparative outcomes of the TSA-LSTMRNN approach with other techniques on two datasets.

Accuracy (%)		
Algorithms	**Human Dataset**	**ChatGPT Dataset**
Decision Tree	86.70	88.01
SVM Model	86.72	82.30
XGBoost	83.86	84.96
CNN Model	84.08	86.93
ELM Model	89.80	86.34
TSA-LSTMRNN	93.17	93.83

Figure 13 represents the classification outcomes achieved by the proposed TSA-LSTMRNN technique and other ML approaches on human-generated text datasets. The result indicates that both XGBoost and CNN methods obtained the least $accu_y$ values, i.e., 83.86% and 84.08%, correspondingly. Then, the DT and SVM models reported moderately improved $accu_y$ values of 86.70% and 86.72%, respectively. Although the ELM model produced a near-optimal $accu_y$ of 89.80%, the TSA-LSTMRNN algorithm established its supremacy with an increased $accu_y$ of 93.17%.

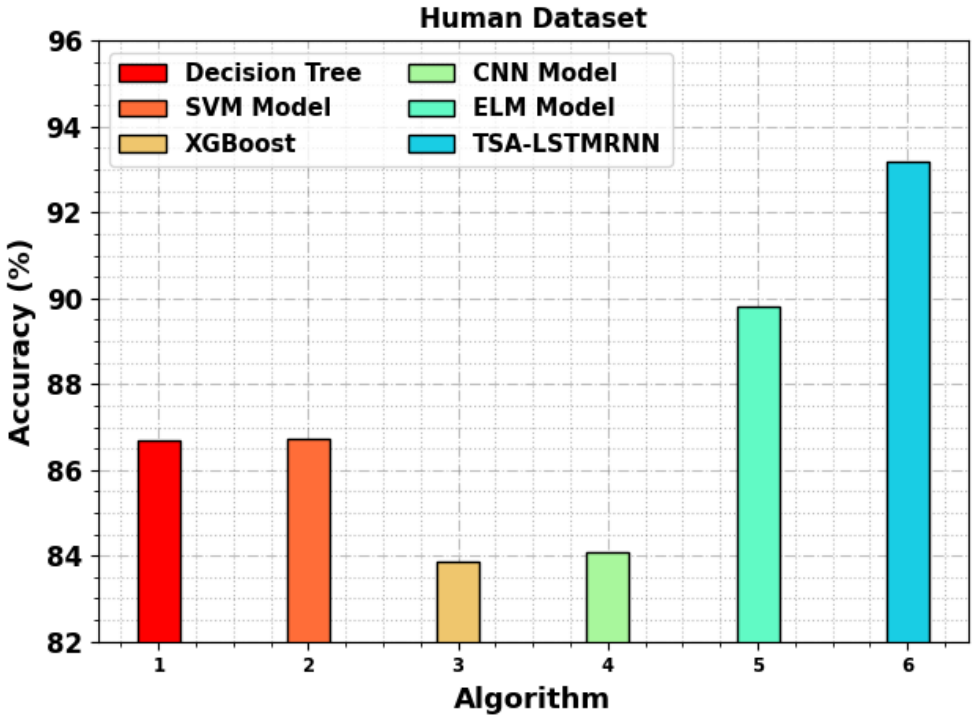

Figure 13. $Accu_y$ outcomes of the TSA-LSTMRNN approach on human-generated text dataset.

Figure 14 represents the classification results of the TSA-LSTMRNN approach and of other ML techniques on ChatGPT-generated text dataset. The outcome indicates that both SVM and XGBoost approaches produced the least $accu_y$ values, i.e., 82.30% and 84.96%, correspondingly, followed by the CNN and ELM approaches, which reported moderately improved $accu_y$ values of 86.93% and 86.34%, correspondingly. Though the DT approach accomplished a near-optimal $accu_y$ of 88.01%, the TSA-LSTMRNN method showed its supremacy with an improved $accu_y$ of 93.83%. These results highlight the significant performance of the proposed TSA-LSTMRNN technique.

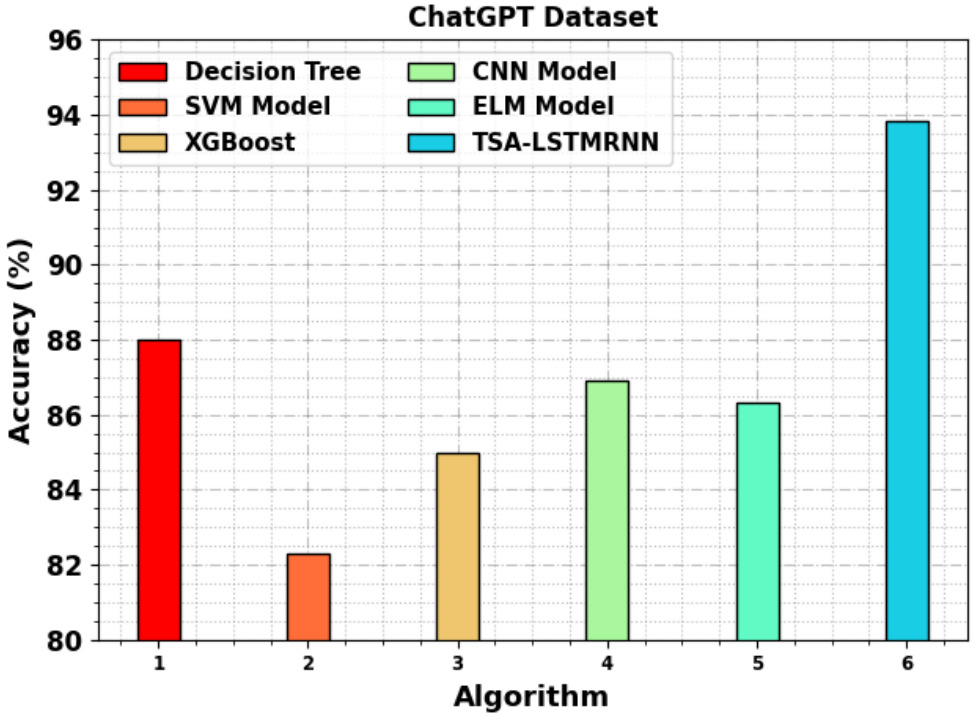

Figure 14. $Accu_y$ outcomes of the TSA-LSTMRNN approach on ChatGPT-generated text dataset.

5. Conclusions

In the current study, the authors developed an automated human-generated text and ChatGPT-generated text detection model, named the TSA-LSTMRNN approach. The purpose of the TSA-LSTMRNN technique is to investigate the model's decision and compute whether any particular pattern can be detected. Moreover, the TSA-LSTMRNN technique focuses on the design of TF-IDF, word embedding, and count vectorizers for the feature extraction process. For the detection and classification processes, the LSTMRNN model is used. At last, the TSA is exploited for the purpose of parameter selection for the LSTMRNN approach, which enables improved detection performance. The proposed TSA-LSTMRNN technique was experimentally validated using two benchmark databases, and the outcomes demonstrate the superior efficiency of the TSA-LSTMRNN algorithm over other recent systems. In the future, the performance of the TSA-LSTMRNN method can be boosted with the help of ensemble models.

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

Funding: This research work was funded by Institutional Fund Projects under grant no. (IFPFP-266-22).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Acknowledgments: The authors gratefully acknowledge technical and financial support provided by the Ministry of Education and Deanship of Scientific Research (DSR), King Abdulaziz University (KAU), Jeddah, Saudi Arabia.

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

References

1. Gao, C.A.; Howard, F.M.; Markov, N.S.; Dyer, E.C.; Ramesh, S.; Luo, Y.; Pearson, A.T. Comparing scientific abstracts generated by ChatGPT to real abstracts with detectors and blinded human reviewers. *NPJ Digit. Med.* **2023**, *6*, 75. [CrossRef] [PubMed]
2. Pavlik, J.V. Collaborating with ChatGPT: Considering the Implications of Generative Artificial Intelligence for Journalism and Media Education. *J. Mass Commun. Educ.* **2023**, *78*, 84–93. [CrossRef]
3. Qadir, J. Engineering Education in the Era of ChatGPT: Promise and Pitfalls of Generative AI for Education. In Proceedings of the 2023 IEEE Global Engineering Education Conference (EDUCON), Kuwait, Kuwait, 1–4 May 2022.
4. Dergaa, I.; Chamari, K.; Zmijewski, P.; Saad, H.B. From human writing to artificial intelligence generated text: Examining the prospects and potential threats of ChatGPT in academic writing. *Biol. Sport* **2023**, *40*, 615–622. [CrossRef] [PubMed]
5. Shen, Y.; Heacock, L.; Elias, J.; Hentel, K.D.; Reig, B.; Shih, G.; Moy, L. ChatGPT and other large language models are double-edged swords. *Radiology* **2023**, *307*, e230163. [CrossRef] [PubMed]
6. Shahriar, S.; Hayawi, K. Let's have a chat! A Conversation with ChatGPT: Technology, Applications, and Limitations. *arXiv* **2023**, arXiv:2302.13817. [CrossRef]
7. Sallam, M. The Utility of ChatGPT as an Example of Large Language Models in Healthcare Education, Research and Practice: Systematic Review on the Future Perspectives and Potential Limitations. *MedRxiv* **2023**. [CrossRef]
8. OguzhanTopsakal, E. Framework for A Foreign Language Teaching Software for Children Utilizing AR, Voicebots and ChatGPT (Large Language Models). *J. Cogn. Syst.* **2022**, *7*, 33–38.
9. Mhlanga, D. *Open AI in Education, the Responsible and Ethical Use of ChatGPT Towards Lifelong Learning*; Springer: Berlin/Heidelberg, Germany, 2023.
10. Kasneci, E.; Seßler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; Gasser, U.; Groh, G.; Günnemann, S.; Hüllermeier, E. ChatGPT for good? On opportunities and challenges of large language models for education. *Learn. Individ. Differ.* **2023**, *103*, 102274. [CrossRef]
11. Yu, P.; Chen, J.; Feng, X.; Xia, Z. CHEAT: A Large-scale Dataset for Detecting ChatGPT-writtEn AbsTracts. *arXiv* **2023**, arXiv:2304.12008.
12. Liao, W.; Liu, Z.; Dai, H.; Xu, S.; Wu, Z.; Zhang, Y.; Huang, X.; Zhu, D.; Cai, H.; Liu, T.; et al. Differentiate ChatGPT-generated and Human-written Medical Texts. *arXiv* **2023**, arXiv:2304.11567.
13. Alamleh, H.; AlQahtani, A.A.S.; ElSaid, A. Distinguishing Human-Written and ChatGPT-Generated Text Using Machine Learning. In *2023 Systems and Information Engineering Design Symposium (SIEDS)*; IEEE: Piscataway, NJ, USA, 2023; pp. 154–158.
14. Chen, Y.; Kang, H.; Zhai, V.; Li, L.; Singh, R.; Ramakrishnan, B. GPT-Sentinel: Distinguishing Human and ChatGPT Generated Content. *arXiv* **2023**, arXiv:2305.07969.
15. Pardos, Z.A.; Bhandari, S. Learning gain differences between ChatGPT and human tutor-generated algebra hints. *arXiv* **2023**, arXiv:2302.06871.
16. Hamed, A.A.; Wu, X. Improving Detection of ChatGPT-Generated Fake Science Using Real Publication Text: Introducing xFakeBibs a Supervised-Learning Network Algorithm. *Preprints* **2023**, *in press*. [CrossRef]
17. Perkins, M. Academic Integrity Considerations of AI Large Language Models in the post-pandemic era: ChatGPT and Beyond. *J. Univ. Teach. Learn. Pract.* **2023**, *20*, 7. [CrossRef]
18. Maddigan, P.; Susnjak, T. Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models. *arXiv* **2023**, arXiv:2302.02094. [CrossRef]
19. Jalil, Z.; Abbasi, A.; Javed, A.R.; Badruddin Khan, M.; Abul Hasanat, M.H.; Malik, K.M.; Saudagar, A.K.J. COVID-19 related sentiment analysis using state-of-the-art machine learning and deep learning techniques. *Front. Public Health* **2022**, *9*, 2276. [CrossRef] [PubMed]
20. Singh, C.; Imam, T.; Wibowo, S.; Grandhi, S. A deep learning approach for sentiment analysis of COVID-19 reviews. *Appl. Sci.* **2022**, *12*, 3709. [CrossRef]
21. Houssein, E.H.; Helmy, B.E.D.; Elngar, A.A.; Abdelminaam, D.S.; Shaban, H. An improved tunicate swarm algorithm for global optimization and image segmentation. *IEEE Access* **2021**, *9*, 56066–56092. [CrossRef]