technologies

Article

A Comparative Analysis on Suicidal Ideation Detection Using NLP, Machine, and Deep Learning

Rezaul Haque [1], Naimul Islam [1], Maidul Islam [2] and Md Manjurul Ahsan [3,*]

[1] Department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh; rezaulh603@gmail.com (R.H.); naimul.islam.pulak@gmail.com (N.I.)
[2] Department of Aerospace Engineering, RMIT University, Melbourne, VIC 3000, Australia; s3801461@student.rmit.edu.au
[3] School of Industrial & Systems Engineering, University of Oklahoma, Norman, OK 73019, USA
* Correspondence: ahsan@ou.edu

Abstract: Social networks are essential resources to obtain information about people's opinions and feelings towards various issues as they share their views with their friends and family. Suicidal ideation detection via online social network analysis has emerged as an essential research topic with significant difficulties in the fields of NLP and psychology in recent years. With the proper exploitation of the information in social media, the complicated early symptoms of suicidal ideations can be discovered and hence, it can save many lives. This study offers a comparative analysis of multiple machine learning and deep learning models to identify suicidal thoughts from the social media platform Twitter. The principal purpose of our research is to achieve better model performance than prior research works to recognize early indications with high accuracy and avoid suicide attempts. We applied text pre-processing and feature extraction approaches such as CountVectorizer and word embedding, and trained several machine learning and deep learning models for such a goal. Experiments were conducted on a dataset of 49,178 instances retrieved from live tweets by 18 suicidal and non-suicidal keywords using Python Tweepy API. Our experimental findings reveal that the RF model can achieve the highest classification score among machine learning algorithms, with an accuracy of 93% and an F1 score of 0.92. However, training the deep learning classifiers with word embedding increases the performance of ML models, where the BiLSTM model reaches an accuracy of 93.6% and a 0.93 F1 score.

Keywords: suicide ideation; text classification; machine learning; NLP; text pre-processing; deep learning

Citation: Haque, R.; Islam, N.; Islam, M.; Ahsan, M.M. A Comparative Analysis on Suicidal Ideation Detection Using NLP, Machine, and Deep Learning. *Technologies* **2022**, *10*, 57. https://doi.org/10.3390/technologies10030057

Academic Editors: Manoj Gupta, Eugene Wong and Gwanggil Jeon

Received: 30 March 2022
Accepted: 27 April 2022
Published: 29 April 2022

1. Introduction

Suicide is regarded as one of today's most serious public health issues. Around 0.7 million individuals die every year, and many more, particularly the young and middle-aged, attempt suicide [1]. For persons aged 10 to 34, it is the second-largest cause of death [2]. Suicidal ideation affects individuals of all ages all over the globe due to shock, rage, guilt, and other symptoms of melancholy or anxiety. Suicidal ideation, often known as suicidal thoughts, refers to the conceptualizing or meanderings about terminating one's life. Although most people who have suicidal thoughts do not attempt suicide, long-term depression may lead to suicide if depressed persons do not get effective counseling [3]. Their suicidal ideation can be cured with the help of healthcare experts and drugs, but most of them shun medical treatments owing to societal stigma. Instead, individuals prefer to convey their suicidal intentions via social media. However, since mental illness can be detected and addressed, early detection of indications or risk factors may be the most effective strategy to avoid suicidal ideation.

Over the years, it has been discovered that online social media data, particularly tweets, include predictive information for various mental health disorders, including depression and suicide. The information provided on Twitter can be helpful in analyzing people's

suicidal thoughts. However, with the widespread usage of internet and technology, the number of tweets has been increasing explosively. It will be very challenging and time consuming for us to go through these tweets and identify people with suicidal ideations. Early detection of suicidal ideations from tweets will help medical experts identify the suicidal intentions of an individual and provide appropriate treatment. In general, an automated suicidal ideation detection system would allow medical professionals to save many lives by detecting the early symptoms of depression from online tweets.

Sentiment Analysis (SA) is a growing field that automatically captures user sentiment [4,5]. We can recognize early suicidal thoughts and avert the majority of suicide attempts by properly using a combination of information from social media and SA. As a result, Machine Learning (ML) and Natural Language Processing (NLP) have arisen as techniques for predicting suicidal intent from social media data. In addition, Deep Learning (DL) architectures provide significant advantages for detecting suicidal thoughts since they perform at very high accuracy with lower-level engineering and processing. Prior research papers that have identified suicidal ideations from tweets using ML algorithms have conducted their studies on a limited dataset. The research of [6] utilized several ML models to conduct depression detection on a collection of 15,000 tweets. Due to the small amount of data, their ML models suffered from poor accuracy. A similar work can be shown by [7], where the author improved the performance of ML classifiers on a dataset of 50,000 tweets. They collected the tweets from news articles and websites using several keywords and manually labeled them to perform binary classification. In [8], the authors proposed an automatic depression detection system using ML models where the dataset was created from a Russian social networking site Vkontakte. All of these papers were unable to achieve a good accuracy score as their focus was on training ML algorithms on a small collection of tweets. With the use of appropriate annotation rules on a huge number of tweets and training DL models, it is possible to improve the classification score of ML models. Previously, DL classifiers have been used to identify the suicidal intentions of social media users with great accuracy [9]. Most of their research was based on humanly annotated datasets, collected from different suicide forums [10] and subreddits [11]. However, tweets are different in nature compared to Reddit or suicide forum posts, as users only get 280 characters to interact with others. The authors in [12] proposed a novel attention-based relational network that can identify mental disorders from 4800 tweets labeled into four classes with an accuracy score of 83%. However, the results can be further improved by collecting more tweets and using appropriate preprocessing and feature extraction techniques.

As discussed above, for the task of suicidal ideation detection from Twitter, most of the studies have focused on implementing only ML algorithms, which results in poor accuracy scores. In addition, DL algorithms were used in research on datasets of Reddit, suicide forums, or a small number of tweets. There is no work on a dataset of around 50,000 tweets, where DL classifiers attain great accuracy. Furthermore, there is still a lack of comparative analysis between the performances of ML and DL classifiers for the task of identifying suicidal ideation in live Tweets. The main contribution of this research is tackling these gaps by performing an experimental study to evaluate the performance of five ML and four DL classifiers on a dataset of around 50,000 tweets labeled as 'suicide' and 'non-suicide'. The primary purpose of our study is to use effective NLP and feature extraction techniques to train several ML and DL models to identify suicidal thoughts on Twitter and provide a comparative analysis between the performance of the classifiers. To our knowledge, this is the first research where around 50,000 tweets have been used to conduct experiments on ML and DL classifiers for the task of suicidal ideation detection. Our study represents that a DL model can outperform typical ML methods for the task of suicidal ideation detection if text pre-processing is appropriately executed. Our contributions are mentioned below:

- Using the Tweepy API [13], we built a dataset of 49,178 tweets gathered live from Twitter with 18 keywords associated with suicidal ideation. Furthermore, we annotated the tweets using VADER and TextBlob, labeling them as non-suicidal or suicidal;

- Tweets include informal language; thus, we used NLTK packages to clean the noisy text data to make the user's content seem more evident and enhance the text analysis. Feature extraction techniques such as CountVectorizer and word embedding were used for ML and DL, respectively, which helped lead to more accurate suicidal detection;
- Several DL classifiers such as Long-Short Term Memory (LSTM), Bi-directional LSTM (BiLSTM), Gated Recurrent Unit (GRU), Bi-directional GRU (BiGRU), and combined model of CNN and LSTM (C-LSTM) were trained using Keras, a high-level API of TensorFlow. Furthermore, their performance was evaluated based on accuracy, AUC, precision, recall, and F1-score;
- The performance of DL was compared with traditional ML approaches such as Random Forrest (RF), Support Vector classifier (SVC), Stochastic Gradient Descent classifier (SGD), Logistic Regression (LR), and Multinomial Naive Bayes (MNB) classifier.

The work is organized into five more sections, in addition to this introduction. The second section lays out the related research works on suicidal ideation detection. Research methodology is discussed in the third section. Experimental findings and analysis are presented in the fourth section. The fifth section interprets the discussion and lastly, in the sixth and final section, the concluding remarks and future works are summarized.

2. Related Works

Sentiment analysis is regarded as one of the fastest-growing study subjects in the field of computer science. According to [14], sentiment analysis can be traced back to the surveys on public sentiment research at the end of the 20th century. In addition, test analysis was started by a group known as computational linguistics around the 1990s. Computer-based sentiment analysis has emerged more prominently with the availability of texts on the web. Apart from that, various fields in terms of identifying the underlying emotions from any text or voice message were improved massively because of the impact of text-availability on the web. Several works on sentiment analysis using several approaches can be found in the literature.

In recent years, much research has been completed to investigate the link between mental health and linguistic use to get novel insights into identifying suicidal thoughts. Previous works on detecting suicidal thoughts made use of language elements from the psychiatric literature, such as LIWC [15], emotion features [16], and suicide notes [17]. However, the fundamental disadvantage of this analysis approach is that it utilizes language-specific strategies that evaluate individual posts in isolation and will not perform well enough when dealing with diverse or enormous quantities of data.

The use of social media in combination with NLP for mental health research is becoming more popular among researchers. With its mental health-related forums, online social media data have been a growing research field in sentiment analysis, such as Michael M. Tadesse [18], who developed a combined model of LDA, LIWCA, bigram, and MLP to reach an accuracy of 90%. In [6–8], the authors used a similar strategy to collect data from Twitter and train multiple ML approaches to classify suicidal thoughts.

DL approaches as LSTM and CNN have already made significant progress in the area of NLP, thanks to the rising popularity of word embedding. Since ML methods have certain limitations, such as dimension explosion, data sparsity, and time consumption, they are unsuitable for all applications. DL considerably improves traditional ML techniques by extracting more abstract characteristics from input data by increasing the number of layers in the model, making the model's final classification information more consistent and accurate. The ability of DL models compared to other ML classifiers was shown in [19,20], where they obtained greater prediction accuracy using DL models for suicidal ideation detection. In [21], Tadesse et al. employed a CNN-LSTM combination model with word2vec to predict suicidal ideation with a 93.8% accuracy, owing to the model's ability to extract long-term global dependencies as well as local semantic information. However, their research was conducted on a small dataset of suicidal ideation content.

Despite its solid foundation, the existing research in depression detection lacks certain critical key factors. Research is scarce on a comparative study of suicidal ideation in which all traditional ML classifiers are compared to DL models such as BiLSTM, LSTM, GRU, BiGRU, and CLSTM that use word embedding for feature extraction with high accuracy performance. Furthermore, most of them conducted their research on limited datasets gathered from social media, which must be carefully pre-processed to attain better results.

3. Experimental Methods

3.1. Data Collection and Labeling

3.1.1. Data Collection

Data collection is the initial step in the analysis process since we need data to train our classifiers. However, the absence of a public dataset is one of the most significant obstacles in the field of suicidal ideation detection. Conventionally, it has been complex extracting data that are related to mental illnesses or suicidal ideation because of social stigma. However, a growing number of people are surfing the Internet to vent their frustration, to seek help, and to discuss mental health issues. We chose Twitter as our primary source of data since it has been shown to be effective in assessing mental conditions, such as suicidal ideation [22–24]. Here, the main idea is to collect different types of posts that are connected to suicide other than those that more directly express suicidal ideation. To maintain the privacy of the individuals in the dataset, we do not present direct quotes from any data or any identifying information. Furthermore, a unique ID is generated as a replacement of their personal information to preserve the users' privacy.

Tweepy, a tweet extraction API, allows us to query through historical tweets with tokenized terms. We required a list of suicide-related search phrases that could be used as a query to acquire raw Twitter data. Therefore, we came up with suicide-related phrases in two stages. Firstly, we looked at several suicide-related tweets. We became acquainted with expressions expressing suicidal thoughts by reading tweets, such as "want to die", "kill myself", and so on. We attempted to collect the phrases used frequently to indicate suicidal thoughts or suicidal ideation. Secondly, we looked through some suicide-related research papers. These publications provided us with useful information regarding suicidal expressions. We extracted a significant list of terms from Twitter based on the previous two processes, including keywords connected to suicide or self-harm. We collected real-time tweets using the Tweepy API using the final keywords list. We manually went through the gathered tweets and modified the terms' list by inserting, removing, and changing terms. We terminated the operation after discovering that the majority of the tweets retrieved were about suicides. It took about 2 to 3 weeks to compile the list of suicide-related key phrases. From 20 February 2021 to 13 May 2021, a total of 65,516 tweets with these phrases were extracted. The following are some of the comprehensive lists of suicide-related terms:

> *Anxiety disorder, depression, help me out, suffering, trapped, kill myself, suffering, sleep forever, my sad life, suicide, struggle, depressed me, stressed out, crying, want to die, emotionally weak, hate myself, burden.*

3.1.2. Data Annotation

The sentiment of the tweets was not known when they were gathered. For example, while collecting tweets with a term, it was unknown whether the tweet was intended for suicide awareness and prevention, the individual was discussing suicide ideas such as ways to kill himself, the tweet reflected a third person's suicide, or the tweet utilized suicide as a narrative. While several of the gathered tweets included suicidal-related phrases, they may have been discussing a suicide film or campaign that did not convey suicidal thoughts. We annotated the gathered tweets in two stages. First, we annotated the tweets with VADER and TextBlob, a Python program that extracts sentiment polarity (positive, neutral, or negative) from the text. Then, using the annotation rules in Table 1, we manually reviewed and corrected the labeled tweets. In total, there were 65,516 tweets in our Twitter dataset, with 24,458 tweets (around 38%) containing suicidal thoughts. Due to

the imbalanced nature of this dataset, the training dataset had a maldistribution of classes, resulting in poor predictive performance, particularly for the minority (suicidal) category. We avoided the imbalance problem by deleting 16,338 non-suicidal tweets. Finally, our dataset contained 49,178 tweets with suicidal and non-suicidal class accounting for 50.3% and 49.7%, respectively, shown in Figure 1.

Table 1. Data Annotation Rules.

Label	Rule	Examples
Suicidal	Expressing suicidal thoughts	'I've got nothing left to live for', 'I hate my life sometimes'
	Potential suicidal thoughts	'I want myself dead I hate men and I hate this planet', 'I want to shoot myself'
Non-suicidal	Discussing suicide	'He wanted to die his partner wanted to live seriously you must watch', 'This dudes life is worthless'
	Irrelevant to suicide	'Can this back pain just end please I'm tired', 'My boyfriend's phone is dying a slow death this would be great for him'

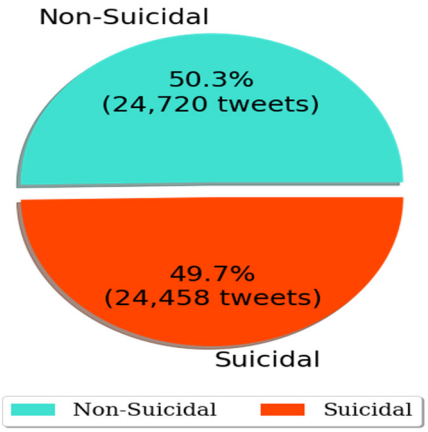


Figure 1. Class distribution of experimental dataset.

3.2. Pre-Processing

Preparing data entails converting raw data into a more usable format that can be fed into a classifier for improved performance. We correctly cleaned the textual data before executing the suicidal ideation detection task since most tweets included a significant noise. Figure 2 shows an example of a tweet under the suicidal class after performing all the pre-processing steps.

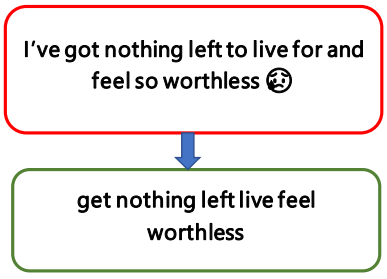


Figure 2. Example of an actual and pre-processed suicidal tweet.

3.2.1. Word Transformation

The majority of the tweets in our sample are made up of short conversational phrases and contractions. We used word segmentation to tokenize the text and replaced it with its complete form to turn them into meaningful words. For example, every tweet containing the phrase "AFAIK" was replaced with "As Far As I Know".

3.2.2. Removing Irrelevant Characters

ML models cannot comprehend nonsensical characters. Their presence in the text causes it to become noisy; thus, they must be deleted from tweets. Emojis, URLs, punctuation, whitespace, numerals, and user references were stripped from the text using regular expressions.

3.2.3. Stemming and Lemmatization

Stemming is a word-shortening approach that seeks to reduce a term to its root. Lemmatization is identical to stemming but combines vocabulary and morphological analysis to restore words to their dictionary form. We applied NLTK's Porter Stemmer and Wordnet Lemmatizer to perform stemming and lemmatization, which improved text categorization accuracy.

3.2.4. Stop Words Removal

A list of unimportant, frequently occurring words with little or no grammatical responsibility for text classification is known as a stop words list. We used NLTK's stop words corpus to eliminate them, to decrease the low-level information in our text and concentrate more on the relevant information. We also took out less frequently used words from the tweets.

3.3. Feature Extraction and Training

One dimensionality reduction approach used in ML is feature extraction, which maps higher dimensional data into a collection of low dimensional feature sets. Extracting valuable and crucial characteristics improves the performance of ML models while reducing computing complexity [25]. So, we will use feature extraction algorithms to turn text into a matrix (or vector) of possibilities. Word CountVectorizer and word embedding are two of the most successful feature extraction methods frequently utilized in ML and DL for text classification among all feature extraction approaches.

3.3.1. Count Vectorizer with ML Training

Text classification requires converting source documents into a vector representation. We implemented a word CountVectorizer using the unigram technique to vectorize our tweets by transforming the source texts into vector representations with the same length as the tweets and an integer count of the number of times a word occurred in each tweet. We obtained a lexicon of 36,121 unique words present in all tweets, which we fed into our vectorizer so that the ML models could conduct classification. To train our model, we split the data into two sets: training and testing, which were 80% and 20%, respectively. Various ML methods for performing suicidal ideation categorization are detailed in the related work. We used five ML algorithms, such as LR, SVC, RF, MNB, and SGD, to determine the existence of suicidal ideation among the users.

Logistic Regression employs a sigmoid function to convert the result to a probability. This minimizes the cost function to attain the best probability. The classifier was penalized using the 'l2' norm, stopping criteria with a tolerance of 0.0001, and the 'liblinear' optimizer for a maximum of 200 iterations. With a random state 42, the default value of the inverse of regularization strength is utilized. SVC estimates a hyperplane based on a feature set to categorize data points. We trained the SVC model with kernel type 'rbf', 0.0001 stopping criterion tolerance, and random state 42. One of the two fundamental Naive Bayes variations used in text classification is the MNB method, which implements the Naive

Bayes algorithm for multinomial distributed data. It presupposes that, given the class, the predictive qualities are conditionally independent, and it indicates that there are no hidden or underlying features that may impact the classification process. To train the MNB model, we employed Laplace smoothing and class prior probabilities based on the data. The RF algorithm is an ensemble-based approach that uses bagging techniques to train various decision trees. It has a low bias and a decent variance in prediction when it comes to categorization. Consequently, the algorithm can keep track of characteristics and predictors, and it is feasible to get excellent accuracy by utilizing it in text classification tasks. We used 200 trees for building a forest whose split quality was determined by 'entropy'.

3.3.2. Word Embedding with DL Training

Word embedding techniques, represented by DL, have recently received much attention and are now commonly employed in text classification. The Word2Vec word embedding generator seeks to discover the interpretation and semantic relationships between words by examining the co-occurrence of terms in texts within a specified corpus. This technique uses ML and statistics to model the proper context of words and generate a vector representation for each word in the dataset. With a post-padding of 60 vectors, we turned the encoded words of the tweets into a padded sequence. A matrix containing these padding sequences as input will be inserted into the embedding layer of 128 dimensions of the deployed DL models to translate the encoded textual comments to the correct word embeddings. We divided our dataset into training, validation, and testing sets of 80%, 10%, and 10%, respectively, where the model is trained on the training and validation sets.

RNN's implementation of the tree structure approach to extract the semantics of a phrase has been used to complete text classification, with good results in the previous research studies. However, developing a textual tree structure takes a long time for lengthy phrases of tweets, since the created model suffers from gradient vanishing and exploding. Two forms of RNN-based design approach, LSTM and GRU, were created, both of which include a gating mechanism to address the limitations of RNN [26]. It includes a 'forget' gate that allows the network to encapsulate longer-term relationships without encountering the vanishing gradient issue. GRUs are less complicated than LSTMs since they employ fewer parameters and do not need a memory unit [27]. The LSTM and GRU models both include bidirectional and directional approaches. To identify the best performing model for the subsequent trials, we used a few Deep Neural Network(DNN) architectures–BILSTM, LSTM, BIGRU, and CLSTM (a mix of LSTM and CNN). These models were trained using a 128 batch size, a memory unit of 128, Adam optimizer, a 0.0001 initial learning rate, and the Relu activation function. However, owing to the enormous set of parameters to be learned, DNNs are prone to overfitting. Due to noise, the learning accuracy of DNN models stops increasing or even worsens beyond a certain point. In order to minimize overfitting, we adjusted our network architecture and regularization parameters to fit the training data. In addition, we included ReduceLROnPlateau, an early stopping technique, which lowers the learning rate when the model stops improving. The proposed architecture of the CLSTM network is shown in Figure 3.

3.4. Evaluation Matrices

We opted to use the standard classification metrics, such as accuracy, precision, recall, f1 score, AUC, and confusion matrix, to evaluate the models. For binary classification tasks, such metrics are simple and can be obtained using Equations (1)–(4):

$$accuracy = \frac{TP + TN}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{FP}} \tag{1}$$

$$precision = \frac{TP}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

$$recall = \frac{TP}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

$$f1 - score = \frac{2 * (\ precision * recall)}{\text{precision} + \text{recall}} \tag{4}$$

where, TP (True Positive) stands for the number of accurate positive predictions, FP (False Positive) for wrong positive predictions, FN (False Negative) for incorrect negative predictions, and TN (True Negative) for correct negative predictions.

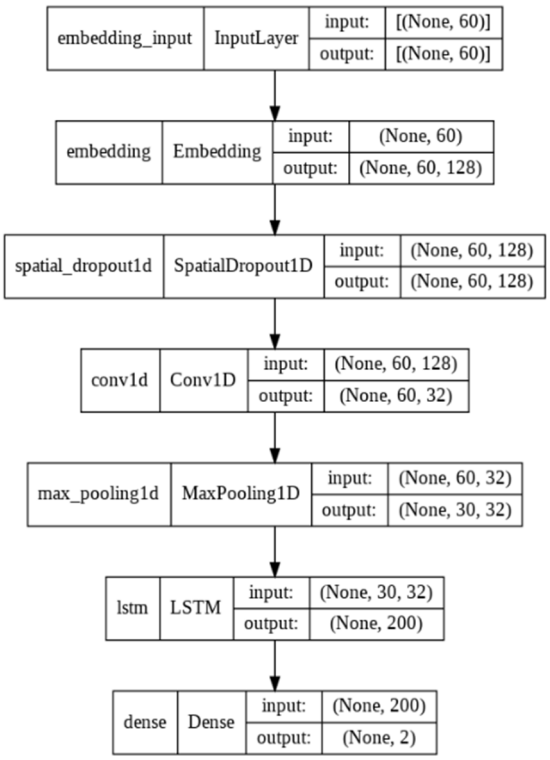


Figure 3. CLSTM model architecture.

4. Experimental Results and Analysis

4.1. Data Analysis Results

We examined the whole pre-processed textual dataset to evaluate the occurrence of suicidal thoughts to assess dissimilarities in the lexicon. We computed the frequencies of all unigrams in both suicidal and non-suicidal tweets. The top 200 unigrams from each category were chosen using Python's WordCloud visualization package to investigate their nature and relationship with suicidal thoughts. Figure 4 illustrates a WordCloud representation of the top 200 unigrams derived from the dataset, divided into two categories: suicidal and non-suicidal tweets.

Figure 4. WordCloud of suicidal and non-suicidal tweets.

The WordCloud of suicidal class shows that suicidal intent tweets include terms such as "fuck", "shit", "hate", "pain", "I'm tired," and "worthlessness", as well as negation phrases such as "don't want", "never", and "nothing". We then discovered that terms with death implications also represent the user's suicidal intentions ('death', 'want die', 'kill'). In contrast to the suicidal postings, the unigrams evaluated in the non-suicidal posts primarily include words expressing happy moments, positive attitudes, and emotions ("I'm happy", "want fun", "laugh loud", "beautiful feel"). Moreover, users are more likely to seek to maintain a positive perspective ("get better") or engage in social activities ("job", "work").

4.2. Classifiers Performance Analysis

Following the n-grams frequency analysis, we examined the experimental technique for detecting suicide thoughts using ML and DL models. We employed CountVectorizer feature extraction on ML models such as MNB, LR, SGD, RF, and SVC. Furthermore, we used word embedding on DL models such as LSTM, BiLSTM, BiGRU, and CLSTM. Then we used evaluation matrices to compare the performance of both the ML and DL models after training. Finally, we performed a comparative analysis of the performance of the classification model. Table 2 shows the classification results for all classifiers based on the evaluation matrices.

Table 2. Performance Table for Classifiers.

Method		Evaluation Metrics				
		Accuracy	**Precision**	**Recall**	**F1-Score**	**AUC**
DL	BiLSTM	93.6%	0.93	0.93	0.93	0.93
	LSTM	93.5%	0.93	0.93	0.93	0.93
	BiGRU	93.4%	0.93	0.93	0.93	0.93
	CLSTM	93.2%	0.91	0.93	0.93	0.93
ML	RF	93.0%	0.92	0.92	0.92	0.92
	SVC	91.9%	0.91	0.91	0.91	0.91
	SGD	91.7%	0.91	0.91	0.91	0.91
	LR	91.2%	0.91	0.91	0.91	0.91
	MNB	84.6%	0.84	0.84	0.84	0.84

Despite its extensive applicability, accuracy is not always the best performance statistic to use, particularly when the target variable classes in the dataset are imbalanced. Consequently, we employed the F1 score, which takes precision and recall into account when calculating an algorithm's efficiency. The classification scores for each ML and DL algorithm are shown in Figure 5. When comparing the performance of ML models, we found that RF exceeds other traditional ML approaches with an accuracy score of 93% and a 0.92 F1-score. The SVC, SGD, and LR classifiers' performance was slightly lower than RF's, where the obtained accuracy score was between the range of 91.2% and 91.9%, with an F1 score of 0.91. Though previous research [28,29] has shown that MNB performs well for the task of text classification, it fared the lowest in our study, with an accuracy of 84.6%. Comparing the performance of the DL classifiers, it can be seen that all of the models performed equally well in our experiment, with an accuracy score of 93.2%. With a performance gain of 93.6% accuracy and a 0.93 F1-score, the BiLSTM model outperformed other DL models. In our experiment, the LSTM and BiGRU models attained similar accuracy and an F1 score of 93.4% and 0.93, respectively. Lastly, the LSTM model performed the worst among the DL classifiers, with an accuracy of 93.2%. By observing the performance of both the ML and DL classifiers, it can be seen that the DL classifiers provide better results for the task of identifying suicidal tweets.

The ROC Curve, also known as the ROC-AUC, is used for binary classification, which shows the trade-off between sensitivity and specificity. To create the ROC curve, we must first compute the True Positive Rate (TPR) and False Positive Rate (FPR) for various thresholds. The FPR and TPR values are plotted in the x-axis and y-axis, respectively, for each threshold. As a starting point, a random classifier is supposed to provide points

along the diagonal where FPR is equal to TPR. The further the curve gets to the ROC space's 45-degree diagonal, the more precise the test is. Figure 6 displays the area under the AUC-ROC curve for all ML models, indicating that the RF model has an AUC close to 1. The AUC of RF suggests it is better than other ML models in distinguishing between suicidal and non-suicidal classes.

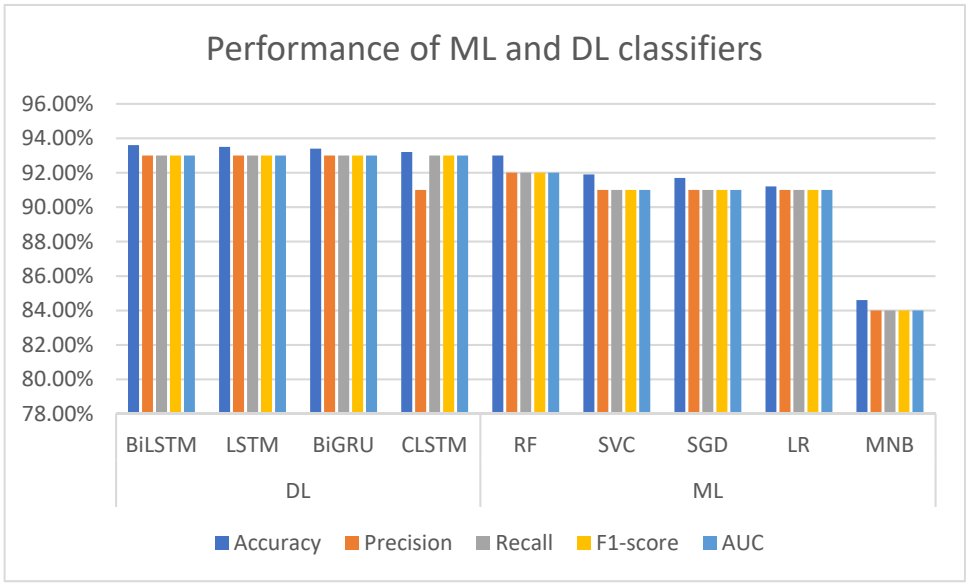


Figure 5. Accuracy score for ML models.

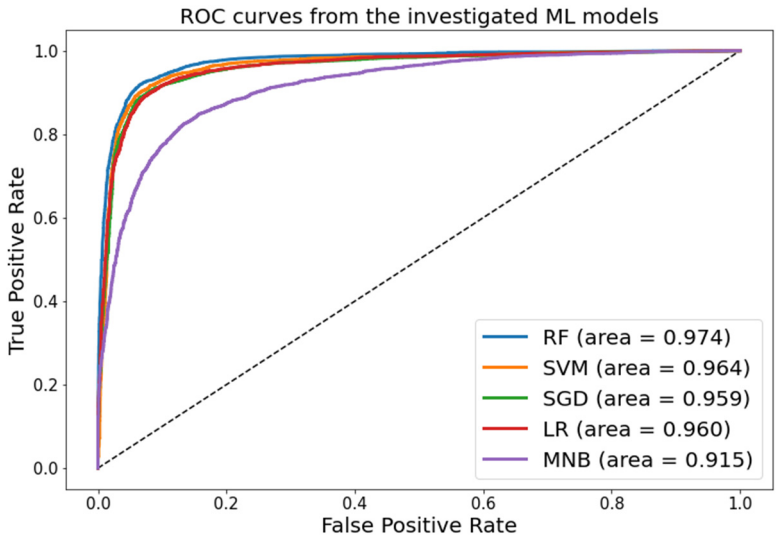
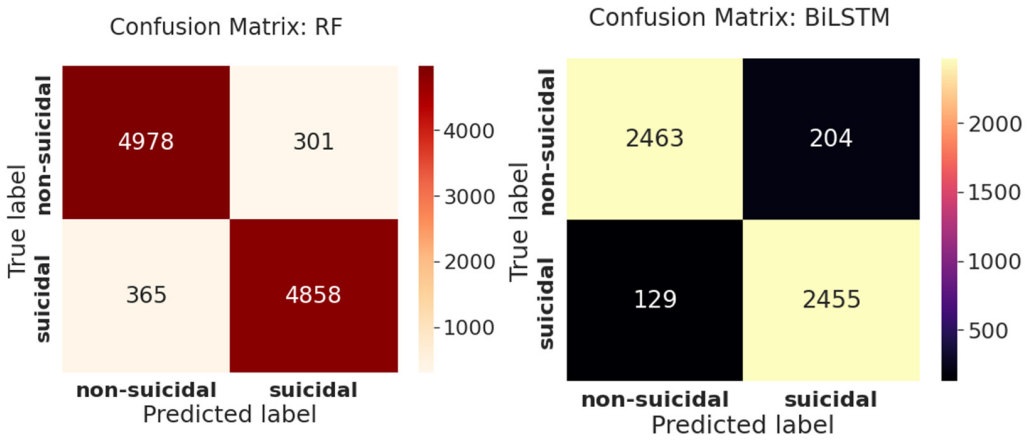


Figure 6. AUC ROC curve for each ML model.

One of the most informative and straightforward methods for evaluating the accuracy and completeness of an ML algorithm is the confusion matrix. Its principal use is in classification tasks where the output might comprise two or more forms of classes. From Figure 7a, we can see on the testing data of 9836 tweets, the confusion matrix of RF demonstrates that the model can predict non-suicidal tweets better than suicidal tweets, as TP (94.0%) is greater than TN (92.5%). Furthermore, the FN is 7.5%, indicating that the model incorrectly forecasts suicidal tweets as non-suicidal, making it challenging to identify the suicidal class. The confusion matrix of BiLSTM, shown in Figure 7b, reveals that the model can predict suicidal tweets better than non-suicidal tweets on a test dataset of 4918 tweets as TN (94.7%) seems to be higher than TP. The model's FN (5.3%) is lower than the FP (8.3%), indicating that it is less likely to misclassify suicidal tweets as non-suicidal.



(**a**) (**b**)

Figure 7. Confusion Matrix of best performed model of ML and DL (**a**) Random Forest; (**b**) BiLSTM.

4.3. Model Validation

To better understand the learning mechanism, we used k-fold cross-validation to determine the mean accuracy in our model. Cross-validation is one of the most extensively used data resampling strategies for assessing the generalization capabilities of predictive models and estimating the actual estimation error of models. The learning set is divided into k disjoint subgroups of roughly equal length in k-fold cross-validation. The number of subgroups produced is referred to as "fold." This partition is accomplished by randomly picking examples from the learning set without replacing them. Our ML models were trained using k = 10 subsets representing the training set as a whole. The model is then applied to the remaining subset, known as the validation set, and its performance is evaluated. This approach is continued until all k subsets have served as validation sets. Figure 8 demonstrates the accuracy of ML models for each fold.

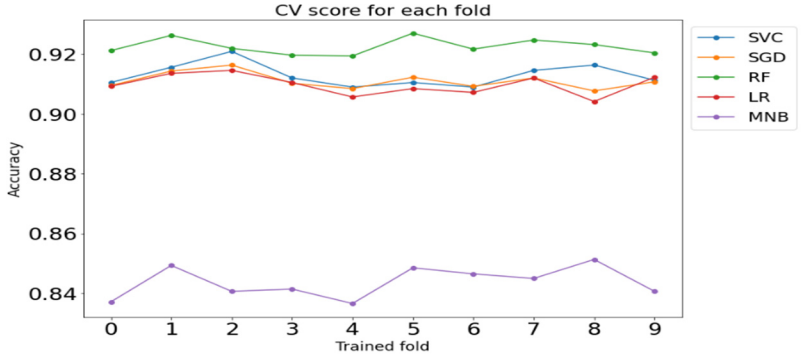
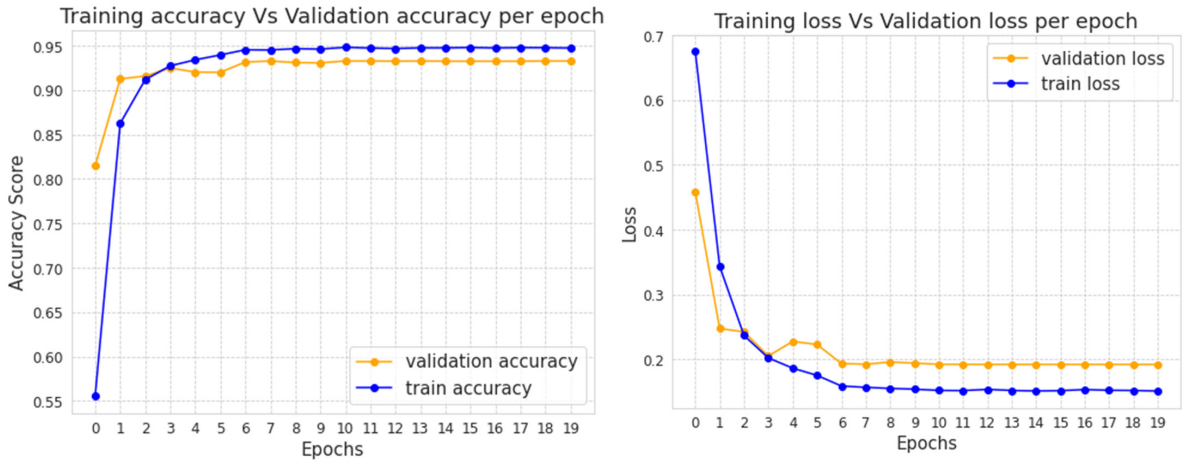


Figure 8. 10-fold Cross-Validation for each ML model.

The validation accuracy and loss for each epoch of the BiLSTM model are shown in Figure 9a,b, respectively. It shows that when the number of epochs increases, our BiLSTM models' validation accuracy and loss tend to increase and decrease, respectively. The 11th epoch model is kept for future testing on the test dataset since we employed a model checkpoint to monitor validation accuracy and saved the best model. We also used ReduceLROnPlateau to lower the learning rate when the model starts overfitting. The validation accuracy and loss appear to be constant after the 12th epoch.



(**a**) (**b**)

Figure 9. BiLSTM models training vs. validation performance comparison. (**a**) Training and validation accuracy per epoch; (**b**) Training and validation loss per epoch.

5. Discussion

It is generally accepted that for a better learning text classifier it needs to have a growing amount of contextual information. The BiLSTM processing chain replicates the LSTM processing chain, allowing inputs to be processed in both forward and backward time sequences. BiLSTM extends the unidirectional LSTM by allowing hidden-to-hidden interconnections to propagate in the opposite temporal sequence by adding a second hidden layer. Therefore, the model can exploit information from both the past and the future. It is advantageous for a model to have knowledge of both the past and future contexts for sentiment classification problems. The approach allows BiLSTM to consider the future context. At the same time, its layer learns bidirectional long-term dependency between time steps in time series or sequence data without maintaining duplicate context information. These dependencies are crucial when we want the network to learn from the entire time series at each time step while also having access to contextual information. Therefore, it demonstrated an excellent performance for our research. The BiLSTM model does, however, have the disadvantage of requiring more training data and effort than the other classifiers.

It is worth noting that most studies only provide unclear information on pre-processing procedures, which are an essential part of text classification. One of the main reasons for our high accuracy score was by using several NLP techniques to pre-process the tweets effectively. In addition, both the ML and DL models were trained with appropriate parameters to minimize overfitting. Models often appear to perform poorly in the unbalanced class due to the problem of class imbalance. We did not encounter the class imbalance issue since we conducted our experiment with two evenly split classes; as a result, all of the evaluation matrices performed equally well. Even though our experimental results indicate that evaluated matrices work relatively well, cross-validation was not completed on the DL classifiers, which would have resulted in an extremely time and resource-consuming setup. In addition, we ran the experiment on a single dataset of 49,178 tweets. A Tweet's length restriction is 280 characters, often insufficient to comprehend a person's suicidal intentions. If we had gathered more postings from other social media platforms, we could have detected a difference in classifier performance. Furthermore, the psychology of suicide attempts is complicated. Our models can extract statistical indications from suicidal tweets, but they cannot reason about risk variables by adding suicide psychology.

6. Conclusions

Early detection of suicidal thoughts is a crucial and effective method of preventing suicide. The majority of work on this topic has been completed by psychologists using

statistical analysis, while computer scientists have used feature engineering-based ML- and DL-based representation learning. Detection of early suicidal intentions on microblogging sites such as Twitter will help medical experts identify and save many lives. The DL and ML approaches can offer new opportunities for improving suicidal ideation detection and early suicide prevention.

In this study, we attempted to compare and analyze several ML and DL models for detecting the presence of suicidal ideation signs in user tweets. The main purpose of the study was to find out the best performing model that can identify suicidal ideations of Twitter users with great accuracy. There are some publicly available datasets on several subreddits or suicide forums for the task of suicidal ideation, but there is no ground truth dataset for analyzing online tweets. As a result, we created the experimental dataset from live tweets by users using suicide-indicative and non-suicidal keywords, then pre-processed the text using different NLP approaches to train on ML and DL algorithms. Five ML algorithms were trained using CountVectorizer feature extractor and four DL models were trained with word embedding technique. Our experimental results show that the BiLSTM model performed best in training, validation, and testing. The model surpasses the other ML and DL models of our experiment with an accuracy of 93.6%. The reason behind the superior performance of BiLSTM is because it can extract relevant information from lengthy tweets more effectively by dealing with forward–backward dependencies from feature sequences resolving gradient disappearance and long-term dependence.

It is important to mention that the ML and DL classifiers were trained with CountVectorizer and word embeddings. Utilization of other feature extraction techniques such as Word2Vec, GloVe, Bag-of-words, and TF-IDF could have resulted in better classification scores. The experiments were carried out on a balanced dataset which implies that all the evaluation metrices performed equally well. On an imbalanced collection of datasets, the model would struggle to perform well on minority class. The problem can be solved with the use of a hierarchical ensemble model. Moreover, due to the lack of ground truth datasets, we have only focused on binary classification on a single dataset in our experiment. However, binary classification is not enough to comprehend the actual sentiments of the users. In the future, we will provide a better comparative analysis between the performances of DL and transfer learning classifiers on different word embedding techniques such as Word2Vec, GloVe, and FastText on several multi-class suicide related datasets. As the transfer learning algorithms can outperform DL classifiers for the task of text classification, the results can be further improved by hyperparameter fine tuning. Future work should reapply the same ML and DL approaches in other disease diagnoses such as heart disease [30–32], and COVID-19 [33–36] with explainable AI, which would also incorporate numerical, categorical, and text data.

It is important to develop a real-world web application that incorporates the classifiers that mental health professionals can utilize to identify online texts having suicidal thoughts in the hopes of preventing suicide. In the future, we also plan to improve the model's performance and produce a practical online application for clinical psychologists and healthcare practitioners.

Author Contributions: Conceptualization, R.H. and N.I.; methodology, R.H. and N.I.; Software, R.H. and N.I.; validation, M.I. and M.M.A.; writing—original draft preparation, R.H. and N.I.; writing—review and editing, R.H., N.I., M.I. and M.M.A.; formal analysis, M.M.A. and M.I. 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.

References

1. Suicide. Available online: https://www.who.int/news-room/fact-sheets/detail/suicide (accessed on 27 September 2021).
2. Ivey-Stephenson, A.Z.; Demissie, Z.; Crosby, A.E.; Stone, D.M.; Gaylor, E.; Wilkins, N.; Lowry, R.; Brown, M. Suicidal Ideation and Behaviors Among High School Students—Youth Risk Behavior Survey, United States, 2019. *MMWR Suppl.* **2020**, *69*, 47–55. [CrossRef] [PubMed]
3. Gliatto, M.F.; Rai, A.K. Evaluation and Treatment of Patients with Suicidal Ideation. *Am. Fam. Physician* **1999**, *59*, 1500. [PubMed]
4. Giachanou, A.; Crestani, F. Like it or not: A survey of Twitter sentiment analysis methods. *ACM Comput. Surv.* **2016**, *49*, 1–41. [CrossRef]
5. Oussous, A.; Benjelloun, F.-Z.; Lahcen, A.A.; Belfkih, S. ASA: A framework for Arabic sentiment analysis. *J. Inf. Sci.* **2019**, *46*, 544–559. [CrossRef]
6. Pachouly, S.J.; Raut, G.; Bute, K.; Tambe, R.; Bhavsar, S.; Students, U. Depression Detection on Social Media Network (Twitter) using Sentiment Analysis. *Int. Res. J. Eng. Technol.* **2021**, *8*, 1834–1839. Available online: www.irjet.net (accessed on 23 April 2022).
7. Machine Classification for Suicide Ideation Detection on Twitter. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 4154–4160. [CrossRef]
8. Stankevich, M.; Latyshev, A.; Kuminskaya, E.; Smirnov, I.; Grigoriev, O. Depression detection from social media texts. *CEUR Workshop Proc.* **2019**, *6*, 2523.
9. Abdulsalam, A.; Alhothali, A. Suicidal Ideation Detection on Social Media: A Review of Machine Learning Methods. 2022. Available online: http://arxiv.org/abs/2201.10515 (accessed on 23 April 2022).
10. Aladag, A.E.; Muderrisoglu, S.; Akbas, N.B.; Zahmacioglu, O.; Bingol, H.O. Detecting suicidal ideation on forums: Proof-of-concept study. *J. Med. Internet Res.* **2018**, *20*, e215. [CrossRef]
11. Shah, F.M.; Haque, F.; Un Nur, R.; Al Jahan, S.; Mamud, Z. A Hybridized Feature Extraction Approach to Suicidal Ideation Detection from Social Media Post. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 5–7 June 2020; pp. 985–988. [CrossRef]
12. Ji, S.; Li, X.; Huang, Z.; Cambria, E. Suicidal ideation and mental disorder detection with attentive relation networks. *arXiv* **2020**, arXiv:2004.07601. [CrossRef]
13. Tweepy. Available online: https://www.tweepy.org/ (accessed on 9 December 2021).
14. Mäntylä, M.V.; Graziotin, D.; Kuutila, M. The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. *Comput. Sci. Rev.* **2018**, *27*, 16–32. [CrossRef]
15. Lumontod, R.Z., III. Seeing the invisible: Extracting signs of depression and suicidal ideation from college students' writing using LIWC a computerized text analysis. *Int. J. Res. Stud. Educ.* **2020**, *9*, 31–44. [CrossRef]
16. Masuda, N.; Kurahashi, I.; Onari, H. Suicide Ideation of Individuals in Online Social Networks. *PLoS ONE* **2013**, *8*, e62262. [CrossRef]
17. Pestian, J.; Nasrallah, H.; Matykiewicz, P.; Bennett, A.; Leenaars, A. Suicide Note Classification Using Natural Language Processing: A Content Analysis. *Biomed. Inform. Insights* **2010**, *3*, BII.S4706. [CrossRef] [PubMed]
18. Tadesse, M.M.; Lin, H.; Xu, B.; Yang, L. Detection of Depression-Related Posts in Reddit Social Media Forum. *IEEE Access* **2019**, *7*, 44883–44893. [CrossRef]
19. Sawhney, R.; Manchanda, P.; Mathur, P.; Shah, R.; Singh, R. Exploring and Learning Suicidal Ideation Connotations on Social Media with Deep Learning. In Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Brussels, Belgium, 31 October 2018; pp. 167–175. [CrossRef]
20. Ji, S.; Yu, C.P.; Fung, S.-F.; Pan, S.; Long, G. Supervised Learning for Suicidal Ideation Detection in Online User Content. *Complexity* **2018**, *2018*, 6157249. [CrossRef]
21. Tadesse, M.M.; Lin, H.; Xu, B.; Yang, L. Detection of suicide ideation in social media forums using deep learning. *Algorithms* **2020**, *13*, 7. [CrossRef]
22. Abboute, A.; Boudjeriou, Y.; Entringer, G.; Azé, J.; Bringay, S.; Poncelet, P. Mining Twitter for suicide prevention. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*; Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2014; Volume 8455, pp. 250–253. [CrossRef]
23. Colombo, G.B.; Burnap, P.; Hodorog, A.; Scourfield, J. Analysing the connectivity and communication of suicidal users on twitter. *Comput. Commun.* **2016**, *73*, 291–300. [CrossRef]
24. Hswen, Y.; Naslund, J.A.; Brownstein, J.S.; Hawkins, J.B. Monitoring Online Discussions About Suicide Among Twitter Users With Schizophrenia: Exploratory Study. *JMIR Mental Health* **2018**, *5*, e11483. [CrossRef]
25. Dara, S.; Tumma, P. Feature Extraction by Using Deep Learning: A Survey. In Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018, Coimbatore, India, 29–31 March 2018; pp. 1795–1801. [CrossRef]
26. Lu, Y.; Salem, F.M. Simplified gating in long short-term memory (LSTM) recurrent neural networks. *Midwest Symp. Circuits Syst.* **2017**, 1601–1604. [CrossRef]
27. ArunKumar, K.E.; Kalaga, D.V.; Kumar, C.M.S.; Kawaji, M.; Brenza, T.M. Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends. *Alex. Eng. J.* **2022**, *61*, 7585–7603. [CrossRef]

28. Singh, G.; Kumar, B.; Gaur, L.; Tyagi, A. Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification. In Proceedings of the 2019 International Conference on Automation, Computational and Technology Management, ICACTM 2019, London, UK, 24–26 April 2019; pp. 593–596. [CrossRef]
29. Zhao, L.; Huang, M.; Yao, Z.; Su, R.; Jiang, Y.; Zhu, X. Semi-supervised multinomial naive bayes for text classification by leveraging word-level statistical constraint. *Proc. AAAI Conf. Artif. Intell.* **2016**, *30*, 2877–2884.
30. Ahsan, M.M.; Mahmud, M.A.; Saha, P.K.; Gupta, K.D.; Siddique, Z. Effect of data scaling methods on machine learning algorithms and model performance. *Technologies* **2021**, *9*, 52. [CrossRef]
31. Ahsan, M.M.; Siddique, Z. Machine learning-based heart disease diagnosis: A systematic literature review. *Artif. Intell. Med.* **2022**, *128*, 102289. [CrossRef]
32. Ahsan, M.M.; Luna, S.A.; Siddique, Z. Machine-Learning-Based Disease Diagnosis: A Comprehensive Review. *Healthcare* **2022**, *10*, 541. [CrossRef] [PubMed]
33. Ahsan, M.M.; Gupta, K.D.; Islam, M.M.; Sen, S.; Rahman, M.; Shakhawat Hossain, M. COVID-19 symptoms detection based on nasnetmobile with explainable ai using various imaging modalities. *Mach. Learn. Knowl. Extr.* **2020**, *2*, 490–504. [CrossRef]
34. Ahsan, M.M.; EAlam, T.; Trafalis, T.; Huebner, P. Deep MLP-CNN model using mixed-data to distinguish between COVID-19 and Non-COVID-19 patients. *Symmetry* **2020**, *12*, 1526. [CrossRef]
35. Ahsan, M.M.; Ahad, M.T.; Soma, F.A.; Paul, S.; Chowdhury, A.; Luna, S.A.; Yazdan, M.M.S.; Rahman, A.; Siddique, Z.; Huebner, P. Detecting SARS-CoV-2 from chest X-Ray using artificial intelligence. *IEEE Access* **2021**, *9*, 35501–35513. [CrossRef]
36. Ahsan, M.; Nazim, R.; Siddique, Z.; Huebner, P. Detection of COVID-19 Patients from CT Scan and Chest X-ray Data Using Modified *MobileNetV2* and *LIME*. *Healthcare* **2021**, *9*, 1099. [CrossRef]