Received: 10 August 2020 | Revised: 25 December 2020 | Accepted: 9 February 2021

DOI: 10.1111/coin.12440

ORIGINAL ARTICLE

Computational Intelligence WILEY

Review prognosis system to predict employees job satisfaction using deep neural network

Furqan Rustam[1] | **Imran Ashraf**[2] | **Rahman Shafique**[1] | **Arif Mehmood**[3] | **Saleem Ullah**[1] | **Gyu Sang Choi**[2]

[1]Department of Computer Science, Khawaja Freed University, Punjab, Pakistan

[2]Department of Information & Communication Engineering, Yeungnam Univeristy, Gyeongsang, Korea

[3]Department of Computer Science & Information Technology, The Islamia University of Bahawalpur, Bahawalpur, Pakistan

Correspondence
Imran Ashraf, Department of Information & Communication Engineering, Yeungnam Univeristy, Gyeongsang, Korea.
Email: ashrafimran@live.com

Funding information
Brain Korea 21 Plus Program funded by the National Research Foundation of Korea (NRF, Grant/Award Number: 22A20130012814; MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program, Grant/Award Number: IITP-2020-2016-0-00313; National Research Foundation of Korea (NRF) funded by the Ministry of Education, Grant/Award Number: NRF-2019R1A2C1006159

Abstract

With the multitude of companies that flourish today, job seekers want to join companies with highly satisfied employees. So, job satisfaction prediction is an important task that helps companies in sustaining or redesigning employee policies. Such predictions not only help in reducing employee attrition but also affect the goodwill and reputation of a company. The higher satisfaction level of current employees attracts potential new employees and confirms the positive policies of a company toward its employees. Job satisfaction prediction can be performed using employee reviews either manually or via automated machine learning algorithms. This study first evaluates four widely used machine learning algorithms, that is, random forest, logistic regression, support vector classifier, and gradient boosting, and then proposes a deep learning model to predict employee job satisfaction level. Experiments are carried out on a dataset that contains text reviews from the employees of Google, Facebook, Amazon, Microsoft, and Apple. Three feature extraction methods are analyzed as well including term frequency-inverse document frequency (TF-IDF), bag-of-words (BOW), and global vector for word representation (GloVe). Performance is evaluated using accuracy, precision, recall, F1 score, as well as, macro average precision, and weighted average.

Furqan Rustam and Imran Ashraf contributed equally to this study.

The performance of the proposed model is compared with state-of-the-art deep learning models. Results demonstrate that the proposed model performs better than both the machine learning and state-of-the-art approaches.

KEYWORDS

bag-of-words, deep learning, employee satisfaction prediction, GloVe, term frequency-inverse document frequency, text classification

1 | INTRODUCTION

Recent years have witnessed the widespread usage of social platforms like Twitter, Facebook, and Google+, etc. People can share their thoughts and opinions about shopping by talking on these websites. Consequently, a large amount of data is generated every day that represent opinions. This data can be mined to analyze people's opinions from the public to private companies and effective policies can be devised based on those opinions. However, mining and analyzing so huge data is neither a trivial task nor appropriate with human experts. Instead, a large variety of artificial intelligence techniques have been proposed to perform this task. Machine learning is a subcategory of artificial intelligence that aims to learn from experience and perform human-like tasks. Many machine learning algorithms have been utilized for text mining and sentiment classification recently.[1-3]

Text classification serves as an essential module in many applications, such as web searching, information filtering, data organization, and sentiment analysis.[4] The wide use of the internet during the last few years elevated the importance of text classification. Apart from the text posted on social platforms in the form of reviews, many big companies like Google, Amazon, and Samsung, etc., have their platforms for their employees to share their opinions about company policies and comment on various products. This, in turn, let the companies design or revise policies and products in the light of posted reviews and comments. Similarly, reviews and comments about employees' job experience at a particular company play a central role to define the company's reputation for its supportiveness, positivity, and employee-centeredness. These reviews can be used to predict employee job satisfaction levels. A higher satisfaction level helps tighten the employee–company bond and attracts new potential employees for the company.

Traditional text classification works mainly focus on three topics: preprocessing, feature extraction, and classification using different types of machine learning algorithms.[5] Preprocessing involves the use of different techniques such as spell correction, stopwords removal, punctuation removal, etc. to transform the raw text into a standardized form that can be used for feature extraction later. Various feature extraction techniques like term frequency-inverse document frequency (TF-IDF), and bag-of-words (BoW), etc., are used on the preprocessed data to extract features that can be used by algorithms for training. Classification is later performed using the trained model on the testing data features. Deep learning models has been given special consideration recently owing to their superior performance in many areas including voice recognition,[6] text processing,[7] data mining,[8] text classification,[9] and sentiment classification,[10] etc. Contrary to machine

learning approaches, deep learning models do not require the feature extraction phase as they perform feature engineering automatically.[5] Although superior in performance, yet, deep learning models are not well studied for employee job satisfaction prediction.

This study proposes a deep neural network to predict the job satisfaction of employees from their reviews and comments. The dataset contains employee reviews and comments from Amazon, Apple, Microsoft, Google, and Facebook. Supervised machine learning models of random forest (RF), logistic regression (LR), support vector classifier (SVC), and gradient boosting (GB) are investigated as well to compare their performance with the proposed deep learning model. Machine learning models show different performance concerning the technique used for feature extraction. Consequently, three different feature extraction methods are used with machine learning algorithms. In summary, this study makes the following contributions

- Five machine learning algorithms are investigated to evaluate their performance for predicting employee job satisfaction using a dataset that contains employees' reviews and comments. Investigated machine learning algorithms are RF, LR, SVC, GB, and extreme GB (EGB).
- Three feature extraction methods are evaluated for their efficacy including term TF-IDF, BoW, and global vectors for word representation (GloVe).
- A novel deep neural network model is proposed to perform employee job satisfaction from employees' reviews. The performance of the proposed model is analyzed against the selected machine learning algorithms as well as, state-of-the-art deep learning models.

The rest of this paper is ordered as follows. Section 2 discusses the research works which are related to the current study. Section 3 first describes the structure of the dataset and machine learning algorithms used for the current study. Later, the proposed methodology as well as, the performance evaluation metrics are given in detail. Section 4 analyzes the results from the proposed approach and its comparison with other deep learning models. In the end, conclusion, limitations, and future work are given in Section 5.

2 | RELATED WORK

Employees of an organization play a vital role in its development and success and are called assets of an organization. Employee's job satisfaction represents his bond with the organization and reflects the nature of the organization's policies toward its employees. A higher job satisfaction level is indicative of an employee's constructive attitude as well as, attracts new potential employees for the organization's future progress. Concerning the increased use of social platforms like Twitter and Facebook, etc., employees show their content and discontent of particular organizations that depict whether the interest of employees and organizations coincide or collide. During the past decade, researchers and practitioners alike have increasingly challenged the use of users, particularly employees' feedback, of different organizations. Analysis of the reviews and feedback is made to form opinions that can help organizations design and revise policies for employees' higher satisfaction. A variety of research works can be found in the literature that focuses on the use of employees' feedback to predict many factors associated with employees like classification into satisfied and unsatisfied employees, turnover prediction, and productivity prediction, etc.

For example, authors in Reference 11 predict employee turnover of a company using the data collected from the human resource information system. Employee turnover is a very important aspect of a company and represents the impact on the growth and productivity of an organization. Several machine learning algorithms are used for this purpose and their performance is analyzed for prediction accuracy. Results demonstrate that EGB outperforms other machine learning classifiers. Similarly, another study[12] performs employee classification into satisfied and unsatisfied groups. The authors used the Waikato environment for knowledge analysis (WEKA) tool to generate a classification decision tree model. A total of 309 records of employees working at the Institute of Nigeria from 1978 and 2006 are used as the dataset. Results show that the decision tree works better than those of other machine learning algorithms for employee classification.

The study[13] proposed an approach that uses employee feedback to investigate factors associated with employee satisfaction, retention, and productivity. The data is collected through employee communication on social media platforms. Three feature engineering methods including Doc2Vec, BoW, and Word2Vec are utilized with a support vector machine for this purpose. Research proves that learning representations show superior performance than those of working with standard TF-IDF weighting schemes. Similarly, the authors investigate the employee's dissatisfaction associated factors in Reference 14. The analysis is carried out using employee reviews collected from "indeed.com." A decision support system is built that can predict employee satisfaction and dissatisfaction level. General sentiment analysis is compared with dictionaries curated for this domain to show the efficacy of sentiment analysis. In the same way, sentiment analysis techniques can be used to investigate factors that represent the satisfaction and dissatisfaction level of employees. The study[15] carries out a similar task using unsupervised machine learning techniques. Employee job satisfaction level is predicted using online reviews. Linear discriminant analysis is used for identifying reviews' salient features. Employee sentiment is measured using the general inquirer dictionary for counting positive and negative terms. Employee satisfaction is weighted using firm outlook and employee sentiment. The authors extract the best factors that describe the satisfaction and dissatisfaction of employees and then divide these factors into different categories such as industry level satisfaction, organization level satisfaction, and group level satisfaction.

The authors introduced a sentiment embedding technique in Reference 16 for company profiling. The aim is to rank companies into various general categories using the sentiment values of the salary, location, and work-life aspects. Reviews are gathered from the Glassdoor website for this task and classification is performed using a novel ensemble classifier of unsupervised and machine learning approaches. RF and C4.5 are used for talent forecasting using employee experience and knowledge. Employee performance prediction is another important aspect for many large companies. It helps to foresee shortcomings in employee performance and take appropriate measures to overcome such limitations. Research work[17] resort to supervised machine learning techniques for the prediction of employee performance.

Despite the above-cited works, the employee satisfaction research area has limited work and holds a huge room for improvement. It can not only elevate employees' job satisfaction levels but can be helpful for companies to foresee and plan future investments as well. Present research works do not make an exhaustive investigation of machine learning techniques and are therefore needed to be extended. Moreover, massive documents and review collections from social platforms need enhanced information processing methods for searching, retrieving, and organizing text.[18] The performance of customary supervised classifiers has degraded as the number of documents has increased. The use of deep learning models has proven to show first-rate performance

TABLE 1 Description of dataset features

Feature	Description
Index	Index of record
Company	Company names
Location	Location of companies
Date	Employee reviews date
Job-Title	Job title of employee like : software engineer
Summary	Summary of employee review
Pros	Pros for company employees
Cons	Cons for company employees
Overall Rating	Overall ranting given by employee to company 1–5
Work/Life Balance Rating	Rating by employee on work and life balance 1–5
Culture and Values Rating	Rating by employee for culture of company 1–5
Career Opportunities Rating	Career opportunities rating to company 1–5
Comp and Benefits Rating	Rating for company according to benefits 1–5
Senior Management Rating	Rating for company according to management 1–5
Helpful Review Count	A count of how many people found the review to be helpful
Link to Review	This will provide you with a direct link to the page that contains the review

for such tasks. Because of the results of deep learning models, this study adopts the use of a deep learning neural network for employee job satisfaction levels.

3 | MATERIAL AND METHODS

The task of employee satisfaction prediction can be split into three stages, that is, data preprocessing, training the classification models, and prediction on the test set using the trained model. These stages involve using various techniques and are separately described in the following sections.

3.1 | Dataset description

The dataset used for experiments is taken from ”Kaggle” which contains reviews of Google, Amazon, Apple, Netflix, Microsoft, and Facebook employees. The dataset used for experiments is the ”Employees reviews dataset” which is publicly available and can be accessed at Reference 19. It also contains the employee given rating of each company. The dataset has four text variables, that is, summary, pros, cons, and advice (for people interested to work in the company), and seven numeric variables where employees give rating scores from 1 to 5 for departments of companies. The description of each attribute of the dataset is given in Table 1.

TABLE 2 Data samples for the dataset

Company	Location	Dates	Job-title	Summary	Overall rating
Google	Tokyo (Japan)	April 17, 2016	Current Employee	APAC Customer care	3
Google	Taipei (Taiwan)	May 16, 2016	Former Employee	Good	3
Facebook	Menlo Park, CA	May 10, 2014	Current Employee	Big company, good benefits	4
Microsoft	Hyderabad (India)	December 13, 2010	Current Employee	Having fun at Microsoft	5
Apple	Houston, TX	July 4, 2017	Current Employee	Don't rush to apply just yet	2
Netflix	Lansing, MI	November 11,2016	Current Employee	Come join us!	4
Netflix	Los Gatos, CA	October 26, 2011	Current Employee	Great place to work so far	4

Form these sixteen variables, two attributes including summary (employee review about the company) and overall rating are used for current research work. We consider it a classification task and take reviews as features and overall rating as classes. The dataset contains 67,529 records and each record contain reviews and rating.

Table 2 contains a sample of the dataset with four variables. From these variables, we use summary and overall rating in our research experiment. Current research aims at predicting employees' job satisfaction through the use of employee reviews. So, only two attributes of the dataset are considered. Because the summary attribute contains reviews of customers, so it is sufficient to perform satisfaction prediction which is given in the form of rating as the overall rating attribute infers.

Figure 1 shows the numbers of dataset records for each company. It shows that Amazon has more employee reviews in the dataset than any other company. It is followed by Microsoft, Apple, and Google. While Facebook and Netflix do not have a large number of reviews, but still enough to perform an analysis. Figure 2 shows the overall rating for each company in the dataset. It shows that Facebook users are more satisfied than the employees in any other company and ranks it with the highest rating frequently.

3.2 | Feature extraction technique

Three feature extraction techniques have been selected for the experiments. A brief description of each technique is given in the following sections.

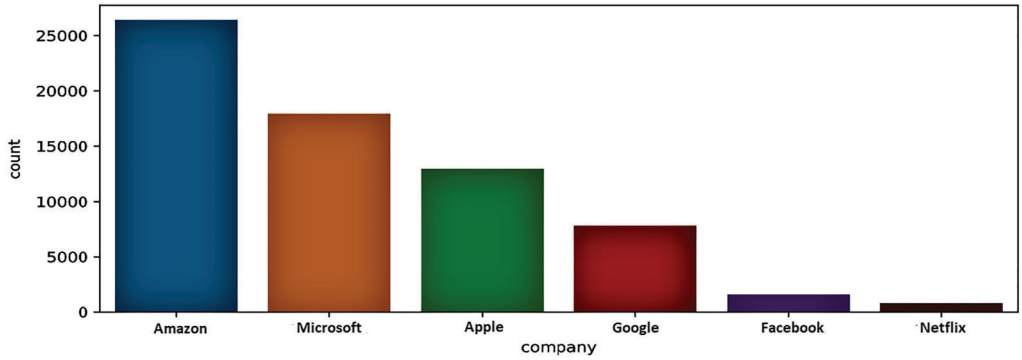

FIGURE 1 Total number of reviews in the dataset for each company

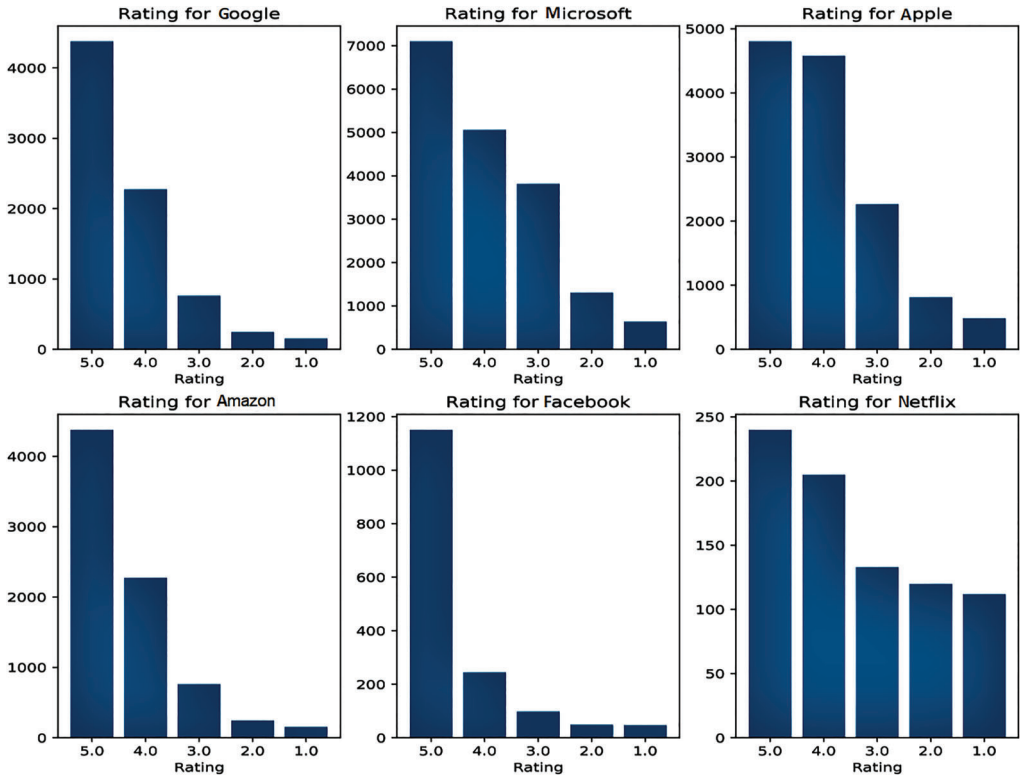

FIGURE 2 Total number of ratings from 1 to 5 for each company

3.2.1 | Term frequency-inverse document frequency

The TF-IDF is one of the most widely used feature extraction techniques for text analysis. Most of the feature extraction techniques use information gain to select the most valuable feature from the data to perform experiments with excellence.[20] Predominantly, text demonstration includes two tasks, that is, indexing and weighting, and the TF-IDF weight is a weight often used in information retrieval and text mining.[21] TF-IDF finds the weight of each term in a given document. Currently,

TABLE 3 Term frequency-inverse document frequency result on samples for the dataset

also	best	company	demand	dream	job	place	unique	work
0.000	0.517	0.680	0.000	0.000	0.000	0.000	0.000	0.517
0.490	0.373	0.000	0.490	0.000	0.000	0.490	0.000	0.373
0.000	0.000	0.000	0.000	0.447	0.447	0.000	0.447	0.000

TABLE 4 Sample data from the dataset for term frequency-inverse document frequency feature extraction

No	Text review
1	Best Company work
2	best place work also demand
3	unique one kind dream job

there are numerous feature weighting methods, which are derived using different assumptions for feature characteristics in texts. TF-IDF is evolved from the inverse document frequency (IDF). Term or feature which occurs in many documents is not a good discriminator, hence not important for processing of text. TF-IDF infers that the terms occurring more frequently should be given less weight and the terms which appear rarely in documents should be given high weight.[21] TF-IDF is the product of TF and IDF. TF represents the occurrence of a term in a document while IDF rewards tokens that are rare overall in a dataset. If a rare word occurs in two documents, then it is more important to the meaning of each document. We can define TF mathematically as follow

$$\mathrm{TF}(t) = \frac{N}{D}, \tag{1}$$

where N is the number of times term t appears in a document and D is the total number of terms in the document.

IDF can be calculated using

$$\mathrm{IDF}(t) = \log \frac{d}{dt}, \tag{2}$$

where d is the total number of documents and dt is the number of documents with the term t in it.

TF-IDF computes the weight of each term by using the equation

$$W_{t,d} = \mathrm{TF}_{t,d} \left(\frac{N}{D_{f,t}} \right), \tag{3}$$

where $\mathrm{TF}_{t,d}$ is the number of occurrences of term t in document d. $D_{f,t}$ is the number of documents containing the term t and N is the total number of documents in the corpus.

Table 3 shows the results when the TF-IDF method is applied to the following sample data as shown in Table 4 :

3.2.2 | Bag of Words

The BoW feature extraction method is an important method used to extract features from text data. The BoW is simple to implement and interpret, yet, often more efficient than other complicated techniques. Using the training set, BoW generates the vocabulary of all unique words

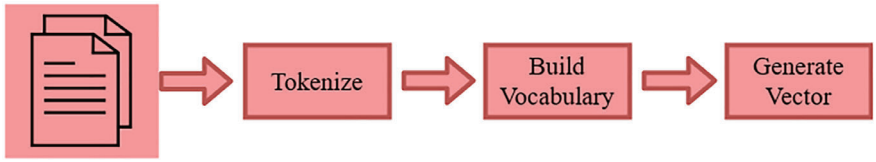

FIGURE 3 Steps involved for gathering bag-of-words features from text data

TABLE 5 Sample data from the dataset for BoW feature extraction

No	Text review
1	Best company to work
2	Best place to work
3	Wonderful company for wonderful employees

TABLE 6 Result of applying bag of words on sample data

No	best	company	to	work	place	wonderful	for	employees
1	1	1	1	1	0	0	0	0
2	1	0	1	1	1	0	0	0
3	0	1	0	0	0	2	1	1

from the text. To put it simply, BoW is a collection of words that are used to represent a sentence with word count. Owing to its simplicity and performance, it is widely used for text classification, information retrieval, natural language processing, and topic modeling.[3,22] In this study, we use sci-kit learn library class "CountVectorizer" for BoW features which converts a collection of text reviews into vocabulary vector that is later used as input for machine learning models.[23] Figure 3 shows the steps carried out in BoW to convert text data into an input vector.

To further explain the working of BoW, we took sample data from the dataset as shown in Table 5, and applied the BoW method on it.

"CountVectorizer" is applied to the sample data and the result is shown in Table 6. "CountVectorizer" extracts the numbers of features from sample data and represents the occurrence of each feature in reviews. The number of features depends on the vocabulary size which means that the feature vector is proportional to the vocabulary size.

3.2.3 | GloVe

GloVe (Global Vectors) for word representations is by Standford University and often used to obtain the vector representation for words.[24] The GloVe is an alternative method of creating word embedding which works exactly like matrix factorization. It does not build the term-document matrix-like BoW and TF-IDF but the word-word co-occurrence matrix. The GloVe is advantageous for large corpus due to its scaling capability. To show its working mechanism, we take the sample text, that is, "I love to work here. I love Facebook. I love Google" and applied GloVe. Results are shown in Table 7.

TABLE 7 Result of applying Global Vectors on sample data

	I	love	to	work	here	Facebook	Google
I	0	3	0	0	0	0	0
love	3	0	1	0	0	1	1
to	0	1	0	1	0	0	0
work	0	0	1	0	1	0	0
here	0	0	0	1	0	0	0
Facebook	0	1	0	0	0	0	0
Google	0	1	0	0	0	0	0

TABLE 8 Model hyper parameters settings.

Model	Hyper parameters
midrule RF	n_estimators=150, max_depth=100
LR	solver= 'liblinear' multi_class= 'ovr', C=3.0, random_state=5
SVC	kernel='linear', C=3.0, random_state=5
GB	n_estimators=150, learning_rate=0.2, max_depth=100
EGB	n_estimater=150, learning_rate=0.2, max_depth=100

Abbreviations: EGB, extreme GB; GB, gradient boosting; LR, logistic regression; RF, random forest; SVC, support vector classifier.

Table 7 shows the GloVe output word-word matrix Z, where $Z_{p,q}$ is the strength which shows how frequently the word p occurs in the context of word q. The co-occurrences which are rare or never occur in matrix Z contain less information as compared to more frequent co-occurrence, as "I, love" co-occurrence is more frequent co-occurrences than that of others. GloVe predicts the surrounding word by maximizing the probability of P(ContextWord)|P(CenterWord) by performing dynamic LR.

3.3 | Machine learning models used for experiments

We employ five machine learning models including RF, LR, SVC, GB, and EGB for our experiments. A brief description of each of these classifiers is given in the following sections. Classifiers are trained with different hyper-parameters on the training data and the best hyper-parameters setting are selected using the hit and trial method. Best hyper-parameters for selected classifiers are shown in Table 8.

3.3.1 | Random forest

RF is an ensemble model that uses the bagging method; in the bagging method numbers of base-learners train on bootstrap data samples.[25] By subsampling of the training dataset with replacement, a bootstrap data sample is obtained, where the size of a sample is the same as the original training dataset size. We optimized and tuned RF with different hyper-parameters to get

the best result. In RF, multiple decision trees are generated to predict the target class by voting these decision trees. In the decision tree construction, the main challenge is the identification of attributes for the root node at each level. This process is known as feature selection. We have two popular feature selection measures, that is, information gain and Gini index. To determine the Gini value, we consider the probability of finding each class after a node, we sum the square of those values and we subtract this amount by 1. For this reason, when a subset is pure, the Gini value will be 0, because the probability of finding that class is 1 indeed. And in that case, we say we have reached a leaf because there is no need to split anymore as we achieved our goal. So in RF to construct a decision tree we compute the Gini value by using the below equation

$$\text{Gini} = 1 - \sum_{i=1}^{NC} (p_i)^2, \tag{4}$$

where P_i represents the probability of an element being classified for a distinct class and NC is the number of categories. Gini value gives us the impurity of data in the dataset and information gain gives the purity of data in the dataset.[26] Information gain is also used for the selection of the best attribute. To find the information gain for the construction of the decision tree, first, we find the entropy. There are two steps for measuring the information gain for each feature: calculate the entropy of the target, and entropy for every feature needs to be calculated. Later, the feature entropy is subtracted from the entropy of the target using the information gain formula. So when multiple decision trees are generated in RF, voting is performed on the predictions of decision trees. In our experiment, we used the n_estimator parameter with the value of 150 which means that RF will generate 150 trees to classify the given data. We use another parameter in the RF algorithm which is max_depth and we set its value to 100. This parameter will restrict each tree to a maximum of 25 level depth which will reduce the complexity of trees and avoid over-fitting of the model.

3.3.2 | Logistic regression

LR is a statistical model that uses the logistic function to process the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities. LR is very efficient, specifically for binary classification. This logistic function is a common "S" shape (sigmoid curve), defined as

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}, \tag{5}$$

where e is the base of the natural algorithm, x_0 is the midpoint value of the sigmoid, L is the maximum value of the curve, and k is the logistic growing rate.

The values of x are real number ranging from $-\infty$ to $+\infty$, while logistic functions are used in LR to calculate how the probability p of an event may be affected by one or more explanatory variables

$$p = f(v + ux), \tag{6}$$

where x is the clarifying variable and v, u are model parameters to be fitted and f is the standard logistic function.

In our research experiment, LR uses different hyper-parameters such as random_state, solver, multi_class, and *C*. We set random_state value to 5 and solver = "liblinear" because "liblinear" is a good choice when the dataset is not very large, whereas "sag" and "saga" are better for large ones. The value of multi_class="ovr," here we choose "ovr" because of the binary classification problem and C ="3.0," where C is the inverse of regularization strength and must be a positive float number.

3.3.3 | Support vector classifier

SVC is another widely used classifiers for text classification. SVC investigates records, characterizes preference, limits, and makes use of the components for the calculation, that is attained within the entrance area.[27] In our experiment, we pass three parameters which are the kernel, *C*, and random_state. A "linear" kernel is used for our experiment, and we set the value of *C* to 2.0 and random_state to 5. Kernel specifies the type of algorithm to be used. Kernel value must be "poly," "rbf," "sigmoid," "linear," "precomputed," or "callable." If it is "callable" then it is used to precompute the kernel matrix from data matrices; that matrix should be an array of shapes (n_samples, n_samples). In SVC default kernel is "rbf." We use the "linear" kernel in our experiment because it is recommended for text classification, *C* is the penalty parameter of the error term. Like in LR, a smaller value of *C* specifies stronger regularization. With these settings of the parameter, we achieve the best result from SVC.

3.3.4 | Gradient boosting

GB is a supervised machine learning model used for both classification and regression problems.[28] GB trains multiple weak learners sequentially to make a strong learner under ensemble learning criteria. The sequential coupling of weak learners proves to be very useful in reducing errors and boosting the accuracy of the model. Decision trees are usually used as weak learners in GB. GB trains the first tree on original data and the next/following tree on the error of the previous tree to minimize the prediction error using the boosting method. Each new decision tree is a step toward minimizing the prediction error. GB can be overfitted on training data because it uses the greedy approach. To overcome the overfitting problem of GB, the learning rate and the number of the appropriate weak learner are used.[29] Given its simplicity, GB is used in many tasks such as imbalanced credit scoring,[30] and scene classification,[31] etc.

We use different hyper-parameters to get the best results from GB. We tune these parameters to boost the performance of GB such as the n_estimator parameter is used with a value of 150 which means that GB uses 150 weak learners (decision tree) in the prediction procedure. To stop over-fitting we use a learning_rate parameter with a value of 0.2. Another parameter max_depth is used with a value of 100 to reduce the complexity of the learning model. It is also helpful to reduce the overfitting of the model.[30]

3.4 | Multilayered perceptron

A deep neural network (DNN) is an intricate form of neural network, that is, with more than two hidden layers. DNN performs complex mathematical calculations to process the data. Neural

networks are applied to learn word demonstrations in text mining.[32] The neural demonstration of words is called word embedding (real-valued vector). The word embedding enables neural networks to measure word similarity by finding the distance between two embedding vectors. The pretrained word embed neural networks show their superior performance in many NLP tasks.[5] DNNs are although often harder to train than simple neural networks, yet more powerful than simple neural networks. Deep learning models efficiently perform computational tasks by combinations of nonlinear processing elements organized in layers. This association of simple elements allows the DNN to predict correctly on new data.[33] Convolution neural networks (CNN) and recurrent neural networks (RNN) are widely used networks for image and text data, respectively.[34,35] We use DNN for the experiment owing to its higher performance in a variety of NLP tasks.

The use of neural networks for text mining witnessed an increased interest during recent years. For example, the authors in Reference 36 use the MLP model to learn semantic text classification. Similarly, research[37] utilizes MLP for topic spotting in the text. Several different architectures including variant numbers of layers and hyperparameters are tested to achieve high performance. Another study[38] proposes a diachronic propagation framework to incorporate the historical impact into currently learned features through diachronic connection. This approach improves the performance of traditional neural networks for text classification. In the same vein, the authors use deep neural network architecture for text classification and prove the significant performance of the neural network on text data in Reference 39. Results demonstrate that vector data representation can also improve the accuracy of the classification.

Keras framework is used to implement the MLP algorithm. MLP algorithm uses a deep neural network to find the patterns and sequences from data. In our MLP implementation, we use two activation functions, that is, rectified linear unit (relu) and sigmoid. The relu is linear for all positive values, and zero for all negative values. The relu can be defined as $f(z) = \max(0, z)$. The relu activation function is used on account of its ability to achieve high precision in many applications.[34] The sigmoid function is sometimes preferred because it has a steady-state at 0. However, relu has been found to yield superior results in many different settings. The sigmoid function can be defined as

$$g(z) = \frac{1}{1 + e^{-z}}. \tag{7}$$

Sigmoid is particularly used for problems where we have to predict probability as an output. Meanwhile, the probability of anything exists only between the range of 0 and 1, so sigmoid is the right choice with relu in our algorithm. To avoid over-fitting we use the "dropout" technique in our algorithm because the dropout rate can avoid a learning algorithm from complex learning and overfitting.[40] We set a dropout rate of 0.2 in the neural network layers used in our research experiment. Because of this dropout technique we use 100 epochs in the training phase while minimizing the risk of over-fitting. We applied the "Adam" optimization algorithm, which is an effective method for training neural networks. We use "Adam" because this optimizer is frequently used for RNNs in text analysis. "Adam" finds the most optimal value for the neural network. Data are fed to the network in batches of 128 (batch-size = 128). We set the input dimension of 3000 because we use 3000 features in our experiment. We compile our neural network model with loss="binary_crossentropy," optimizer= "Adam," and metrics = ["accuracy"]. With these settings, our algorithm performs very well and gives us a high accuracy. Table 9 shows the parameter settings for the neural network used in the current study.

TABLE 9 Parameters setting for the neural network used in current study

Parameter name	Used value
Dropout layer	0.2
Epochs	100
Optimizer	Adam
Batch size	128
Feature size	3000
Loss	binary_crossentropy
Metrics	Accuracy

TABLE 10 Distribution of the dataset for training ad testing

Classes	Records	Selected records	Experiment records	Training	Testing
Unsatisfied	58,341	9188	9150	6886	2264
Satisfied	9188	9188	9188	6867	2321
Total	67,529	18,376	18,338	13,753	4585

3.5 | Proposed approach

In this study, machine learning and deep learning techniques are used to solve the employee classification problem. The problem in the current study is to classify satisfied and unsatisfied employees using the reviews of their working companies. We have the summary of reviews as features and overall rating 1–5 as classes. First, we reduce the dimension of classes, for this, we convert overall rating values of 1–5 into two classes, that is, satisfied and unsatisfied. Those employees who give ratings equal to or greater than 2.5 are assigned to the satisfied employee category and those who give a rating less than 2.5 are assigned to the unsatisfied employee category. In a total of 67,529 records, 58,341 belongs to satisfied employees and 9188 records to the unsatisfied employee category. This shows the imbalanced data problem which could substantially degrade the classification performance. To overcome this issue, an equal number of records are extracted from both satisfied and unsatisfied classes. So, a total of 18,376 examples are taken from the dataset from each class. Records from the satisfied employee class are selected randomly. After removing the null value from taken records we get 18,338 records to perform our experiments. Optimization is performed only during training the model using the training data. Distribution of data for total records, training, and testing records are given in Table 10.

Figure 4 shows the architecture of the methodology adopted for the experiments. The same flow as well as, preprocessing is adopted for all classification algorithms.

3.5.1 | Preprocessing steps

Preprocessing the dataset aims at cleaning the data and involves the steps of tokenization, punctuation removal, lowercase conversion, removal of numbers and stopwords, and stemming. Preprocessing plays a very important role to reduce data complexity, lower the training time, and

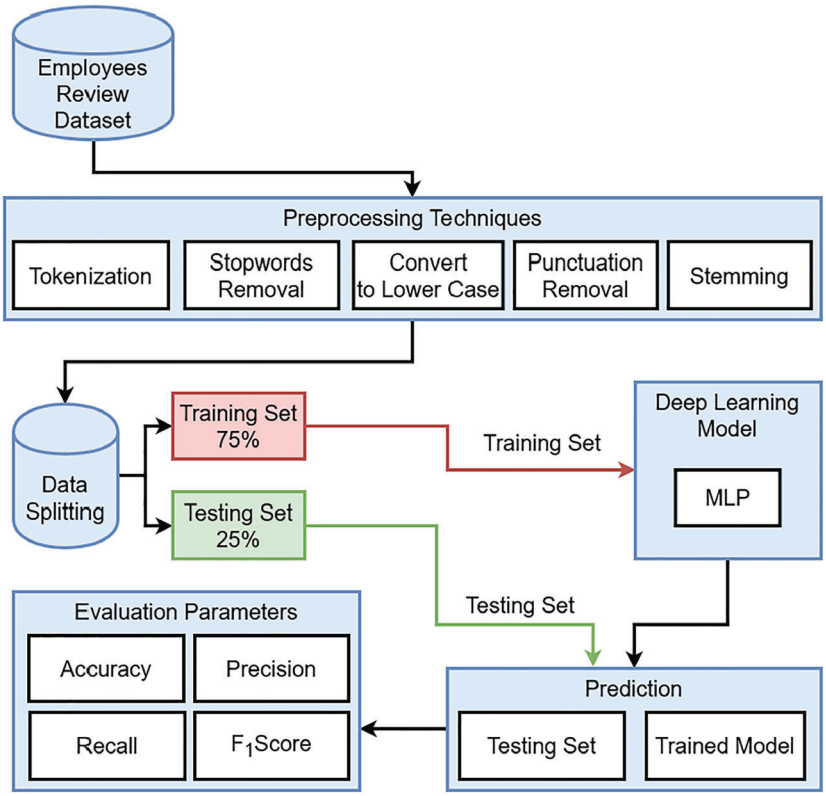

FIGURE 4 Architecture of the methodology followed for experiments

improve classification accuracy.[41] Preprocessing id performed using the natural language toolkit (NLTK) library of Python.[42]

Tokenization: This process splits the text into smaller pieces known as tokens. Words, numbers, punctuation marks, and others can be considered as tokens. Tokenization divides user reviews into terms or words. The various criterion is used for tokenization, for example, whitespace, and a punctuation mark, etc.

Punctuation removal: In the text, punctuation affects the structure and interpretation of the meaning of a sentence. It is used for the ease of human beings to understand the text easily. However, it complicates the learning process for machine learning algorithms because such algorithms can not easily differentiate it from other characters.[43] Because of that, we remove the punctuation symbols. After observation of data following punctuation is removed from text: $ @ "[ ] ( ) / | , ; . ".

Conversion to lowercase: It is very important to convert text data to lower case as text analysis is case sensitive. It works on word count, so words are counted separately if the case is different and the accuracy of learning algorithms can be affected. Such as "go" and "Go" are the same words concerning meaning but machine learning algorithms take them as separate words.[43]

Removal of numbers: Removing numbers can increase the accuracy of the model when they are not relevant to the analyses. We remove the number because they are not important to find the satisfaction of an employee. Instead, words like "good," "bad," in a sentence are important and represent the user sentiment.

TABLE 11 Result of data preprocessing steps

Before preprocessing	After preprocessing
Good company with good benefits, lots of red tape and big company process, however.	Good company good benefit lot red tape big company process however
Fun But Hard to work 24 h.	Fun hard work hour
Still the best place I have ever worked in 15-year work history.	Still best place ever work year work history
Requests everything from you, but provides for all your needs :).	Request everything provide need

TABLE 12 List of parameters for MLP used in the current study

Parameter	Used value
Input layer	256 neurons @ relu
Dropout layers	After input and hidden layer 1 with 0.2
Hidden layer 1	128 neurons @ relu
Hidden layer 2	64 neurons @ relu
Output layer	@ sigmoid

Stopwords removal: Stopwords are the most common words used in a language such as ”a,” ”on,” ”the,” ”all,’ ”is.” These words usually do not carry important information and their removal from the data does not reduce the performance of the algorithm. Similarly, conjunctions like “and,” “or,” “but” and pronoun“he,” “she,” “it” do not largely affect the performance of text classification and are removed from the data kadhim2018evaluation.

Stemming: Stemming is a process to get the root or base form of a word. As we mentioned above, text analysis works with the word count so stemming helps to increase accuracy. For example “go” and “goes” are words with the same meaning but the machine technique takes them as a different term. if we use the stemming technique it will convert “goes” to its root form “go.” For stemming we use the Porter stemming algorithm.[44]

Table 11 shows the results of the preprocessing phase on sample data taken from the dataset.

Once preprocessing is complete, we divide the dataset into the training set and testing set with a ratio of 75:25. We use 75% data for training and 25% for testing purposes. Features are extracted and fed to RF, LR, SVC, and GB for training. For the second approach, preprocessed data is given to the proposed neural network. The data split ratio is the same for the neural network as that of other algorithms. The same features are used to train neural network architecture that we use for traditional machine learning classifiers. Figure 5 shows the architecture of the proposed neural network for employee job satisfaction.

The proposed MLP is a deep learning model that uses multiple hidden and dropout layers to achieve the classification of input data to satisfied and unsatisfied employee classes. First, at the input layer, we use 256 units with a relu activation function followed by a dropout layer with a 0.2 dropout rate. At the first hidden layer, we use 128 units with the relu function again followed by a 0.2 dropout rate. The second hidden layer has 64 units and at the output layer, we use the sigmoid function. We compile the model with ”binary_crossentropy” loss function and the “Adam” optimizer. All parameters used in the design of the MLP are listed in Table 12.

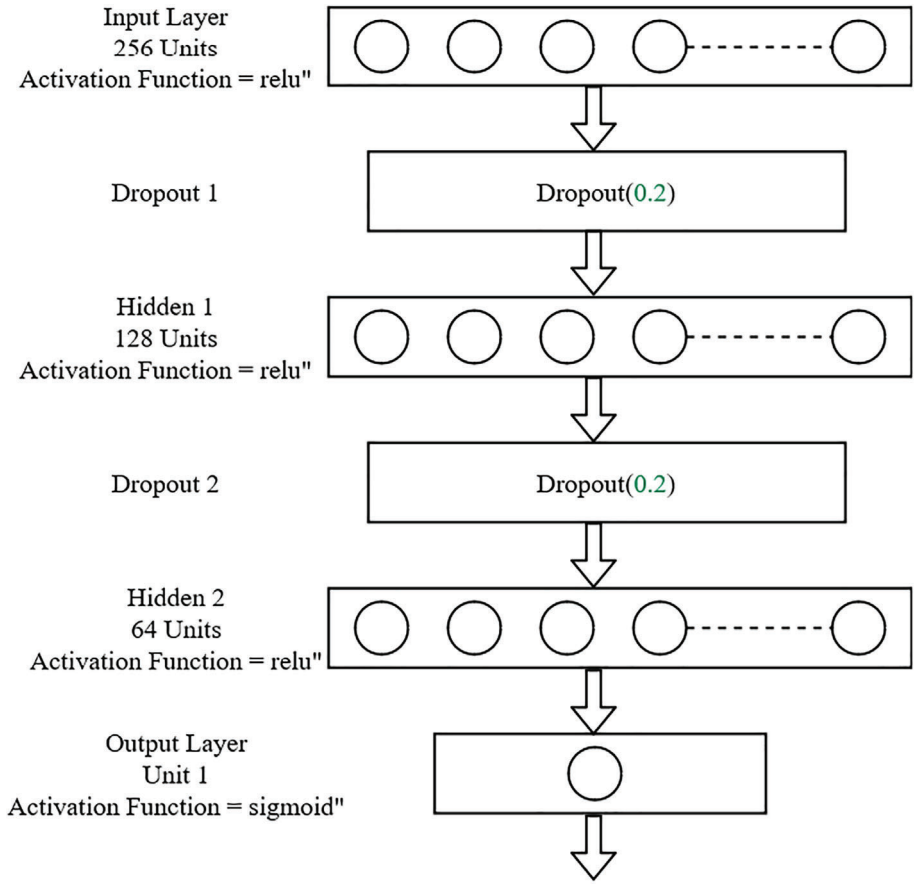

FIGURE 5 Architecture of the proposed deep neural network

3.6 | Experiment setup

Although several research works use the WEKA tool concerning the availability of classification algorithms, we do not use WEKA. Instead, we implemented the proposed MLP using Python 3.0 on Jupyter notebook. For using the machine learning algorithms, scikit-learn toolkit is used whereas MLP architecture is implemented using Keras framework.[45,46] Experiments are performed on an Intel Core i7 7th generation machine operating on Windows 10.0. Preprocessing steps are performed using the NLTK while feature extraction and implementation of machine learning models are done using scikit-learn library. For GloVe features, we used the zeugma 0.41 library for word embedding which is compatibles with scikit-learn pipeline.[47]

3.7 | Evaluation parameters

In this study, we use four evaluation parameters accuracy, precision, recall, and F1 score.

Accuracy is used for the evaluation of learning models. Accuracy is the correctness of trained models. To compute accuracy, the total number of correct predictions is divided by

the total number of predictions. The lowest accuracy score is 0 and the highest accuracy score is 1.

$$\text{Accuracy} = \frac{\text{Number of correct } \textit{predictions}}{\text{Total number of } \textit{predictions}}. \tag{8}$$

Recall and precision are calculated using four basic notations, explained as follows.

True Positives (TP) is when the model predicted the example as satisfied (the employee is satisfied), and the actual label of the example is also satisfied.

True Negatives (TN) is when the model predicted the example as unsatisfied (the employee is not satisfied), and the actual label of the example is also unsatisfied

False Positives (FP) is when the model predicted the example as unsatisfied, but the actual label of the example is satisfied. (FP is also known as a "Type I error.").

False Negatives (FN) is when the model predicted the example as satisfied, but the actual label of the example is unsatisfied. (FN is also known as a "Type II error.").

Recall is the completeness of the classifiers. The lowest recall score is 0 and the highest recall score is 1.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{9}$$

Precision is the ratio of correct positive prediction to the total positive prediction. Precision is the number of TP divided by the number of TP and FP. The lowest precision score is 0 and the highest precision score is 1.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{10}$$

F1 score shows the balance between the precision and the recall. F1 score is the harmonic mean of precision and recall. The lowest F1 score is 0 and the highest F1 score is 1.

$$\text{F1score} = 2 \times \frac{\text{Precison} \times \text{Recall}}{\textit{Precision} + \text{Recall}}. \tag{11}$$

We also find the macro average and weighted average for both class scores. A macro-average calculates the metric independently for each class and then takes the average (hence treating all classes equally). For macro precision and recall, the method is straight forward. Just take the average of the precision and recall of different classes. Here are the formulas for finding macro average precision (MAP), macro average recall (MAR), and macro average F1 score (MAF).

$$\text{MAP} = \frac{\text{Precison}_1 + \text{Precision}_2}{2}. \tag{12}$$

$$\text{MAR} = \frac{\text{Recall}_1 + \text{Recall}_2}{2}. \tag{13}$$

Here Precision_1 is the precision of the unsatisfied employee class and Precision_2 is the precision of the satisfied employee class. Macro F1 score is the harmonic mean of both macro average precision and macro average recall.

$$\text{MAF} = 2 \times \frac{\text{MAP} \times \text{MAR}}{\text{MAP} + \text{MAR}}. \tag{14}$$

4 | RESULTS

This section contains the experiment results along with a discussion on the results. Experiments are performed with various classification models to check the performance of these classifiers with the same data and preprocessing. Results discussed in this section show the performance with a 75:25 split. Two other split ratios are evaluated as well including 70:30 split and 80:20 split. However, the accuracy using these two splits are 81% and 82%, respectively. Due to the high accuracy of 75:25, we adopt this split ratio for the experiments.

4.1 | Results using TF-IDF

The sole purpose of using a variety of classification models such as LR, RF, SVC, GB, and MLP is to give a comparison between machine learning and deep learning algorithms. With these conditions and scenarios, the LR performs a little bit better than RF and SVC, but the deep learning algorithm MLP outperforms the other algorithms. Table 13 shows the results of experiments using TF-IDF.

We can see that MLP performs very well on the test dataset and reward with a maximum F1 score of 0.83. Accuracy as well as, precision and recall values are high with MLP than those of other classifiers. Supervised MLP shows better performance for text classification. MLP is very flexible which is helpful to learn the mapping from input to output. The words of a document can be reduced to one long row of data and fed to an MLP which shows the flexibility of MLP. In our study, MLP performs better because MLP does not make any assumption regarding the underlying probability density functions or other probabilistic information about the pattern classes under consideration in comparison to other probability-based models.[48] Hence, MLP performs better than CNN and long short-term memory networks even with a comparatively smaller dataset. Regarding machine learning algorithms, LR performs better than those of other algorithms in terms of accuracy, precision, recall, and F1 score.

4.2 | Results using BoW

Experiments are performed with BoW features as well to analyze the impact of changing the feature extraction from TD-IDF to BoW. The objective is to find the suitability of a specific feature extraction procedure for review classification. Results for performance metrics using BoW are shown in Table 14. Results indicate that the performance of machine learning classifiers have been slightly reduced when BoW is used. MLP, on the other hand, experience a large accuracy fall of 0.04, as well as, a substantial decrease in precision, recall, and F1.

TF-IDF shows superior performance than that of the BoW feature for text classification. The BoW contains only the count of word occurrence for a given review and does not provide information regarding the importance of a word, that is, whether it is a rare word or a common one. At the same time, TF-IDF contains information on the most important and common words and performs better. Moreover, as the size of the review increases, vocabulary size increases in the same order. It leads to sparsity in BoW because with an increase in vector size it contains many 0s. As a result of the increased size of the

TABLE 13 Accuracy, precision, recall, and F1 results for all the classifiers using term frequency-inverse document frequency

Model	Accuracy		Precision	Recall	F1
RF	0.78	Unsatisfied employee	0.74	0.85	0.77
		Satisfied employee	0.81	0.70	0.76
		Macro average	0.77	0.77	0.77
		Weighted average	0.78	0.78	0.77
LR	0.78	Unsatisfied employee	0.76	0.83	0.79
		Satisfied employee	0.81	0.76	0.78
		Macro average	0.79	0.80	0.79
		Weighted average	0.78	0.78	0.78
SVC	0.77	Unsatisfied employee	0.76	0.81	0.78
		Satisfied employee	0.78	0.74	0.77
		Macro average	0.77	0.78	0.78
		Weighted average	0.77	0.78	0.77
GB	0.77	Unsatisfied employee	0.75	0.80	0.77
		Satisfied employee	0.79	0.74	0.76
		Macro average	0.77	0.77	0.77
		Weighted average	0.75	0.74	0.74
EGB	0.78	Unsatisfied employee	0.78	0.82	0.78
		Satisfied employee	0.80	0.77	0.77
		Macro average	0.78	0.79	0.78
		Weighted average	0.77	0.75	0.75
MLP	0.83	Unsatisfied employee	0.82	0.83	0.83
		Satisfied employee	0.82	0.82	0.70
		Macro average	0.82	0.83	0.82
		Weighted average	0.81	0.83	0.82

training vector, the performance of the classifiers is affected and accuracy is degraded with BoW features.

4.3 | Results using GloVe

GloVe method works on word embedding where equal importance is given to word–word occurrence in addition to the occurrence of words alone. However, research proves that TF-IDF performs better than GloVe.[49] Results shown in Table 15 are in conformity with the results from previous research.[50] Classification accuracy for machine learning classifiers, as well as the MLP, is decreased when the GloVe method is used for feature extraction. However, this decrease is associated with the choice of the dataset, preprocessing, and classifier, and it is possible that

TABLE 14 Accuracy, precision, recall, and F1 results for all the classifiers using Bag of Words features

Model	Accuracy		Precision	Recall	F1
RF	0.77	Unsatisfied employees	0.74	0.82	0.78
		Unsatisfied employees	0.80	0.72	0.76
		Macro average	0.77	0.77	0.77
		Weighted average	0.77	0.77	0.77
LR	0.78	Unsatisfied employees	0.77	0.80	0.79
		Unsatisfied employees	0.79	0.77	0.78
		Macro average	0.78	0.78	0.78
		Weighted average	0.78	0.78	0.78
SVC	0.77	Unsatisfied employees	0.78	0.80	0.78
		Unsatisfied employees	0.76	0.74	0.74
		Macro average	0.77	0.77	0.78
		Weighted average	0.77	0.76	0.76
GB	0.76	Unsatisfied employees	0.73	0.81	0.77
		Unsatisfied employees	0.79	0.72	0.75
		Macro average	0.76	0.76	0.76
		Weighted average	0.76	0.76	0.76
EGB	0.77	Unsatisfied employees	0.74	0.81	0.78
		Unsatisfied employees	0.80	0.73	0.75
		Macro average	0.76	0.76	0.77
		Weighted average	0.77	0.76	0.77
MLP	0.79	Unsatisfied employees	0.79	0.80	0.79
		Unsatisfied employees	0.79	0.79	0.79
		macro average	0.79	0.80	0.79
		Weighted average	0.79	0.79	0.79

with a different preprocessing and classifier GloVe proves to produce higher classification accuracy. Results show that the performance of RF is less affected by the choice of feature extraction method than that of other classifiers.

4.4 | Performance evaluation using cross-validation

For evaluating the efficacy of the proposed approach, experiments are performed using 10-fold cross-validation. Results are shown in Table 16. Results indicate that RF and SVC perform better than GBM and ADA when used with BoW and TF-IDF features and achieve an accuracy of 0.76. Conversely, GloVe features show better performance with RF and GBM with an accuracy of 0.72. MLP, on the other hand, achieves better accuracy than the selected classifiers

TABLE 15 Accuracy, precision, recall, and F1 results for all the classifiers with global vector approach

Model	Accuracy		Precision	Recall	F1
RF	0.74	Unsatisfied employees	0.71	0.82	0.76
		Unsatisfied employees	0.79	0.66	0.72
		Macro average	0.75	0.74	0.74
		Weighted average	0.75	0.74	0.74
LR	0.66	Unsatisfied employees	0.65	0.67	0.66
		Unsatisfied employees	0.66	0.65	0.65
		Macro average	0.66	0.66	0.66
		Weighted average	0.66	0.66	0.66
SVC	0.66	Unsatisfied employees	0.65	0.68	0.66
		Unsatisfied employees	0.67	0.64	0.65
		Macro average	0.66	0.66	0.66
		Weighted average	0.66	0.66	0.66
GB	0.72	Unsatisfied employees	0.70	0.80	0.75
		Unsatisfied employees	0.76	0.65	0.70
		Macro average	0.73	0.72	0.72
		Weighted average	0.73	0.72	0.72
EGB	0.74	Unsatisfied employees	0.72	0.81	0.76
		Unsatisfied employees	0.77	0.67	0.71
		Macro average	0.73	0.73	0.73
		Weighted average	0.74	0.72	0.73
MLP	0.76	Unsatisfied employees	0.77	0.77	0.76
		Unsatisfied employees	0.76	0.77	0.76
		Macro average	0.76	0.77	0.76
		Weighted average	0.77	0.77	0.76

with every feature. However, the highest accuracy of 0.80 is achieved when MLP is trained on TF-IDF features.

4.5 | Performance comparison with deep learning works

The performance of the proposed DNN is compared with two other similar deep learning models that are used for classification. Research[51] uses a bidirectional long short-term memory network (Bi-LSTM) is to classify text data into two classes of span or eligible message. Bi-LSTM is considered more powerful than simple LSTM because a single LSTM layer can, however, be too small in some examples and the use of two layers of LSTM gives better results in classification.[52] We took the model and applied it to the dataset that we used for the current study.

TABLE 16 Experimental results using 10-fold cross-validation

Model	Accuracy		
	BoW	TF-IDF	GloVe
RF	0.76 (± 0.21)	0.76 (± 0.20)	0.72 (± 0.16)
GBM	0.75 (± 0.18)	0.75 (± 0.18)	0.72 (± 0.16)
ADA	0.74 (± 0.21)	0.74 (± 0.18)	0.68 (± 0.14)
SVC	0.76 (± 0.18)	0.76 (± 0.19)	0.66 (± 0.10)
MLP	0. 79(± 0.16)	0.80 (± 0.15)	0.76 (± 0.15)

Abbreviations: BoW, bag of words; GloVe, global vector; TF-IDF, term frequency-inverse document frequency.

TABLE 17 Comparison of results for the proposed approach

Model	Accuracy		Precision	Recall	F1
Bi-LSTM[51]	0.77	Unsatisfied employee	0.78	0.78	0.77
		Satisfied employee	0.77	0.77	0.77
		Macro average	0.77	0.77	0.77
		Weighted average	0.77	0.77	0.77
CNN[53]	0.76	Unsatisfied employee	0.76	0.77	0.76
		Satisfied employee	0.76	0.75	0.76
		Macro average	0.76	0.76	0.76
		Weighted average	0.76	0.76	0.76
MLP-proposed	0.83	Unsatisfied employee	0.82	0.83	0.83
		Satisfied employee	0.82	0.82	0.70
		Macro average	0.82	0.83	0.82
		Weighted average	0.81	0.83	0.82

Abbreviations: Bi-LSTM, bidirectional long short-term memory network; CNN, convolution neural networks.

The second work[53] makes the use of CNN for classifying business reviews using word embedding. Higher accuracy is achieved in Reference 53, via three-fold cross-validation on Yelp 2017 dataset with word-based CNN. Table 17 shows the performance results of the proposed approach with other deep learning approaches.

Results show that the proposed approach performs better than that of Bi-LSTM and a word-based CNN approach for employee job satisfaction prediction using the review dataset. While the accuracy with Bi-LSTM and CNN is 0.77 and 0.76, respectively, the proposed approach can secure accuracy of 0.83 with the selected dataset. Similarly, other performance metrics like precision, recall, F1, and macro and weighted averages are higher with the proposed MLP than that of Bi-LSTM and word-based CNN.

When the inputs are assigned a class as in our case, MLP is more suitable for text classification. MLP is very flexible in dealing with the text data which is helpful to learn the mapping from input to output. The words in a document can be transformed into one long row of data and fed to an MLP. In our, study MLP performs better because MLP does not make any assumption

regarding the underlying probability density functions or other probabilistic information about the classes under consideration in comparison to other probability-based models. While the other deep learning models CNN and Bi-LSTM that require a larger dataset to learn better as compared to MLP. At the same time, MLP can show higher accuracy with relatively smaller datasets than that of CNN and Bi-LSTM.

5 | CONCLUSIONS

This study proposes the use of deep neural networks to perform text mining. Deep neural network (multilayered perceptron) architecture is devised to classify satisfied and unsatisfied employees utilizing their reviews about various companies including Google, Facebook, Amazon, Microsoft, and Apple. Four machine learning algorithms including RF, LR, SVC, and GB are investigated as well for their performance on the selected dataset. Three feature extraction methods are analyzed with these algorithms to study their suitability and impact on the classification accuracy. Results demonstrate that the proposed MLP outperforms other machine learning algorithms in terms of accuracy, as well as, precision, recall, and F1 score. MLP is flexible and more suitable for text classification that can show good performance even with relatively smaller datasets than that of LSTM and CNN. The performance comparison of the proposed MLP with Bi-LSTM and CNN models conforms to its superior performance.

Results show that TF-IDF feature extraction shows better results with machine learning algorithms than that of BoW and GloVe. Switching from TF-IDF to BoW or GloVe leads to degradation in the performance of machine learning algorithms, however, the performance of RF is less affected. LR shows better results with TF-IDF and BoW than that of other classifiers. Current results show the performance of RF, LR, SVC, GB, and MLP with the selected dataset and specified preprocessing only. Changing the dataset and preprocessing steps may change the results slightly, as the preprocessing steps affect the performance of various algorithms differently. We aim to conduct further experiments to study the impact of various preprocessing strategies on classification accuracy, as the accuracy is not solely dependent upon the tuning of classifiers' hyperparameters alone.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A2C1006159), MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2020-2016-0-00313) supervised by the IITP(Institute for Information & communications Technology Promotion), and the Brain Korea 21 Plus Program(No. 22A20130012814) funded by the National Research Foundation of Korea (NRF).

CONFLICT OF INTEREST

The authors declare no conflict of interest. The funding agency had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

DATA AVAILABILITY STATEMENT

Data derived from public domain resources

ORCID

Imran Ashraf https://orcid.org/0000-0002-8271-6496

REFERENCES

1. Zhou C, Sun C, Liu Z, Lau F. A C-LSTM neural network for text classification; 2015. arXiv preprint arXiv:1511.08630.
2. Tang D, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification. Paper presented at: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal; 2015:1422-1432.
3. Rustam F, Ashraf I, Mehmood A, Ullah S, Choi GS. Tweets classification on the base of sentiments for US airline companies. *Entropy*. 2019;21(11):1078.
4. Amin MZ, Nadeem N. Convolutional neural network: text classification model for open domain question answering system; 2018. arXiv preprint arXiv:1809.02479.
5. Lai S, Xu L, Liu K, Zhao J. Recurrent convolutional neural networks for text classification. Paper presented at: Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX; 2015.
6. Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. Paper presented at: Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada; 2013:6645-6649; IEEE.
7. Tsujii J, Hajic J. Proceedings of COLING 2014, 25th International Conference on Computational Linguistics: Technical Papers; 2014.
8. Li Y, Cao L, Zhu J, Luo J. Mining fashion outfit composition using an end-to-end deep learning approach on set data. *IEEE Trans Multimed*. 2017;19(8):1946-1955.
9. Liu J, Chang WC, Wu Y, Yang Y. Deep learning for extreme multi-label text classification. Paper presented at: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan; 2017:115-124.
10. Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: a survey. *Wiley Interdiscip Rev Data Mining Knowl Discov*. 2018;8(4):e1253.
11. Ajit P. Prediction of employee turnover in organizations using machine learning algorithms. *Algorithms*. 2016;4(5):C5.
12. Alao DABA, Adeyemo AB. Analyzing employee attrition using decision tree algorithms. *Comput Inf Syst Develop Inform Allied Res J*. 2013;4:17-28.
13. Costa A, Veloso A. Employee analytics through sentiment analysis. *SBBD 2015*; RSBrazil: Sociedade Brasileira de Computação, Porto Alegre; 2015:101-112.
14. Goldberg David, Zaman Nohel. Text analytics for employee dissatisfaction in human resources management; 2018.
15. Moniz Andy, Jong Franciska. Sentiment analysis and the impact of employee satisfaction on firm earnings. Paper presented at: Proceedings of the European Conference on Information Retrieval; 2014:519-527; Springer, New York, NY.
16. Bajpai R, Hazarika D, Singh K, Gorantla S, Cambria E, Zimmerman R. Aspect-sentiment embeddings for company profiling and employee opinion mining; 2019. arXiv preprint arXiv:1902.08342.
17. Jantan H, Hamdan AR, Othman ZA. Towards applying data mining techniques for talent management; 2011.
18. Kowsari K, Brown DE, Heidarysafa M, Meimandi KJ, Gerber MS, Barnes LE. Hdltex: hierarchical deep learning for text classification. Paper presented at: Proceedings of the 2017 16th IEEE international conference on Machine Learning and Applications (ICMLA) 2017 December 18, Cancun, Mexico; 2017:364-371; IEEE.
19. Kaggle Employees reviews dataset.
20. Nejad A, Mohammad B, Hashemi BSM, Sayahi CA, Kiaeimehr DB. Feature selection techniques for text classification. *Int J Comput Sci Netw Solut*. 2014;2(1):90-94.
21. Zhang W, Yoshida T, Tang X. A comparative study of TF* IDF, LSI and multi-words for text classification. *Expert Syst Appl*. 2011;38(3):2758-2765.
22. Liu L, Tang L, Dong W, Yao S, Zhou W. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*. 2016;5(1):1608.

23. Komer B, Bergstra J, Eliasmith C. Hyperopt-sklearn: Automatic Hyperparameter Configuration For Scikit-Learn. *ICML workshop on AutoML*. 9. Austin, TX; Citeseer: 2014;50.
24. Pennington J, Socher R, Manning CD. Glove: global vectors for word representation. Paper presented at: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar; October 2014:1532-1543.
25. Breiman L. Bagging predictors. *Mach Learn*. 1996;24(2):123-140.
26. Quinlan J. *Ross. C4. 5: Programs for Machine Learning*. Amsterdam, Netherlands: Elsevier; 2014.
27. Bennett KP, Campbell C. Support vector machines: hype or hallelujah? *ACM SIGKDD Explor Newslett*. 2000;2(2):1-13.
28. Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann Stat*. 2001;29(5):1189-1232.
29. Mohan A, Chen Z, Weinberger K. Web-search ranking with initialized gradient boosted regression trees. Paper presented at: Proceedings of the Learning to Rank Challenge, Fort Lauderdale, FL; 2011:77-89.
30. Zhang L, Zhan C. Machine learning in rock facies classification: an application of XGBoost. Paper presented at: International Geophysical Conference; April 17-20, 2017:1371-1374; Society of Exploration Geophysicists and Chinese Petroleum Society; Qingdao, China.
31. Zhang F, Du B, Zhang L. Scene classification via a gradient boosting random convolutional network framework. *IEEE Trans Geosci Remote Sens*. 2015;54(3):1793-1802.
32. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science*. 2006;313(5786):504-507.
33. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436-444.
34. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier. *Neural Netw*. 2011;15:315-323.
35. Liu P, Qiu X, Huang X. Recurrent neural network for text classification with multi-task learning; 2016. arXiv preprint arXiv:1605.05101.
36. Wermter S. Neural network agents for learning semantic text classification. *Inf Retrieval*. 2000;3(2): 87-103.
37. Wiener E, Pedersen JO, Weigend AS. *A Neural Network Approach to Topic Spotting*. Vol 332. Las Vegas, NV: University of Nevada; 1995.
38. He Y, Li J, Song Y, He M, Peng H. Time-evolving text classification with deep neural networks. *IJCAI*. 2018;18:2241-2247.
39. Amajd M, Kaimuldenov Z, Voronkov I. Text classification with deep neural networks. *International Conference on Actual Problems of System and Software Engineering (APSSE), Moscow, Russia*. 2017;364-370.
40. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;15(1):1929-1958.
41. Khalid M, Ashraf I, Mehmood A, Ullah S, Ahmad M, Choi GS. GBSVM: sentiment classification from unstructured reviews using ensemble classifier. *Appl Sci*. 2020;10(8):2788.
42. Loper E, Bird S. NLTK: the natural language toolkit; 2002. arXiv preprint cs/0205028.
43. Kadhim AI. An evaluation of preprocessing techniques for text classification. *Int J Comput Sci Inf Secur (IJCSIS)*. 2018;16(6):22-32.
44. Karaa WBA. A new stemmer to improve information retrieval. *Int J Netw Secur Appl*. 2013;5(4):143.
45. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011;12:2825-2830.
46. Gulli A, Pal S. *Deep Learning with Keras*. Birmingham, UK: Packt Publishing Ltd; 2017.
47. Python Unified framework for word embeddings compatible with scikit-learn Pipeline.
48. Su MC, Jean WF, Chang HT. A static hand gesture recognition system using a composite neural network. Paper presented at: Proceedings of IEEE 5th International Fuzzy Systems, New Orleans, LA; September 11, 1996:786-792; IEEE.
49. Eklund Martin. Comparing feature extraction methods and effects of pre-processing methods for multi-label classification of textual data, New Orleans, LA; 2018.
50. Bharadwaj P, Shao Z. Fake news detection with semantic features and text mining. *Int J Natural Lang Comput*. 2019;8(3).
51. Nowak J, Taspinar A, Scherer R. LSTM recurrent neural networks for short text and sentiment classification. Paper presented at: Proceedings of the International Conference on Artificial Intelligence and Soft Computing 2017 June 11; 2017:553-562; Springer, Cham, Switzerland.

52. Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw*. 2005;18(5-6):602-610.
53. Salinca A. Convolutional neural networks for sentiment classification on business reviews; 2017. arXiv preprint arXiv:1710.05978.

How to cite this article: Rustam F, Ashraf I, Shafique R, Mehmood A, Ullah S, Sang Choi G. Review prognosis system to predict employees job satisfaction using deep neural network. *Computational Intelligence*. 2021;37:924–950. https://doi.org/10.1111/coin.12440