5 Review Analysis of Ride-Sharing Applications Using Machine Learning Approaches

Bangladesh Perspective

*Taminul Islam[1], Arindom Kundu[1], Rishalatun Jannat Lima[1], Most Hasna Hena[1], Omar Sharif[*2], Azizur Rahman[3] and Md Zobaer Hasan[4]*
[1]Department of Computer Science & Engineering, Daffodil International University, Ashulia, Bangladesh
[2]Universal College Bangladesh (Monash College), Dhaka, Bangladesh
[3]School of Computing, and Mathematics and Engineering, Charles Sturt University, Wagga Wagga, Australia
[4]School of Science, Monash University Malaysia, Selangor D. E., Malaysia
*Corresponding Author: omar.sharif@monashcollege.edu.au

CONTENTS

DOI: 10.1201/9781003253051-7

5.1 INTRODUCTION

There are several ride-sharing applications available in Bangladesh like Uber, Patho, Obhai, Grab, etc. The quality of each company depends on their providing services. Users are able to submit their reviews of services using number of stars as well as comments. The ranking and popularity of a ride-sharing app are determined by the reviews left by its users. Reviews help future passengers better evalute ride-sharing apps, while the authenticity of those evaluations is unknown and could have varied effects. The use of machine learning techniques can make it easier to spot fake ride-sharing app reviews. Web mining techniques (Sharma et al., 2022) employ a variety of machine learning algorithms to locate and collect specific data from the internet. Content mining is used to gather reviews (Lai et al., 2021), and involves using machine learning to train a classifier to assess review attributes and user sentiments for determining the user overall experience (positive or negative) (Agarwal, 2022c; Agarwal et al., 2016). Fake reviews are usually detected by looking at specific factors that are not directly related to the content of ride-sharing, such as the category in which the review appears. Some people manipulate reviews to disseminate false information (Agarwal et al., 2022a). False information can be used to boost or degrade a company or application, depending on the intent. Fake reviews, review spams, and opinion spams are all terms used to describe this type of activity. In accordance with Rausch et al. (2022), creating a false review is a form of opinion spamming. Instead of expressing their true ideas or experiences, reviewers attempt to deceive readers or automated opinion mining and sentiment analysis algorithms (Hossain et al., 2021; Kumari et al., 2021), which is considered as an unlawful behaviour (Rausch et al., 2022).

A fake review is one in which the reviewer knowingly provides untruthful or irrelevant information regarding the review item, whether it is partly false or completely false. Aside from being dubbed fake reviews, other terms for them include bogus, scamming, misleading, and spam (Chowdhury et al., 2018a). Spammers may intend to build excitement for a product or service by generating good reviews in large numbers. This is the fundamental concern with review spams (Chowdhury et al., 2018b). False reviews now have a significant impact on how customers perceive a brand (Krishna Rao et al., 2022). For organizations, positive reviews can result in large financial gains; on the other hand, poor evaluations can quickly destroy a company's good name. Automated systems or paid reviewers can create reviews. Fake positive evaluations for a company's products or services can be written by people or third-party groups hired by companies or merchants. Since anybody can simply create and submit a review on the internet, the practice of spamming ride-sharing applications with fake reviews has become more common. Therefore, we utilize some machine learning approaches such as Decision Tree, Random Forest,

TABLE 5.1
Research Question Criteria

Criteria	Details
Population	Bangladeshi ride-sharing application users
Mediation	Machine learning and deep learning approaches to prediction
Outcome	Important attributes, accuracy, and classification
Context	Ride-sharing application's review section

Gradient Boosting, AdaBoost and Bidirectional Long Short-Term Memory (Bi-LSTM) (Rahman, 2019), and Bi-LSTM to achieving an optimal level of accuracy of reviews. A suitable research question is vital for uncovering related works in machine learning and approaches for ride-sharing applications data analysis. Kitchenham et al. (2010) outlines the steps necessary to answer the appropriate research questions, such as population, intervention, outcomes, and context. Table 5.1 shows the research topic criteria.

According to the review findings, the following research questions should be pursued:

Q1: How can we fetch reviews from ride-sharing applications?
Q2: What are the approaches to finding real reviews?
Q3: What are the approaches to pre-processing data?
Q4: What is the market value of ride-sharing applications?
Q5: What machine learning approaches are used for review analysis?
Q6: What is the performance of the present proposed models?

This chapter is organized as follows. A background analysis is found under the "Related Work" section. In the methodology section, the essential modelling idea is thoroughly explained. Figures and tables are used in the process to illustrate the suggested thought. All the models' performances are evaluated in great depth in the Results section. This study's Discussion and Conclusion sections discuss the results of this study and where the research is headed.

5.2 RELATED WORK

The research community in the domain of natural language processing with machine learning and deep learning has grown significantly (Agarwal, 2022c; Agarwal, 2019). More than 15 million evaluations from more than 3.5 million users from three major travel sites were included in a study by Minnich et al. (2015). There were three main goals in their work. They developed brand-new tools for detecting disparities across many sites. They also carried out the first comprehensive research of cross-site variations using real data and produced a data-science-based technique with 93% accuracy. The TrueView score was then presented, and 20% of hotels appeared to have a low trustworthiness score, based on the results. Moreover, the study by

Heydari et al. (2015) analyzes various modelling tools to categorize them according to models that mostly identify spam in reviews. It is important to note that each sort of detection method has various strengths and drawbacks. Although the above study method has continued to improve in terms of accuracy and output, it's main limitation is that they can't achieve more precise results without going through the process of systematic analysis (Crawford et al., 2016). Most of these studies have one thing in common: they turn reviews into word vectors, which can provide tens of thousands of unique characteristics. However, little research has been done on how to appropriately reduce the size of the feature subset to a tolerable quantity. Filter-based element rankers and term feature selection were applied by researchers to lower the size of a feature subset. These approaches are used in the review spam domain.

These results illustrate that there is no one-size fits-all method to feature selection. Also, the optimum technique to minimize the size of the feature subset depends on the classifier employed and the intended size of the feature subset. Researchers have used Decision tree, Logistic Regression, Naïve Bayes, SVM, and Multinomial Naïve Bayes to determine the accuracy of their proposed model (Shiraz et al., 2017; Thevaraja & Rahman, 2019), and many found decision trees to be most accurate at about 83%. Review spam detection is no different in that finding labelled datasets is always a difficulty for machine learning researchers. Using Amazon Mechanical Turk (AMT) to produce fake reviews for their dataset and combining them with "true" TripAdvisor ratings, Ott et al. (2011) developed a unique technique. To come up with their final dataset, they gathered a total of 400 false and 400 true reviews. These classifiers were tested on a variety of different datasets, including unigrams, bigrams, and trigrams. There was no statistical analysis done to see if the difference between SVM and bigrams in terms of performance was significant since the dataset was rather small.

Some published fake news detecting works are summarized in Table 5.2.

5.3 METHODOLOGY

This section introduces the main methodology of this study. An outline of this research process is shown in Figure 5.1 This section explores the dataset's source and features. In addition, contextual aspects are addressed here. Some classification models and evaluation procedures are briefly explored in the later portion of this chapter. The steps in our study process are as follows:

Dataset Creation: The dataset was created from the review sections of Bangladeshi ride-sharing applications: Uber, Pathao, Obhai, and Shohoz.

Pre-processing: To deal with noisy and inconsistent data, pre-processing techniques are employed (Rahman et al., 2013; Rahman & Harding, 2016; Rahman, 2017; Rahman, 2019). Many different pre-processing procedures are used to improve the quality of the final product of the data. Main techniques that were applied include tokenization, lemmatization, Punctuation removal, Stopwards removal and others.

TABLE 5.2
Summary of Related Research Work

References	Year	Contribution	Dataset	Models	Accuracy
(Vachane and D. 2021)	2021	Incorporated spam identification architecture that uses display review illuminating lists as metadata structures.	HIN Resource	Naïve Bayes, Decision Tree	Decision Tree achieved the best 92.06% accuracy
(Manaskasemsak et al., 2021)	2021	Developed graph model to detect fake reviews and fake reviewers.	Yelp	CNN	75%
(Yao et al., 2021)	2021	Developed model to detect fake reviews from hotel and restaurant sections.	Crowdsourcing. Amazon Mechanical Turk. TripAdvisor	RF, AdaBoost, SVM, CNN, LSTM	Random Forest achieved the best 90% accuracy
(Budhi et al., 2021)	2021	Proposed a data sampling technique that improves the accuracy of the fake review class.	Yelp	LR, SVM, Multilayer Perceptron (MLP), Bagging Predictor (BP), RF, AdaBoost	SVM achieved the best 85.74% accuracy
(Wang et al., 2020)	2020	Proposed a method to identify false reviews using multiple feature fusion and collaborative rolling training.	Yelp	Rf, LR, Latent Dirichlet Allocation, KNN, DT, Naïve Bayes, SVM	SVM achieved the best 84.45% accuracy
(Kumar, 2020)	2018	Developed model according to behaviour feature of reviewer to detect fake and true reviews.	Yelp	RF, SVM	Random Forest achieved the best 91.396% accuracy

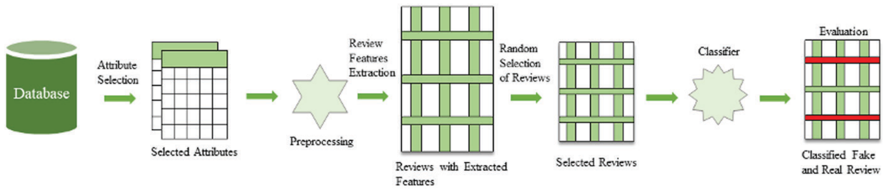

FIGURE 5.1 A step-by-step guide to detecting fake and real reviews.

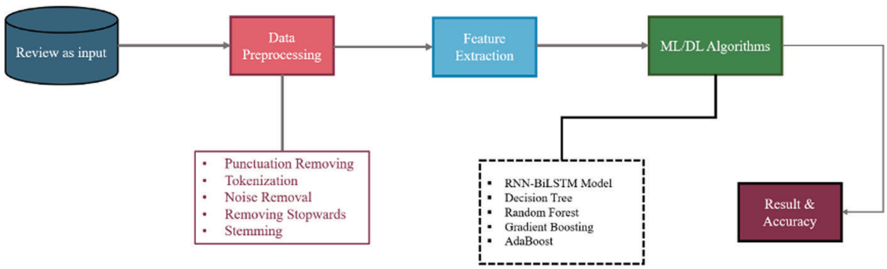

FIGURE 5.2 Proposed model workflow.

Feature Extraction: A feature set for the classification model was built using attributes that were retrieved after pre-processing the data in the review database such as review of quality, service, experience, satisfaction, etc. of hotels.

Training the Model: Several classification algorithms were then trained for experiments associated with our study concerning the accuracy of ride-sharing applications reviews.

Figure 5.2 shows the workflow of our proposed research methodology.

5.3.1 Data Description

Data collection is always a challenging part of a research study. It was a tough task to collect all the fresh data within a short time from all the authorized sites. Data (online user reviews) was collected from four Bangladeshi ride-sharing applications review sections: Uber, Pathao, Obhai, and Shohoz. This data was manually collected from individual websites and social platforms and GooglePlay Store reviews section, resulting in a total of 3315 online reviews from individuals from four platforms. Table 5.3 illustrates the amount of collected data from the individual apps.

Table 5.4 lists the amount of total True reviews, Partially False reviews, and False reviews. The exact number of this statistics is – True (1365), Partially False (930), and False (1020). From Table 5.3 and Table 5.4 it is clearly seen that the data is balanced . This is an important part of this research.

These data were classified into three categories: – True, False, and Partially False. Table 5.5 gives the descriptions of the categories as classified by Balouchzahi et al. (2021).

TABLE 5.3
Amount of Collected Individual Data

Applications	Data Size
Uber	1000
Pathao	985
Obhai	815
Shohoz	515
Total	3315

TABLE 5.4
Amount of Categorical Data

Category	Amount of data
True	1365
False	1020
Partially False	930

TABLE 5.5
Descriptive Category

Category	Description
True	The given text includes contents that are clearly apparent or capable of being logically proven.
Partially False	Main claim in given text might be true but also contain false or misleading information information, not surely true and not certainly false.
False	The main content of given text is fake.

5.3.2 Data Pre-Processing

Pre-processing of data is the initial step in doing research and is the first stage in data mining (Rahman, 2020; Agarwal et al., 2022b). There are several ride-sharing apps platforms from which we obtained our reviews datasets that require preprocessing. These datasets were split down into a wide range of numerical values and processed one at a time since machine learning and deep learning can handle numerical data only. For the text and rating dimensions, we used the following data preparation approaches. We focused on the data pre-processing steps followed by data cleaning, data integration, data transformation, data reduction, and finally data discretization.

Reduce of Dimension: To reduce the size of the data, it is necessary to convert it from a high-dimensional space to a low-dimensional one, while still retaining as many of the original data's attributes as possible (Burges, 2010). Unnecessary features in the data are responsible for increasing the length of time it takes to complete an operation. The public id and title characteristics were deleted from the dataset prior to the data being entered into the model. The text column in the dataset contains the input data, while the target column has the rating.

Punctuation Removal: There are many punctuation marks, links, numbers, and other special characters used in reviews, none of which have any influence on whether the review is true or incorrect in the vast majority of cases. In addition, punctuation appears often and has a substantial influence on the measurements for punctuation, but it has no effect on the classification of the text, which is a mixed bag (Pradha et al., 2019). Figure 5.3 shows an example of punctuation removal.

Noise Removal: Noise removal refers to the process of removing letters, numbers, and fragments of text that might obstruct text analysis. It is a vital step in the preparation of data (Tang et al., 2022). All data must be clear and free of noise. Words that are unnecessary to tokenize and vectorize must be removed from the input sequence. Tokenization is improved by converting uppercase characters to lowercase ones.

Tokenization: Tokenization is the process of separating review material into words (tokens). To calculate the Reviewer Content Similarity (RCS) and capital variety, tokenization is a critical step since it allows each word in the review to be separated (Sockin et al., 2022). For word tokenization, we made use of the NLTK library. For example, for the word "greatest." we used character tokens: g-r-e-a-t-e-s-t and subword tokens: great-est.

Removing Stopwards: Due to the fact that stopwards are widespread in natural language and do not convey any unique meaning, they are not significant in a phrase (Gerlach et al., 2019). Stopwards such as 'is', 'an', 'the', etc., may increase the

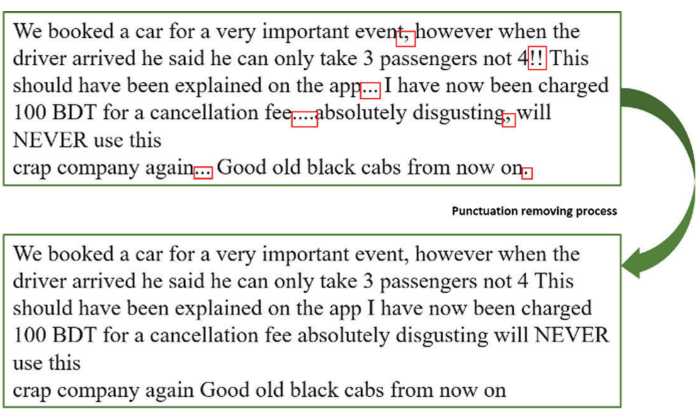

FIGURE 5.3 Example of punctuation removing from the text.

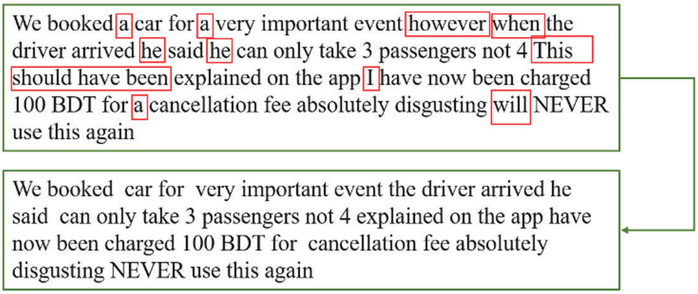

FIGURE 5.4 Example of stopwards removing from the text.

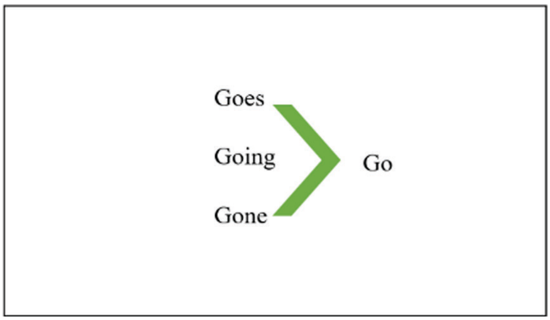

FIGURE 5.5 Example of stemming removal from the text.

amount of time it takes to process data in data analysis. Because of this, it is important to remove stopwords from phrases. We employed the NLTK library for this purpose. We must remove all of the unnecessary text (e.g., stopwards) and strings from the data in order to make it trainable. Because of this, we convert all of our text into numbers so that it may be utilized as a teaching aid. Figure 5.4 is an example of stopwords being removed from a text.

Capitalization: In a computational model, it is ideal to use the same register level regardless of whether upper- or lowercase characters are used (Păiş et al., 2022). It does not matter what kind of register level you use when it comes to digits. Lowercase letters were used in this study.

Stemming Removal: Eliminating suffixes and prefixes from a word is known as stemming. Stemming is a technique that is frequently employed in information retrieval activities. Numerous researchers have demonstrated that stemming increases information retrieval system performance (Atchadé et al., 2022). Using the stemming method, we can get a word back to its root structure. Figure 5.5 shows how stemming is done.

5.3.3 Proposed Model Working Procedure

In this research data was collected from four different ride-sharing applications. Our main goal was to find the best machine learning model to detect fake, true, or partially fake reviews smoothly. Five machine learning algorithms were applied to find the best accuracy on this dataset: Decision tree, Random Forest, Gradient Boosting, AdaBoost, and Bi-LSTM. We got the best accuracy (85%) from the Bi-LSTM model.

To begin the modeling procedure, data were divided into two parts:

- Dataset for Training
- Dataset for Testing

About 80% of the data was utilized for training, and 20% for testing. And this is also what we expected to observe in our model.

Feature Selection and Extraction

The Keras library was used to create our Bi-LSTM model, which was then tested. It is possible to make a model using a glove embedding of 100d. The sequential model was used as the foundation for this experiment's analysis. A number of different techniques are used in the model for accurate feature extraction, including embedding, dropout layers, and a layer with 256 neurons that is totally connected to the rest of the network. This dataset has many classes, which is why soft-max activation was used to apply the output layer to the final layer. It is consistent with other algorithms, such as Random Forest, Gradient Boosting, and AdaBoost. N-gram features such as unigram, bigram, and trigram are employed in all machine learning techniques for improved model outcomes. The model was trained using 20 epochs and 128 batch sizes of training data to achieve optimal performance. The accuracy of this model was determined to be 85%, while the F_1 Score was found to be 89%.

5.3.3.1 Machine Learning Models

In this study, the major aim was to create the best machine learning model to recognize fake, real, or partially fake reviews. On this dataset, we used five different machine learning methods to determine the one with the greatest accuracy: Decision Tree, Random Forest, Gradient Boosting, AdaBoost, and Bi-LSTM models. These models are discussed in this section.

Decision Tree

Classification and regression models may be built at regular intervals using a decision tree. In terms of categorization and predictions, this is the most effective and widely used technology available today. There are many different types of decision trees; the most common is the flowchart-like tree structure, in which each internal node symbolizes a test on a certain characteristic, and each branch reflects a conclusion of the test (Pappalardo et al., 2021). The last word is a node in a tree having nodes for decisions and nodes for leaves. Other nodes are either a few or many branches in the decision tree. Decisions or classifications are represented by a leaf node. The root node of a passing tree, which corresponds to the highest successful predictor, is the simplest decision node in the tree. Decision trees can deal with any type of data, whether it is numerical or categorical (Fletcher et al., 2019).

Random Forest

The choice tree is the basic component of random forest classifications. The choice tree is littered with living trees including a variety of elements at each node. The entropy of a specified collection of characteristics is supported by the nodes. In the random forest, a collection of decision trees is linked to a collection of bootstrap samples derived from the source dataset. Trees are the building blocks of a forest, and the more trees there are, the more stable it will be (Khan et al., 2022). By creating call trees out of data samples, the random forest algorithm receives the forecast for every one of these trees, then votes on which is best. Breiman (2001) include extensive information on random forest classifiers. At times while using the quality random forest strategy, the bootstrapping technique is used to help create an appropriate random forest with the requisite number of decision trees thus boosting classification accuracy using the notion of overlap dilution as described. To train and optimize the process, random forests are often used such as growing trees, making it easier to achieve a decision in each level. As a result, random forest is a good method for numerous packets (Magidi et al., 2021).

Gradient Boosting

Many machine learning methods are combined into Gradient Boosting Classifiers (GBCs) in order to create a strong predictive model. Gradient boosting is a technique in which decision trees are occasionally employed. Gradient boosting models have lately been used to win multiple Kaggle informatics challenges due to their success in categorizing large datasets. The main goal is to lower the amount of error in the next model by aligning the desired outcomes (Bahad et al., 2020). There are many different methods to build gradient boosting classifiers in the Python machine learning, Scikit-Learn. This reference examines the theory underlying gradient boosting models and looks at two distinct techniques to construct gradient boosting models in Scikit-Learn (Chakrabarty et al., 2019).

AdaBoost

Multi-learner approaches to problem-solving are referred to as "ensemble learning" (Liu et al., 2022). When it comes to learning, ensemble techniques are a popular choice because of their superior capacity to generalize. Due to its strong theoretical foundation, precise prediction, tremendous simplicity (Wang noted it required only "only 10 lines" of code), and extensive and successful use cases, the AdaBoost algorithm created by Wang et al. was among the most significant ensemble techniques. Because AdaBoost is the most widely used ensemble algorithms, its huge influence is not surprising. The theoretical and practical aspects of these two topics are briefly discussed in this reference (Wang et al., 2021). Because of AdaBoost, there has been an abundance of theoretical research on ensemble approaches, which is readily available in the machine learning and statistical literature.

Bi-LSTM

In comparison to Long Short-Term Memory, Bi-Directional Long Short-Term Memory (Bi-LSTM) excels at categorizing sequences (LSTM). It is the process of creating a neural network that can process information in both forward and reverse orientations. The Bi-LSTM is composed of two LSTMS, one for forward and one for reverse input. It is feasible to communicate data in both directions using disguised states. Each time step, the outputs of two LSTMs are merged to generate one (Liu

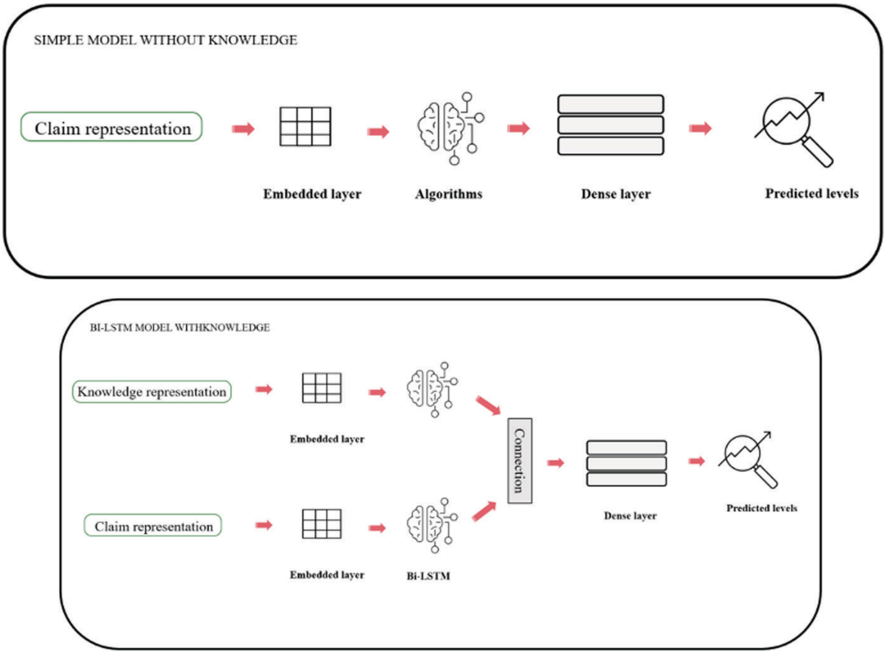

FIGURE 5.6 Working sketch between Simple and Bi-SLTM models.

et al., 2022). The Bi-LSTM technique contributes to the reduction of the restrictions associated with traditional RNNs. The context is more easily comprehended as a result of Bi-LSTM's high degree of accuracy. However, with bi-directional input, we can ensure that both the future and the past are preserve (Braşoveanu et al., 2019). Based on the previous validated reviews, a natural language inference (NLI) model is built using Bi-LSTM neural networks in this phase to assess the validity of each individual claim. We started by training a simple machine learning model with simply the assertions (hypotheses) as input. The NLI-based model is then trained to infer the accuracy of the claim based on previous information (premises). We test the suggested NLI-based strategy to identifying false reviews by comparing the outcomes of these two models. Figure 5.6 shows the process of this method.

A key feature of Bi-LSTM is the ability to learn the forward and backward information contained within the input words. A given input phrase X has N words and is represented as a vector $(x_1, x_2 \ldots, x_n)$. After taking into account prior hidden states h_{t-1} and cell states c_{t-1} the Eq. 5.1 is used to determine the present state:

$$i_t = \sigma\left(W_i w_t + U_i h_{t-1} + b_i\right) f_t = \sigma\left(W_f w_t + U_f h_{t-1} + b_f\right) 9 \quad 99v$$

$$= tanh\left(W_c w_t + U_c h_{t-1} + b_c\right) C_t = f_t C_{t-1} + i_t \tilde{c}_t h_t = o_t tanh\left(c_t\right)$$

$$o_t = \sigma\left(W_o w_t + U_o h_{t-1} + b_o\right) \tag{5.1}$$

The sigmoid function denoted by σ and the hyperbolic tangent function tanh are both used in the equation proposed by Zheng and Chen (2021). The authors provide an algorithmic process for matrix multiplication that stands for elementwise multiplication. There are two sets of weight matrixes: W represents the current input vector w_t and U represents the prior hidden state vector h_{t-1}; both sets of weight matrixes are shown in the above equation. The Bi-LSTM method's pseudocode is given in the following example.

Input: Character embeddings that have been pre-trained X.

Output: The probability distribution P of the input sequence is returned:

(1) The forward LSTM layer receives character vectors from X
(2) **for** i $length(X)$ do
(3) send X_i to Bi-LSTM layer
(4) end **for**
(5) Set 2: The current LSTM network's cell state was updated.
(6) $f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right)$
(7) $i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right)$
(8) $\widetilde{c_t} = tan\ tanh\left(W_f\left[h_{t-1}, x_t\right] + b_c\right)$
(9) $c_t = f_t * c_{t-1} + i_t * c_t$
(10) $o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$
(11) $h_t = o_t * tanh\left(c_t\right)$
(12) Step 3: Send the X character vectors to the reverse LSTM layer and repeat the previous two steps.
(13) Step 4: The hidden layers' forward and backward sequencing are spliced together to produce a sentence-level hidden unit sequence C that is rich in context.
(14) Step 5: The prediction matrix P is obtained once C was delivered via a complete connection layer.
(15) **Return** P.

5.3.3.2 Performance Measurement Unit

Various writers utilized a number of criteria to judge the effectiveness of their models. Despite the fact that the bulk of the research utilized many indicators to measure their efficiency, a low amount is also used a single statistic. In this study Accuracy, Precision, Recall, and F_1- Score is examined for evaluating this research effort. Text data analysis benefits greatly from using these four measurement units.

5.3.3.3 Accuracy

The ratio of correctly predicted items to all possible predictions indicates the accuracy of a model outcomes. Eq. 5.2 defines the accuracy:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.2}$$

5.3.3.4 Precision

The precision of a machine learning model's prediction is an essential performance parameter. Divide the number of correct forecasts by the number of correct positives. Eq. 5.3 defines the precision:

$$Precision = \frac{TP}{TP + FP} \tag{5.3}$$

5.3.3.5 Recall

Recognizing all possible real values is the ability of a detector to reliably discover and identify them. It is defined as the ratio of TP to the sum of TP and FN in Eq. 5.4:

$$Recall = \frac{TP}{TP + FN} \tag{5.4}$$

5.3.3.6 F_1-Score

This is known as the harmonic mean since it relies on both accuracy and memory. Mathematical formulation of a memory retrieval is given in Eq. 5.5:

$$F_1 - Score = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right) \tag{5.5}$$

5.4 RESULT

Here five machine learning algorithms were applied on this fresh dataset. There is a tight comparison between the algorithms. However, the Bi-LSTM achieved the best accuracy, precision, recall, and F_1 score. Bi-LSTM achieved the best 85% accuracy but Random Forest and AdaBoost work well and achieved 83% accuracy. We found 80% accuracy in Gradient Boosting and 79% accuracy on Decision Tree algorithm. Table 5.6 gives the results of the five machine learning algorithms.

In the following we look at the results of the five different algorithms, then evaluate the results.

TABLE 5.6
Result Comparison between Five Machine Learning Algorithms

Algorithms	Accuracy	Precision	Recall	F_1 Score
Decision Tree	0.798	0.691	0.723	0.691
Random Forest	0.834	0.690	0.723	0.693
Gradient Boosting	0.803	0.703	0.718	0.629
AdaBoost	0.830	0.706	0.731	0.702
Bi-LSTM	0.850	0.862	0.910	0.893

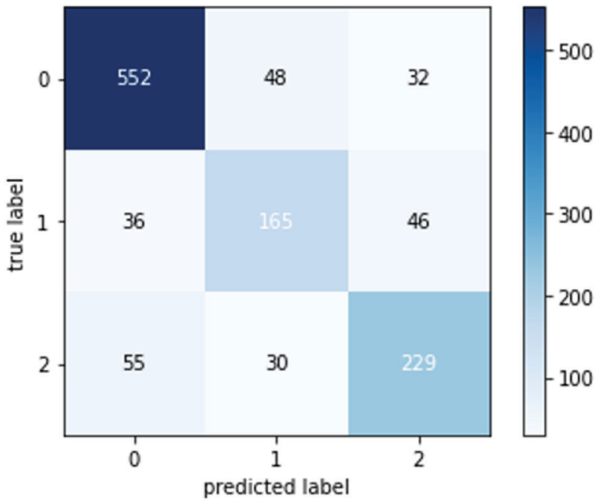

FIGURE 5.7 Confusion matrix of Decision Tree.

Decision Tree

After implementing the decision tree approach to train our model we acquired an accuracy of 79% and a F_1score of 69% where precision was 0.69 and recall 0.72. Figure 5.7 shows the confusion matrix of this algorithm.

Random Forest

After using Random Forest approach to train our model we acquired an accuracy of 83%, which is better than decision tree and a F_1 score of 69%, which is similar with DT, and precision and recall were the same as decision tree. However, the difference on average accuracy was 4%. Figure 5.8 shows the confusion matrix of this algorithm.

Gradient Boosting

In the Gradient Boosting approach after training this model achieved accuracy of 80%, performing relatively better than decision tree and worse than random forest; its F_1score of 62% was not good. The result of precision and recall was 0.70 and 0.71. Figure 5.9 shows the confusion matrix of this algorithm.

AdaBoost

In the AdaBoost approach after training our model we achieved accuracy of 83%, which performs relatively better than DT and GB and it scores F_1score of 70% that performs good. The result of precision and recall was 0.70 and 0.73. Figure 5.10 shows the confusion matrix of this algorithm.

Bi-LSTM

The proposed model's output identifies the review item presented. The review is either true, false, or partially false. The RNN model can't handle text, that's why true

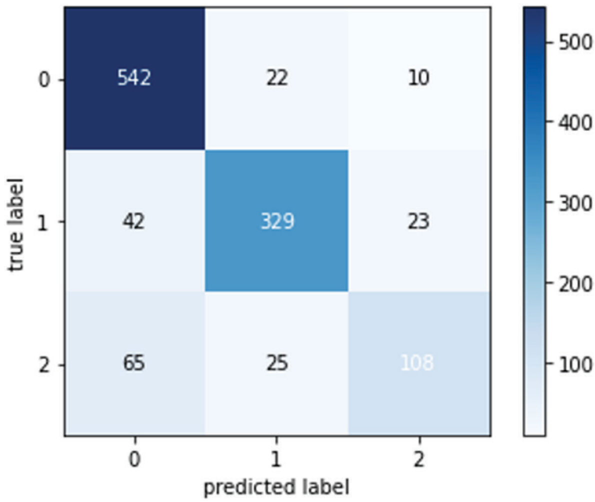

FIGURE 5.8 Confusion matrix of Random Forest.

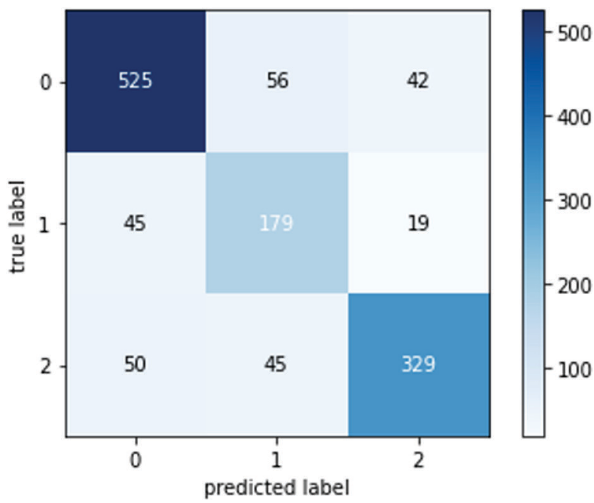

FIGURE 5.9 Confusion matrix of Gradient Boosting.

is considered to be 2, false is 0, and partially false is 1. We applied 20 epochs to train our model and got the best accuracy of 85% and the best F_1score of 89%, where precision and recall was 0.86 and 0.91. We found the best performance using the Bi-LSTM method. Table 5.7 shows the classification report of Bi-LSTM. Figure 5.11 shows the confusion matrix of Bi-LSTM.

Figure 5.12 demonstrates the connection between our proposed model's accuracy and evaluation accuracy and loss and evaluation loss, both of which are achieved using the Bi-LSTM model. These graphics demonstrate that our suggested model

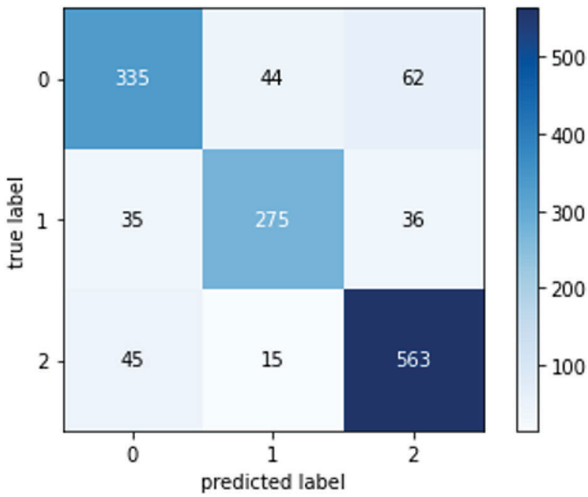

FIGURE 5.10 Confusion matrix of AdaBoost.

TABLE 5.7
Classification Report of Bi-LSTM

Class	Precision	Recall	F_1-Score
True	0.86	0.91	0.89
False	0.91	0.89	0.90
Partially False	0.49	0.37	0.42

accumulates knowledge from its predecessors. From this figure we can see that there are some overfitting issues in the loss vs. evaluation loss graph. While it is not clear the actual reason behind this overfitting, the difference between the learning loss and the evaluation loss increased in later epochs, which may be the reason for this issue, and is a significant limitation of this work.

5.4.1 Cross-Validation

Cross-validation is a term that refers to a variety of similar model validation approaches used to determine the generalizability of the results of a statistical analysis to an independent dataset (Wieczorek et al., 2022). In this work, K-fold cross-validation was used to determine the result. The dataset was divided into k subsets using k-fold cross-validation. The cross-validation process is then used to validate each subset, while the remaining k-1 subsets are merged to utilize as training samples. According to statistical concepts, the best k relies on the number of variables and the type of predictor. K refers to how many groups each data sample should be split into, and it is the only parameter in the technique. k-fold cross-validation is the common

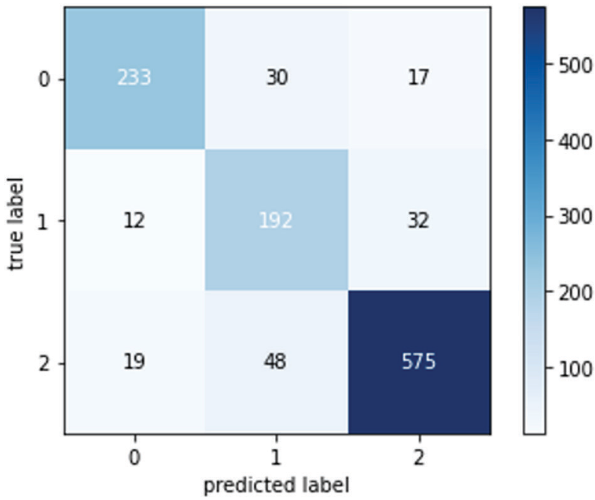

FIGURE 5.11 Confusion Matrix of Bi-LSTM.

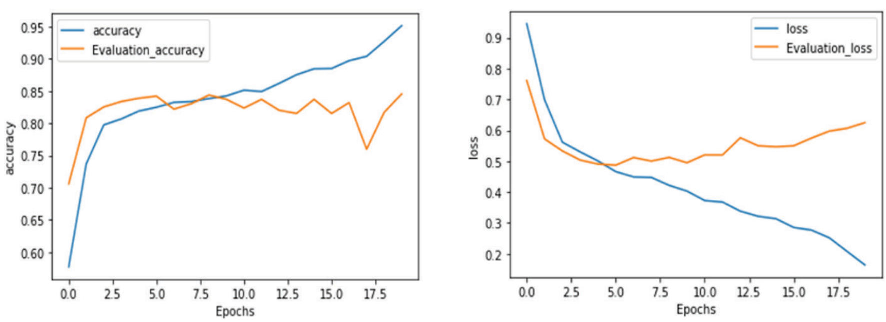

FIGURE 5.12 Graphical representation of accuracy vs evaluation accuracy and loss vs evaluation loss.

name given to this technique (AVUÇLU and E, 2022). It is possible to use a specified value for k in the reference to the model, such as k=10 as 10-fold cross-validation. Table 5.8 clearly shows the k-fold cross-validation of accuracy for each and every model of this work.

In this work, five machine learning algorithms were implemented to find the best model performance. As noted, we got the best model accuracy of 85% from the Bi-LSTM method. To determine this performance, k-fold cross-validation method was used; the detailed results are given in Table 5.8. The five algorithms – Decision Tree, Random Forest, Gradient Boosting, AdaBoost, and Bi-LSTM – performed well in the accuracy cross-validation. We found an 80% 10-fold cross-validation mean score for accuracy from the decision tree whereas we found 84% from the random rorest. On the other hand, gradient boosting performed little bit less accurately than other

TABLE 5.8
K-fold Cross-Validation (for Accuracy) of Five Applied Algorithms

Algorithm	cv=10	cv_score	cv_score (mean)
Decision Tree	1	0.802999	0.803640
	2	0.807806	
	3	0.795142	
	4	0.806100	
	5	0.796545	
	6	0.812104	
	7	0.804949	
	8	0.795977	
	9	0.805268	
	10	0.809513	
Random Forest	1	0.845032	0.847256
	2	0.843038	
	3	0.851093	
	4	0.840588	
	5	0.862186	
	6	0.835555	
	7	0.854597	
	8	0.842178	
	9	0.861001	
	10	0.837284	
Gradient Boosting	1	0.794761	0.794529
	2	0.800492	
	3	0.797103	
	4	0.800222	
	5	0.796060	
	6	0.793273	
	7	0.793011	
	8	0.795018	
	9	0.791043	
	10	0.784308	
AdaBoost	1	0.859676	0.842638
	2	0.839634	
	3	0.864538	
	4	0.828949	
	5	0.854451	
	6	0.823677	
	7	0.827245	
	8	0.864300	
	9	0.823963	
	10	0.839943	

(*continued*)

TABLE 5.8 (Continued)
K-fold Cross-Validation (for Accuracy) of Five Applied Algorithms

Algorithm	cv=10	cv_score	cv_score (mean)
Bi-LSTM	1	0.858471	0.855861
	2	0.857227	
	3	0.854545	
	4	0.856899	
	5	0.860044	
	6	0.859404	
	7	0.854067	
	8	0.850897	
	9	0.855279	
	10	0.851772	

algorithms. We found a 79% 10-fold cross-validation mean score from this algorithm, whereas AdaBoost performed at 84%. Finally, Bi-LSTM performed better than all algorithms and achieved 85% 10-fold cross-validation mean score, which was the highest from all the algorithms. It is clear that an 10-fold cross-validation score is good and shows that the quality of this model for this esearch is good.

5.5 DISCUSSION

Ride-sharing applications are largely influenced by customer and passenger evaluations found online. Passengers who are considering using a service might benefit from reading online reviews before making their final choice of provider. Users' purchasing decisions might be influenced by fake online reviews. On ride-sharing apps, fake reviews are used to promote or degrade services, and can tarnish a good service's reputation, resulting in financial loss for a well-known business. Customers and companies alike are harmed by fake reviews, which are detrimental to both parties. Since 2007, researchers have been focusing on the identification of bogus reviews. Fake reviews, individual spammers, and spammer groups are the focus of the majority of the current research work being done in these fields. The purpose of this research was to determine whether or not some machine learning algorithms could be used to detect bogus reviews. Four Bangladeshi ride-sharing apps were analyzed by our researchers. The classification methods used in this study performed well. Our research has effectively revealed a previously unknown aspect (i.e., machine learning based cross-valiadation of the fake review) by accurately detecting through appropriate classifier.

While all methods behaved similarly, each classifier is notably different from the others, with Bi-LSTM, AdaBoost, and Random Forest being the best and Decision tree and Gradient Boosting being the worst. Among all other algorithms, Bi-LSTM produced the best figure. Figure 5.13 illustrates the sharp difference more clearly. It is critical to notice the confidence interval for the mean accuracy score across all subset sizes, indicating that random forst and AdaBoost are not mutually exclusive.

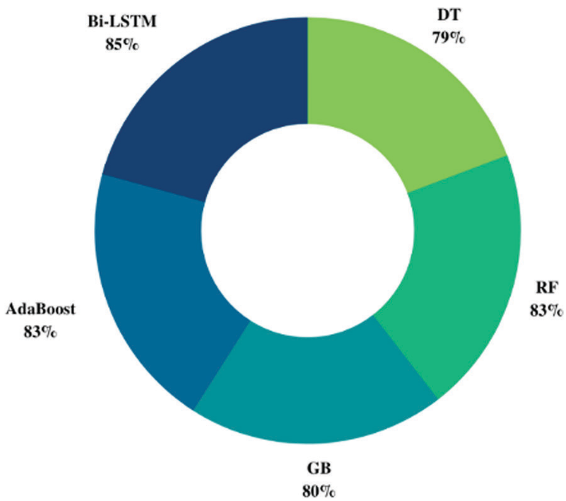

FIGURE 5.13 Accuracy comparison between classifiers.

In this work, data was collected manually by researchers. It was a challenging task to collect quality fresh reviews from different platforms. The result satisfies us in terms of the new dataset. If we work hard to collect the best possible data, then our model should achieve more accuracy in every sector.

5.6 CONCLUSIONS

Today, ride-sharing applications are an important factor of Bangladesh's infrastructure. Fake reviews significantly impair consumers' ability to obtain authentic information. The focus of this research was on identifying fraudulent reviews through the use of well-known machine learning techniques. Data was collected from Uber, Pathao, Shohoz, and Obhai, four of Bangladesh's most popular ride-sharing applications. Following feature extraction and model construction, Bi-LSTM attained the highest accuracy of 85%. Additionally, Random Forest and AdaBoost performed well, with an accuracy of 83%. The model was tested using newly collected data. There are some limitations of this work. Data overfitting is the primary issue of this research. This problem can be solved by using a more appropriate dataset. We could work with a more comprehensive dataset (e.g., big data); this is another limitation of this work. Collecting high-quality data can help enhance accuracy. Performance could be better too. In the future, we hope to test our proposed technique on a bigger, more varied dataset in order to get past existing limitations.

ACKNOWLEDGMENTS

The authors are thankful to the anonymous reviewers for their help in improving this manuscript. We would also like to acknowledge the editing support provided by Charles Sturt University.

REFERENCES

Agarwal, B., Rahman, A., Patnaik, S., & Poonia, R. C. (2022a). Proceedings of International Conference on Intelligent Cyber-Physical Systems. (Algorithms for Intelligent Systems). Springer. www.springer.com/gp/book/9789811671357.

Agarwal, B., Agarwal, A., Harjule, P., & Rahman, A. (2022b). Understanding the intent behind sharing misinformation on social media. Journal of Experimental and Theoretical Artificial Intelligence, 1–15. https://doi.org/10.1080/0952813X.2021.1960637.

Agarwal, B. (2022c) Financial sentiment analysis model utilizing knowledge-base and domain-specific representation. Multimed Tools Appl (2022). https://doi.org/10.1007/s11042-022-12181-y.

Agarwal B., Nayak R., Mittal N., Patnaik S., "Deep Learning Based Approaches for Sentiment Analysis," In Springer. 2015.

Agarwal B., Mittal N., "Prominent Feature Extraction for Sentiment Analysis," In Springer Book Series: Socio-Affective Computing series, Springer International Publishing, ISBN: 978-3-319-25343-5, DOI: 10.1007/978-3-319-25343-5, 2016.

Atchadé, M. N., & Sokadjo, Y. M. (2022). Overview and cross-validation of COVID-19 forecasting univariate models. Alexandria Engineering Journal, 61(4), 3021–3036.

Avuclu, E. (2022). A Statistical Evaluation Between Hybrid, Random, and K-fold Cross Validation to Medical Diagnosis Using Cancerous Breast Tissues.

Bahad, P., & Saxena, P. (2020). Study of adaboost and gradient boosting algorithms for predictive analytics. In International Conference on Intelligent Computing and Smart Communication 2019 (pp. 235–244). Springer, Singapore.

Balouchzahi, F., Shashirekha, H., & Sidorov, G. (2021). MUCIC at CheckThat! 2021: FaDo-fake news detection and domain identification using transformers ensembling. CEUR Workshop Proceedings (CEUR-WS.org). pp. 1–10.

Braşoveanu, A. M., & Andonie, R. (2019). Semantic fake news detection: a machine learning perspective. In International Work-Conference on Artificial Neural Networks (pp. 656–667). Springer, Cham.

Breiman, L. (2001). Random Forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324

Budhi, G. S., Chiong, R., & Wang, Z. (2021). Resampling imbalanced data to detect fake reviews using machine learning classifiers and textual-based features. Multimedia Tools and Applications, 80(9), 13079–13097.

Burges, C. J. (2010). Dimension reduction: A guided tour. Now Publishers Inc.

Chakrabarty, N., Kundu, T., Dandapat, S., Sarkar, A., & Kole, D. K. (2019). Flight arrival delay prediction using gradient boosting classifier. In Emerging technologies in data mining and information security (pp. 651–659). Springer, Singapore.

Chowdhury, M. M. H., Rahman, A., & Islam, M. R. (2018a). Protecting data from malware threats using machine learning technique. In Proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA) (pp. 1691–1694). IEEE, Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICIEA.2017.8283111.

Chowdhury, M. M. H., Rahman, A. & Islam, M. R. (2018b) Malware analysis and detection using data mining and machine learning classification. *Advances in Intelligent Systems and Computing*, 580, pp. 266–274. Springer. https://doi.org/10.1007/978-3-319-67071-3_33.

Crawford, M., Khoshgoftaar, T. M. & Prusa, J. D. (2016). "Reducing Feature SetExplosion to Faciliate Real-World Review Sapm Detection," In Proceeding of 29th International Florida Artificial Intelligence Research Society Conference, Association for the Advancement of Artificial Intelligence (www.aaai.org).

Fletcher, S., & Islam, M. Z. (2019). Decision tree classification with differential privacy: A survey. ACM Computing Surveys (CSUR), 52(4), 1–33.
Gerlach, M., Shi, H., & Amaral, L. A. N. (2019). A universal information theoretic approach to the identification of stopwords. Nature Machine Intelligence, 1(12), 606–612.
Heydari, A., ali Tavakoli, M., Salim, N., & Heydari, Z. (2015). Detection of review spam: A survey. Expert Systems with Applications, 42(7), 3634–3642.
Hossain, A., Karimuzzaman, M., Hossain, M. M., & Rahman, A. (2021). Text mining and sentiment analysis of newspaper headlines. Information, 12(10), 1–15. [414]. https://doi.org/10.3390/info12100414
Khan, M. A., Shah, M. I., Javed, M. F., Khan, M. I., Rasheed, S., El-Shorbagy, M. A., ... & Malik, M. Y. (2022). Application of random forest for modelling of surface water salinity. Ain Shams Engineering Journal, 13(4), 101635.
Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering–a tertiary study. Information and software technology, 52(8), 792–805.
Krishna Rao, L. V., Sai Vamsi, S., Naazneen, S., Gowthami, S., & Bindu Bharati, G. (2022). Product Review Analysis. In ICT Systems and Sustainability (pp. 659–669). Springer, Singapore.
Kumar, J. (2020). Fake Review Detection Using Behavioral and Contextual Features. arXiv preprint arXiv:2003.00807.
Kumari S., Agarwal B., Mittal M., "A Deep Neural Network Model for Cross-Domain Sentiment Analysis," International Journal of Information System Modeling and Design (IJISMD), 12(2), 1–16, 2021.
Lai, C. H., & Hsu, C. Y. (2021). Rating prediction based on combination of review mining and user preference analysis. Information Systems, 99, 101742.
Liu, B., Liu, C., Xiao, Y., Liu, L., Li, W., & Chen, X. (2022). AdaBoost-based transfer learning method for positive and unlabelled learning problem. Knowledge-Based Systems, 108162.
Liu, Y., Wang, L., Shi, T., & Li, J. (2022). Detection of spam reviews through a hierarchical attention architecture with N-gram CNN and Bi-LSTM. Information Systems, 103, 101865.
Magidi, J., Nhamo, L., Mpandeli, S., & Mabhaudhi, T. (2021). Application of the random forest classifier to map irrigated areas using google earth engine. Remote Sensing, 13(5), 876.
Manaskasemsak, B., Tantisuwankul, J., & Rungsawang, A. (2021). Fake review and reviewer detection through behavioral graph partitioning integrating deep neural network. Neural Computing and Applications, 1–14.
Minnich, A. J., Chavoshi, N., Mueen, A., Luan, S., & Faloutsos, M. (2015, May). Trueview: Harnessing the power of multiple review sites. In Proceedings of the 24th International Conference on World Wide Web (pp. 787–797).
Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. arXiv preprint arXiv:1107.4557.
Păiş, V., & Tufiş, D. (2022). Capitalization and punctuation restoration: a survey. Artificial Intelligence Review, 55(3), 1681–1722.
Pappalardo, G., Cafiso, S., Di Graziano, A., & Severino, A. (2021). Decision tree method to analyze the performance of lane support systems. Sustainability, 13(2), 846.
Pradha, S., Halgamuge, M. N., & Vinh, N. T. Q. (2019, October). Effective text data preprocessing technique for sentiment analysis in social media data. In 2019 11th international conference on knowledge and systems engineering (KSE) (pp. 1–8). IEEE.

Rahman, A. (2019). Statistics-based data preprocessing methods and machine learning algorithms for big data analysis. International Journal of Artificial Intelligence, 17(2): 44--65.

Rahman, A. (2020). *Statistics for Data Science and Policy Analysis*. Springer.

Rahman, A. (2017). Small area housing stress estimation in Australia: Calculating confidence intervals for a spatial microsimulation model. Communications in Statistics Part B: Simulation and Computation, 46(9), 7466–7484. https://doi.org/10.1080/03610918.2016.1241406

Rahman, A., & Harding, A. (2016). Small area estimation and microsimulation modeling. Chapman and Hall/CRC.

Rahman, A., Harding, A., Tanton, R., & Liu, S. (2013). Simulating the characteristics of populations at the small area level: new validation techniques for a spatial microsimulation model in Australia. Computational Statistics and Data Analysis, 57(1), 149–165. https://doi.org/10.1016/j.csda.2012.06.018

Rausch, M., & Zehetleitner, M. (2022). Evaluating false positive rates of standard and hierarchical measures of metacognitive accuracy.

Sharma, P. S., Yadav, D., & Thakur, R. N. (2022). Web Page Ranking using Web Mining Techniques: A comprehensive survey. Mobile Information Systems (article ID 7519573), https://doi.org/10.1155/2022/7519573

Shiraz, M., Gani, A., Khokhar, R., Rahman, A., Iftikhar, M., & Chilamkurti, N. (2017). A distributed and elastic application processing model for mobile cloud computing. Wireless Personal Communications, 95(4), 4403–4423. https://doi.org/10.1007/s11277-017-4086-6.

Sockin, M., & Xiong, W. (2022). Decentralization through tokenization (No. w29720). National Bureau of Economic Research.

Tang, X., Dong, S., Luo, K., Guo, J., Li, L., & Sun, B. (2022). Noise Removal and Feature Extraction in Airborne Radar Sounding Data of Ice Sheets. Remote Sensing, 14(2), 395.

Thevaraja, M., & Rahman, A. (2019). Assessing robustness of regularized regression models with applications. In J. Xu, S. E. Ahmed, F. L. Cooke, & G. Duca (Eds.), Proceedings of the Thirteenth International Conference on Management Science and Engineering Management (1 ed., Vol. 1, pp. 401–415). (Advances in Intelligent Systems and Computing; Vol. 1001). Springer. https://doi.org/10.1007/978-3-030-21248-3_30.

Vachane, D. (2021). Online Products Fake Reviews Detection System Using Machine Learning. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(1S), 29–35.

Wang, J., Kan, H., Meng, F., Mu, Q., Shi, G., & Xiao, X. (2020). Fake review detection based on multiple feature fusion and rolling collaborative training. IEEE Access, 8, 182625-182635.

Wang, W., & Sun, D. (2021). The improved adaboost algorithms for imbalanced data classification. Information Sciences, 563, 358–374.

Wieczorek, J., Guerin, C., & McMahon, T. K-fold cross-validation for complex sample surveys. Stat, e454.

Yao, J., Zheng, Y., & Jiang, H. (2021). An Ensemble Model for Fake Online Review Detection Based on Data Resampling, Feature Pruning, and Parameter Optimization. IEEE Access, 9, 16914–16927.

Zheng, X., & Chen, W. (2021). An attention-based bi-LSTM method for visual object classification via EEG. Biomedical Signal Processing and Control, 63, 102174.