TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 21, No. 5, October 2023, pp. 1084~1101
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v21i5.24046

Amazon products reviews classification based on machine learning, deep learning methods and BERT

Saman Iftikhar[1], Bandar Alluhaybi[1], Mohammed Suliman[1], Ammar Saeed[2], Kiran Fatima[3]
[1]Faculty of Computer Studies, Arab Open University, Riyadh, Saudi Arabia
[2]Department of Computer Science, Commission on Science and Technology for Sustainable Development in the South (COMSATS), Islamabad, Wah Campus, Wah Cantt, Pakistan
[3]Technical and Further Education (TAFE), New South Wales, Australia

Article Info

Article history:

Received May 23, 2022
Revised Dec 31, 2022
Accepted Feb 16, 2023

Keywords:

Deep learning
Feature extraction
Machine learning
Sentiment analysis
Transformer technique

ABSTRACT

In recent times, the trend of online shopping through e-commerce stores and websites has grown to a huge extent. Whenever a product is purchased on an e-commerce platform, people leave their reviews about the product. These reviews are very helpful for the store owners and the product's manufacturers for the betterment of their work process as well as product quality. An automated system is proposed in this work that operates on two datasets D1 and D2 obtained from Amazon. After certain preprocessing steps, N-gram and word embedding-based features are extracted using term frequency-inverse document frequency (TF-IDF), bag of words (BoW) and global vectors (GloVe), and Word2vec, respectively. Four machine learning (ML) models support vector machines (SVM), logistic regression (RF), logistic regression (LR), multinomial Naïve Bayes (MNB), two deep learning (DL) models convolutional neural network (CNN), long-short term memory (LSTM), and standalone bidirectional encoder representations (BERT) are used to classify reviews as either positive or negative. The results obtained by the standard ML, DL models and BERT are evaluated using certain performance evaluation measures. BERT turns out to be the best-performing model in the case of D1 with an accuracy of 90% on features derived by word embedding models while the CNN provides the best accuracy of 97% upon word embedding features in the case of D2. The proposed model shows better overall performance on D2 as compared to D1.

Corresponding Author:

Saman Iftikhar
Faculty of Computer Studies, Arab Open University
Riyadh, Saudi Arabia
Email: s.iftikhar@aou.edu.sa

1. INTRODUCTION

In the twenty-first century, the revolutionary growth in technology has changed the perspective by which things are done around the world. The comfortable availability of the internet and access to almost anywhere in the world has encouraged people to rely more on online services to fulfill their daily demands such as shopping, hiring, selling, staying updated with news, and using social media platforms. Online shopping is one of the most prominent domains that has seen tremendous uprising trends with this internet revolution. Several well-established e-commerce websites are already ruling the internet and more are being launched on daily basis [1]. People around the world have started relying more on those e-commerce sites to purchase groceries, technical gadgets, and almost anything they can demand. The reason for using online platforms to purchase goods is the ease of access as people do not have to go through the tiresome way of exploring markets to find what they are looking for, rather they can sit in the comfort of their homes and easily

explore, compare, browse and purchase the items of their choice and can get them at their doorstep [2]. Amazon, the US-based leading e-commerce platform with a brand value of 684 billion USD has registered sales of 386 billion USD in 2020 which mostly included shopping for electronic and technological products, and the net sales revenue generated by it until now is 469 billion USD [3]. Another online business-to-business (B2B) marketplace Alibaba processed 538,000 transactions each second during 2020 and generated sales revenue of 717 billion Yuan by March 2020. Alibaba is also expected to achieve net sales of 3.62 trillion USD in China by 2025 [4]. Flipkart made a revenue of 433 billion INR in a year which accounted for an increase of 25% as compared to 2020 [5]. The sales revenue generated by the same platforms during the year 2015 shows that Amazon, Alibaba, and Flipkart sold products worth 75.6 USD, 76.2 million Yuan, and 12.5 billion USD. When the sale trends of these giant online stores are compared from 2015 to 2020, it demonstrates a tremendous increase in online shopping trends, therefore, leading to massive revenue production.

The statistics indicate the increasing trends of online shopping which are overtaking conventional means. As far as e-commerce platforms and other online stores are concerned, it becomes necessary for them to meet customer demands, correctly interpret them, listen to their complaints, and maintain quality to cement their place and establish themselves. All authoritative e-commerce platforms use a review and rating system through which customers and buyers can leave a review post product purchase and retrieval. This provides a real-time evaluation of several aspects of online shopping including product quality feedback, customer satisfaction with delivery service and procedure as well as suggestions for improvement [6]. The successful standing giants use these evaluations and feedback for their constant improvement thus enhancing their customer's experience which in turn increases their sales and leads to augmented revenue origination. Since the number of reviews and feedback are real-time and are performed by millions of users depending upon the diversity of the concerned platform, it becomes very difficult to go through all of them. Most of the time, this phase is performed manually, and it involves an inspection or quality assurance team that goes through some of the feedback to formalize a report stating complaints, improvements, and recommendations. Due to the involvement of physical personnel, it becomes a time-consuming, less accurate, and costly task. Even after investing time and money, all reviews cannot be covered because they are in bulk and add up quickly in real-time [7]. This creates an urge for the development of an automated system that can automatically take those reviews as input, perform text-based analysis also known as sentiment analysis (SA) over them, deduct the meanings inducted inside them, and classify them as good, and bad, or neutral. This can help e-commerce platforms to be more productive in terms of customer feedback understanding and can timely adopt the measures and methods they are currently lacking. An automated system can also rectify the problem of not being able to cover all the reviews as it can perform sentiment analysis and generate results within seconds therefore it can go through the whole corpus within no time.

Evolutionary research in the field of artificial intelligence has made it very easy to develop such an automated system. The approaches of machine learning (ML) and deep learning (DL) are being widely used to formulate self-working systems that are implemented in mainstream real-world businesses and companies to increase productivity, reduce manual workforce, increase accuracy and maintain brisk speed [8]. Moreover, the use of word embedding methods such as term frequency-inverse document frequency (TF-IDF), N-gram-based feature extraction techniques, transformer-based methods, and topic modeling approaches are being extensively used in the SA tasks in the domain of data analytics and engineering [9]. Keeping in view the aspects, the proposed work starts with the acquisition of two datasets from Amazon. The acquired datasets contain user reviews for cell phones and other products and are randomly chosen. Certain preprocessing steps are employed on the data to cleanse it and make it ready for the upcoming phases. The reviews are labeled in positive and negative classes based on their star rating out of 5 stars. For feature extraction, N-gram methods TF-IDF, bag of words (BoW), and word embedding techniques global vectors (GloVe), and Word2vec are implemented. Bidirectional encoder representations (BERT) is also adopted to derive deep insights from data based on its transformation layers. Certain ML classifiers including support vector machine (SVM), Naïve Bayes (NB), random forest (RF), logistic regression (LR), and DL models including a custom convolutional neural network (CNN) and long-short term memory (LSTM) are employed for classification. The results are evaluated based on certain evaluation measures. The main contributions of proposed work are as:

a) Two datasets D1 and D2 have been acquired from Amazon. D1 is a collection of cell phone reviews, while D2 is an amalgamation of D1 and random product reviews. Datasets are prepared and engineered using various SA techniques.
b) The proposed work used both textual features extracted by TF-IDF, BoW and deep features extracted by Word2Vec, Glove, and classified them using ML and DL models, respectively.
c) Transformer-based model BERT is also implemented to classify features and later on, its results are compared with those of ML and DL models.
d) ML classifiers are used to classify textual features, DLs are used to classify deep features, and BERTs are used to classify overall standalone features. BERT performs better on D1 while DL-based CNN proves to be better in case of D2.

The rest of the paper is formatted as: section 2 gives an overview of the strategies employed in prior publications for the analysis and classification of Amazon product reviews. The proposed methodology of this research work is discussed in section 3. Section 4 lists all the experiments done, together with their findings and performance evaluations. Section 5 discusses the findings, and section 6 concludes with the conclusion of the proposed work.

2. RELATED WORK

Several studies have been performed for SA of reviews posted on e-commerce platforms and their classification. We discussed some of them in the following subsections. We looked at their opted methodologies and the results they achieved.

2.1. ML-based techniques for reviews classification

Daniel and Meena [10] proposed a hybrid framework composed of lexicon methods and ML techniques for the SA of Amazon data that contains 48,500 reviews performed on various product categories such as electronics, furniture, kid's toys, and camera accessories. Each review contains information regarding the reviewer, date, time, location, and review itself. The data is cleaned through several steps including URL removal, conversion into a single case, tokenization, and stop word rectification. The data is then manually labeled into positive and negative reviews based on their rating out of 5 using VADER. Feature extraction is performed using fastText and GloVe attribute embedding methods. To minimize the attribute dimensionality and maintain the model's speed, a tunicate swarm intelligence (TSA) based feature selection approach is introduced over the extracted features. Finally, the result is given to several ML classifiers localized support vector machine (LSVM), DT, NB, and K-nearest neighbor (KNN). The proposed model achieves a maximum accuracy of 93% over the furniture dataset with LSVM classifier without TSA and 91% with TSA whereas the execution time is also reduced to 43% when TSA is implemented. Li *et al*. [11] obtained a dataset based on 10,261 reviews performed on musical instruments from Amazon. Analphabetic sign removal, lowercase conversion, and removal of unnecessary words are among the preprocessing steps performed for data cleansing. The frequency of all the words is calculated using TF-IDF and a term set is formulated based on 160 terms with the highest frequency. WordNet and SentiWordNet are utilized as lexicons to map each word with a corresponding weight in the feature space. To generate word embeddings, BERT is implemented where the included words are vectorized. Bi-directional LSTM along with an attention scheme is used for feature classification where the model achieves an accuracy of 96% after the adjustment of the loss rate. The proposed model performs better when compared with other baseline methods using several evaluation metrics. Shrestha and Nasoz [12] retrieved 3.5 million reviews conducted on random product categories from Amazon. The dataset contains all information about the review, user, time, and date and is compared with the 5 stars baseline. Preprocessing steps such as hyperlinks, unwanted space, stop words, informal words, and punctuation removal are performed on the corpus obtained. For data vectorization and semantic information deduction, the paragraph vectors (PV) are utilized that perform next-word prediction and provide context to the sample paragraphs. Both memory distribution (PV-DM) and bag of words distribution (PV-DBOW) versions are implemented in the proposed work. After converting text-based reviews to dimensional vectors, the input is given to gated recurrent unit (GRU) for the derivation of embedding information. Finally, the output is fitted to the SVM classifier that achieves a maximum accuracy of 81.29% and 81.82% on embedding derived for reviews and products, respectively.

Elmurngi and Gherbi [13] obtained the reviews performed on clothing, shoes, and jewelry categories on the Amazon platform. The dataset is divided into 5 classes based on its star rating while the empty rows are rectified. String to word vector (STWV) is used as the filtration method that performs stop words removal and tokenization. For feature selection, a combination of best first, subset evaluation, and genetic search is used. The final stage of classification is performed with the help of NB, DT, LR, and SVM classifiers upon the selected features. The LR classifier maintains accuracies of 81.61%, 80.09%, and 60.72% on clothing, shoes, and jewelry datasets, respectively, and stands out among other classifiers. Rao and Sindhu [14] performed sarcasm analysis on product reviews from Amazon. A random dataset comprising product reviews is retrieved from Amazon that contains information about the brand, product identity, and some useful information about the reviewer. Each review is treated as a separate document and its labeling is performed based on rating polarity. The preprocessing includes tokenization, stemming, lemmatization, and labeling based on polarity. Feature extraction is performed using TF-IDF and N-gram methods (uni, bi, and trigrams). The extracted features are finally provided to certain ML classifiers SVM, KNN, and RF where the SVM classifier achieves a higher accuracy rate of 67.58% as compared to RF (62.34%) and KNN (61.08%). Wassan *et al.* [15] formulated a data collection comprising reviews from 28,000 customers for 60 products on Amazon. The dataset is categorized into positive, negative, and neutral reviews based on 5-star rating polarity. Stop word removal and other data cleansing steps are performed using the natural language Toolkit (NLTK) and text blob libraries.

Features are extracted using the bag of words method and computation matrices such as recall, and f-measurement are adopted along with cross-validation to map the reviews in their respective categories.

2.2. DL-based techniques for reviews classification

Hawlader *et al*. [16] collected a pre-labeled dataset from Amazon containing product reviews. Preprocessing steps including tokenization, stop words removal, stemming and tagging are performed on the data. The rating parameter is binarized to construct a premise for the classification. A new feature is introduced into the dataset that categorizes the review into positive or negative classes. Search the classification is binary so the neutral reviews are discarded given the dataset with 24000 negative and 70000 positive assessments. Feature extraction is performed using BoW, TF-IDF, and Word2vec techniques all implemented individually on the dataset. Several ML classifiers such as SVM, NB, LR, DT, RF as well as DL-based MLP are utilized to map data into their concerning categories. The proposed model achieves the maximum accuracy of 91% via MLP for TF-IDF features, 92% via MLP for BoW features, and 71% via MLP for Word2vec features. This indicates that MLP outperforms ML models in terms of performance for the currently used dataset. Alharbi *et al.* [17] obtained a publicly available dataset containing 400,000 reviews written on sold mobile phones on Amazon and passed through several preprocessing steps which include spelling correction, tokenization, stop words removal, punctuation removal, and lemmatization. The factorization of data is performed by using fastText, GloVe, and Word2vec methods. Data are classified using a custom tuned recurrent neural network (RNN) that takes as input the combination of embeddings extracted in the previous phase of feature engineering, passes it through several normalization layers, and performs the prediction using the Softmax layer. Two variants of RNN are utilized in this work which includes LSTM-RNN and general regression neural network (GRNN). The proposed model achieves maximum accuracies of 88.38% and 87.25% on GLSTM and update gate RNN (UGRNN) based frameworks, respectively. Dadhich and Thankachan [18] formulated a system that takes the user reviews and comments obtained from Flipkart and Amazon as input, applies certain data cleansing steps, performs feature extraction using the SentiWordNet algorithm, and classifies the data with the help of ML classifiers NB, LR, RF, and KNN. The system achieves an overall accuracy of 91% with the Flipkart dataset. Norinder and Norinder [19] collected an imbalanced reviews dataset from Amazon for five product and their 12 categories along with their star ratings. Non-alpha numeric values are removed from the data together with stop words removal and tokenization using the NLTK toolkit. A deep architecture DNN is used for data classification that comprises embedding, LSTM, and output layers. Conformal and mondrian predictions are used for model calibration using the original data. The model shows accuracy rates ranging from 89.8% to 92.2% with an error rate of 12.5%.

Bhuvaneshwari *et al.* [20] employed data from Amazon containing 19,988 reviews on various products where 15,000 reviews appear to have less than 100 words. 15,990 reviews are maintained for training and 3997 for testing. Some mandatory preprocessing steps such as URL, stop words, punctuation, and connecting words removal are performed along with tokenization, stemming, and spelling correction. The skip-gram-based Word2vec model is utilized to device word vectors from the prepared corpus. A Bi-LSTM model is formulated that contains 100 parameters united and self-attention functionality. It is based on ReLU, CNN, pool, ully connected, and classification layers. It used RMSProp as an optimizer and entropy as a loss function. The model achieves an accuracy of over 85% when compared with standard CNN, Bi-directional gated recurrent unit (BGRU), BCNN, and NB models along with decent training time. Nandal *et al.* [21] utilized a web crawler to retrieve data on user reviews on Amazon products. The data contains information about the user, rating, review, and URL. Vectorization, stop word removal, parts of speech (POS) tagging, stemming and lemmatization are among the preprocessing steps employed. Aspect aggregation is employed to derive key aspects from the data. The derived aspects and their polarities are given to the SVM classifier. Mean square error (MSE) is used as a loss rate evaluator where the bipolar inputs show the minimum loss rate of 0.4% and the model's validation accuracy reaches 86%. Dey *et al.* [22] utilized a dataset from Amazon based on 147,000 reviews of various books. Tokenization, stop word removal, and re-filling the missing values are some of the key preprocessing steps employed. The feature extraction is performed using TF-IDF and classification is performed using the SVM and NB classifiers. The LSVM classifier shows better accuracy (84%) as compared to NB (82%) Zhao *et al.* [23] proposed a model for the SA of reviews given on e-commerce platforms including Amazon, eBay, and Taobao. Data is passed through tokenization, lemmatization, and snowball stemming phases followed by a term weighting phase based on least term frequency. The earthworm and bat feature selection algorithms are used to reduce feature dimensionality and increase model briskness. The feature extraction is performed using Word2vec and TF methods. The LSIBA-ENN model achieves better recall and precision rates when compared with standalone NB, SVM, and ENN classifiers for both TF and Word2vec features. Mukherjee *et al.* [24] obtained 82,000 reviews posted on sold cellphones on Amazon and performed standard preprocessing steps on it such as stop words removal, tokenization, URL, and punctuation removal. TF-IDF is used for feature extraction whereas ML classifiers NB, SVM, and DL models RNN and artificial neural networks (ANN) are used to classify data in their respective classes. The ANN model along with negation yields the best accuracy rate of 95.67% in competition with other employed

models. ANN also performs better in terms of other performance metrics. Sivakumar and Uyyala [25] implemented fine-tuned LSTM based on fuzzy logic over the amazon reviews for cell phones (ACPR), Amazon reviews for video games (AVGR), and consumer random reviews for Amazon products (CRAP) datasets. They performed tokenization, spelling correction, normalization, lemmatization, and long-sentence splitting over the datasets and passed it through BoW-based word embedding. The proposed LSTM is tested on 100–500 epoch settings, batch sizes of 4–32, Adam optimizer, and dropout layer where it achieves an overall accuracy rate of 96.93%, 83.82 and 90.92% for ACPR, AVGR, and CRAP datasets, respectively.

2.3. Opinion-based reviews classification using ML and DL models

Huang [26] collected the data from e-commerce, social, and comment websites and applied certain preprocessing steps including frequent term filtration, rules mining and dictionary formation. After the mining phase, the feature vectors are generated against each acquired comment sentiment by identifying the comment's polarity, considering of formulated dictionary, and computing the scores of effectiveness. Finally, the data is analyzed for any trace of risk or fraud by merging all the computed indices with feature vectors and mining the anomalies. The analysis phase used methods of syntactic analysis, true comments are detected based on their unanimity, and false comments are detected based on the deviation of their indices from those of the original comment's indices. The proposed schema reached a maximum credibility (accuracy) score of 83.14% on the utilized e-commerce product datasets. Vanaja and Belwal [27] performed SA on customer opinions and feedback against Amazon products after identifying and tagging parts of speech from the formulated dataset. The Apriori algorithm is used to perform feature derivation. It is applied to the dataset to extract commonly used elements and is based on association rules, which are employed in databases to establish relationships between various features. Feature simplification is followed by the extraction of opinion words. Opinion words are a group of adjectives that describe the features of the product. Finally, the phase of classification is performed to categorize the opinions into positive, negative, and neutral classes. SVM and NB are amongst the ML-based classifiers used in this case that are integrated with SentiWordNet for polarity score computation. Results indicate that NB achieves the highest accuracy score of 90.423% as compared to SVM's score of 83.423%.

Alzahrani *et al.* [28] utilized the Amazon technology products reviews dataset and implemented mandatory preprocessing steps such as stop words removal, tokenization, and speech tagging on it. The opinion lexicon is utilized to compute sentiment scores. The integration of CNN and LSTM is formulated to classify reviews into positive or negative classes. The standalone LSTM achieved an accuracy of 91.03% while the integrated CNN-LSTM framework achieved an accuracy of 94%. Dadhich and Thankachan [18] performed the SA and classification of product reviews provided on Amazon and Flipkart. They implemented certain data preparation steps and generated a knowledge tree. Natural language toolkit is employed to generate a word dictionary, compute word information, and extract textual features. Five ML classifiers are used to categorize comments and reviews into their respective polarity classes including NB, LR, SentiWordNet, KNN, and RF. The proposed model achieved an accuracy of 91.13% on a total of 79655 reviews. They integrated several DL models including robustly optimized BERT (RoBERT), LSTM, GRU, and bidirectional LSTM (BiLSTM) to perform SA of internet movie database (IMDb), American airline and Sentiment140 data corpora. Case conversion and punctuation corrections are amongst some of the preprocessing steps performed on the utilized datasets. Glove-based word embedding is applied to the data to perform data augmentation and feature extraction. Experiments are conducted with various utilized DL-models' combinations where RoBERT-LSTM models yielded 91.37% accuracy, RoBERT-BiLSTM model yielded 91.21% accuracy, RoBERT-LSTM model yielded 91.37% accuracy and RoBERT-GRU achieved 91.52% accuracy. Tan *et al.* [29] derived a large amount of textual data from various web sources, blogs, networks, and search mediums during the COVID tenure. Language, geographical, time, and creator filtering are performed for opinion mining. Noise filtering, tokenization, and normalization are performed for aspect mining. Opinion classification is then performed using supervised, semi and non-supervised ML algorithms. The proposed work shows that ML models show promising accuracy in the classification of opinions within sentiments. Qureshi *et al.* [30] applied SA to Roman Urdu upon reviews dataset collected from YouTube. The reviews were based on comments given against different Pakistani and Indian songs. After applying mandatory preprocessing steps, the data from different files are integrated into a single comprehensive file and annotated as either positive or negative by the language experts. Finally, the classification is carried out using several ML algorithms such as NB, SVM, LR, DT, KNN, and CNN. Among all the applied models, LR turned out to be the best-performing model with an accuracy of 92.25% while the CNN performed worst with an accuracy of just 66.54% when applied to a total of 24,000 reviews containing half divisions of positive and negative ones.

Taking the advancements in AI to the next level, Cambria *et al*. [31] suggested a commonsense-based neuromyotonic framework named SenticNet 7, that seeks to address these problems. They developed reliable symbolic representations that transform natural language into a kind of protolanguage and, as a result, extract polarity from text in a fully interpretable and comprehensible way using unsupervised and reproducible sub

symbolic techniques like auto-regressive language models and kernel methods. The formulated model utilizes techniques of primitive discovery, affective similarity, primitive pairing and setting sentiment pavements for the learned predicates. The proposed model achieves an average accuracy of over 80% when tested against 10 benchmark SA datasets. For aspect-based sentiment analysis, Zhao and Yu [32] suggested the BERT model of knowledge-enhanced language representation. By incorporating sentiment domain knowledge into the language representation model, which extricates the vectorization formats of mappings included in the knowledge graph and predicates in the text in a consistent vector space, their proposal makes use of the additional information from a sentiment knowledge graph. Additionally, by introducing outside information into the language representation model to make up for the sparse training data, the model can perform better with less training data. The model may therefore deliver comprehensive and understandable findings for aspect-based sentiment analysis.

After going through some of the mentioned literature work, it is noticed that most works have either used standalone N-gram methods or deep word embedding techniques in their works but very few have used a combination of both for the classification of Amazon product reviews. Also, there has not been any elaborative comparison of core natural language processing (NLP) transformer-based methods with the standard ML and DL models. Taking the lead from all these aspects, the presented study aims to provide an elaborative comparison of transformer-based methods with standard ML and DL models. Also, the proposed work looks to implement multiple feature extraction methodologies including both N-gram and word embedding models for better conceptualization.

3. METHOD

A framework for the classification of user-posted reviews on Amazon products is proposed in this work. Two publicly available Amazon product datasets are obtained where one contains reviews for cell phones and the other contains consumer reviews for random products. The datasets are cleaned and preprocessed with steps such as stop words and null values removal, data balancing, tokenization, and lemmatization. The dataset reviews are labeled as positive and negative based on their star ratings out of 5 stars. Features are extracted using N-gram methods TF-IDF, BoW, and word embedding models GloVe, and Word2vec. The extricated features are classified using several ML algorithms including SVM, NB, RF, LR, DL-based CNN, and LSTM. For performance comparison of model performance with current data, a transformer-based BERT model is also formulated in this work that operates directly on preprocessed data and performs classification in parallel with the rest of the procedure. The results of standard ML and DL approaches with BERT are compared and analyzed with the help of certain evaluation metrics. Figure 1 shows the architectural framework for the proposed model. All mentioned steps are discussed in their specific sections.

3.1. Data acquisition and pre-processing

Two publicly available datasets from Kaggle are obtained for this work. Both datasets are based on reviews posted on Amazon products. The first dataset (D1) contains reviews posted for locked and unlocked cell phones belonging to ten brands such as ASUS, Google, Xiaomi, Sony, Samsung, and others. It contains important information such as title, brand, URL, reviews, ratings, and price of cell phones. The second dataset (D2) is the combination of the cell phone reviews dataset (D1) with another dataset comprising reviews posted by 34,000 customers against random Amazon products including electronic appliances, gadgets, and products mostly. It contains main data attributes such as brand, category, manufacturer, reviews, date of review posting, and date of it being seen. Table 1 provides an overview of data statistics after exploratory data analysis (EDA). EDA on both datasets is performed after using preprocessing techniques including tokenization and lemmatization.

Table 1. Dataset statistics

Attribute	Value
Dataset 1 (D1)	67986
D1-positive sentiments	51328
D1-negative sentiments	16658
D1-word count	6239978
D1-character count	24727439
D1-sentence count	102656
Dataset 2 (D2)	130978
D2-positive sentiments	109189
D2-negative sentiments	21789
D2-word count	11079466
D2-character count	43865611
D2-sentence count	218378

Both the D1 and D2 are highly unbalanced in their natural state which can cause the model to be biased in its predictions. To make sure both datasets are equally balanced, D1 is balanced using the approach of under-sampling while D2 is balanced using over-sampling. Since the datasets are based on real-world reviews of people around the world and they belong to various product categories, they contain noise and other artifacts that may cause performance deification. Therefore, these problems are addressed through the implication of several preprocessing steps including stop words removal, balancing the data classes, tokenization, and lemmatization. Apart from applying the mentioned preprocessing techniques, data labeling is also performed as per the review score out of 5. Reviews with a rating greater than 3 are considered positive while those with a rating below it, are considered negative and are labeled accordingly.

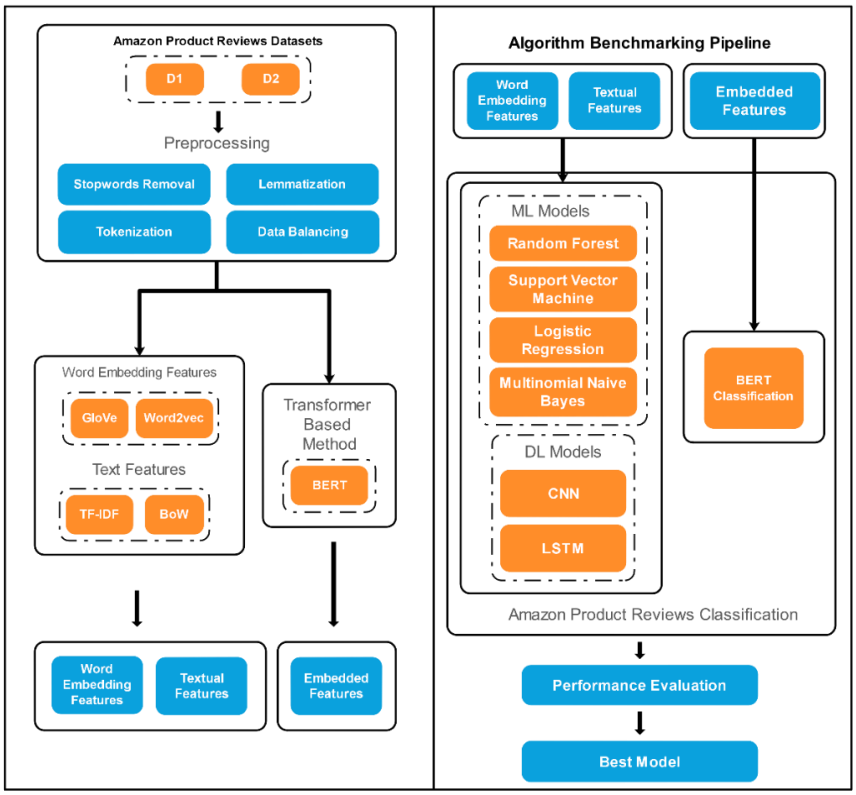


Figure 1. Architecture for the proposed Amazon product reviews analysis and classification

3.2. Feature extraction

The model cannot work on or categorize the data in its regular textual form after data preparation and balancing, which is why it must be translated into mathematical and vector format so that the ML and DL algorithms can interpret it. The vectorial data retrieved from the text is then fed into the ML and DL models as features. A complete representation of the words in the corpus must be extracted, and there are several methods for doing so. Deep and textual feature extrication approaches such as word embedding, and N-gram methods are used to extract the features. GloVe from Sandford NLP and Word2vec from Google news vectors are utilized as pre-trained word embeddings in the proposed study. BoW and TF-IDF are used to extract N-gram-based features. These approaches are addressed below with the planned study.

3.2.1. Textual features

Any series of word tokens in each data is referred to as an N-gram, with $n = 1$ denoting a unigram, $n = 2$ denoting a bigram, and so on. An N-gram model can calculate and forecast the likelihood of N-grams in a data corpus. Such models are effective in text classification problems where the number of particular terms contained in the vocabulary from the corpus must be counted [26]. The TF-IDF is a metric that assesses how closely a word in a catalog corresponds to its meaning or mood. It works by taking the frequency of terms in a document and multiplying it by the inverse frequency of words that appear in several texts regularly [27]. The frequency of documents in a corpus is calculated using TF-capabilities IDF [29], which is represented in (1).

$$w_{m,n} = f_{m,n}^{t} \, x \log(\frac{K}{f_m}) \quad (1)$$

Where, $w_{m,n}$ indicates the weight of data points m and n, $f_{m,n}^{t}$ is used to compute the occurrence frequency of target point m within reference point n, K shows the total number of included documents within the compilation, $log(\frac{K}{f_m})$ is used for log computation of target data point m in all the dataset documents. BoW may also be used to extract valuable qualities from textual material that has to be categorized. It operates based on a predetermined vocabulary and searches for the frequency of certain terms in the document in question using that vocabulary. The model simply cares if known terms appear in the document, not where they appear, and it generates a histogram of such words within the data that can be readily fed to classifiers [30]. The (2) is used by BoW to build bags containing words.

$$d_m = \sum_{m=1}^{K} w_m^n \; x \; w_m \quad (2)$$

Where d_b indicates the document in which target data point m is present. w_m^n assigns the weights to the target point m concerning reference point n. w_m is the weight of target point m which in this scenario is our point of concern. Both TF-IDF and BoW are employed to derive features from the preprocessed dataset in the proposed study. A collection of four ML classifiers is used to evaluate and classify the retrieved features from both models.

3.2.2. Word embeddings

Word embedding [29] is a technique for converting and representing textual data made up of words into a vector and mathematical form. There are several models available for this purpose, however, we used the pre-trained GloVe from Stanford NLP [32] and Wor2vec [33] from Google news vectors in this study. GloVe is an unsupervised learning technique that uses the global word co-occurrence matrix to extract word embeddings from an input data corpus. When applied to any data, it directly obtains information about the words occurring frequently in that data and maps the words into vector spaces [30]. It is trained on global statistics of words included in a large corpus compiled from online sources and when applied to any data, it obtains information about the words occurring frequently in that data and maps the words into vector spaces. It has been frequently used to derive features and pass them on to classification models in text classification challenges. As (3) shows, it is based on the bilinear (LBL) model, which operates on the idea of weighted least squares [31].

$$w_m \, . \, w_n = \log prob(m|n) \quad (3)$$

Here, w_m, w_n is the weightage of target and reference data points and $prob(m|n)$ is their probability of occurrence. The working logic behind GloVe is represented in (4).

$$loss = \sum_{m,n=1}^{K} f(X_{m,n})(w_m^t w_n - \log X_{m,n})^2 \quad (4)$$

Where, $f(X_{m,n})$ is the least-squares mapping function between the data points and $w_m^t w_n$ shows the weights for points concerning time t. Word2vec is a word embedding approach that uses the skip-gram method to provide this capability and is based on shallow deep networks. Based on the frequency of documents and their co-occurrence matrix, it builds vectors of textual data included in the corpus. The skip-gram approach is used by Word2vec to execute computations, as shown in (5).

$$\frac{1}{T} \sum_{pos=1}^{K} \sum_{-1 \le m \le 1, m \ne 0} \log prob(wd_{pos+1}|wd_m) \quad (5)$$

Where K is the size of the corpus, *pos* is the position of a word wd_m in data K, $\log prob(wd_{m+1}|wd_m)$ is the log of wd_m as it keeps on updating its positions and locales within the document [34]. The preprocessed data is also sent to both the GloVe and Word2vec models in the proposed study, and the features created by them are then given a customized CNN as well as LSTM where the results are assessed.

3.3. Transformer-based mode

Deep models based on transformers are currently commonly utilized in NLP. For user-based e-commerce product reviews classification in the proposed study, BERT is implemented. The encoder and decoder are the two major components of a transformer. The encoder takes words as input and generates embedding that encapsulates the meaning of the word, while the decoder uses the encoder's embedding to construct the next word until the sentence is completed. To effectively extract a contextual representation of provided phrases, we used BERT as a sentence encoder. BERT overcomes the unidirectional constraint by employing mask language modeling (MLM) [35]. It masks multiple tokens from the input at random and uses just the input to predict the original vocabulary id of the masked word. MLM has increased BERT's ability to outperform when compared to previous embedding methodologies. It is a deeply bidirectional system that can

analyze unlabeled text at all levels by conditioning on both left and right contexts using a transformer backend and the attention mechanism. When the attention mechanism receives the input data, it maps it to a multidimensional space and calculates the significance of each data point. The inputs are then contained in output transformations, and output solutions are generated by the layer stacks in both the encoder and decoder [36].

3.4. Classification

After all the above-mentioned steps, the feature sets from the N-gram methods are given as input to the four ML classifiers, and those from the word embedding methods to DL-based CNN and LSTM are classified in their respective classes. The proposed work uses four ML-classifiers random forest (RF), support vector machine (SVM), logistic regression (LR), and multinomial Naïve Bayes (MNB)as well as DL-based CNN comprising embedding, convolutional, max-pooling, and SoftMax layer and LSTM for the classification of textual and word embedding feature sets respectively. In parallel, BERT also performs classification by taking in the preprocessed dataset and providing their class prediction. All the results are discussed in detail in section 4.

4. EXPERIMENTATION AND RESULTS

The proposed framework takes raw input data containing Amazon product reviews and applies certain preprocessing and data balancing steps to it. N-gram methods TF-IDF [37], BoW, and word embedding methods GloVe and Word2vec are then implemented for feature extraction. The extricated future sets are finally classified using certain ML and DL algorithms. BERT is also implemented to derive transformer-based features from the preprocessed data and the predictions generated by it are compared with ML and DL models for a comparative study with the help of performance evaluation metrics. The experimental work is carried out on both datasets individually where all the steps are applied on D1 followed by the implementation of the same steps on D2. The sections below will cover the experiments conducted on D1 followed by D2 and then provide an elaborative comparison of both.

In the first experiment conducted on D1, textual features are given to four ML classifiers SVM, RF, LR, and MNB. The results evaluated by performance evaluation measures (PEM) such as accuracy, precision, recall, and f1-score, are shown in Table 2. The experiments are carried out in Python, and the package used to integrate the model into the experimental space is called the "sklearn" ensemble. All the models are trained and evaluated using 90% and 10% of the dataset respectively.

Table 2. Classification results of ML models with textual features of D1

PEM	SVM-TF-IDF (%)	SVM-BoW (%)	RF-TF-IDF (%)	RF-BoW (%)	LR-TFIDF (%)	LR-BoW (%)	MNB-TF-IDF (%)	MNB-BoW (%)
Accuracy	86.16	86.67	82.53	82.32	88.47	86.52	86.88	87.18
Precision	86	87	83	82	89	87	87	87
F1 score	86	87	83	82	89	87	87	87
Recall	86	87	83	82	89	87	87	87

Figure 2 shows a graphical performance comparison of all four ML models with textual features derived from D1. The LR and MNB classifiers provide the highest accuracies of 88.47% and 87.18% with TF-IDF and BoW features respectively. Figure 3 shows the performance comparison for both LR and MNB. The RF classifier yields the lowest results while only achieving accuracies of 82.53% and 82.32% on TF-IDF and BoW features respectively. RF also has the lowest precision, f1-score and recall as compared to the rest. Apart from RF, other classifiers perform almost alike with a small difference with not a huge accuracy gap between TF-IDF and BoW derived features.

In the second experiment conducted on D1, the two DL models including CNN and LSTM are provided with the word embedding features extracted by both GloVe and Word2vec models. CNN, which is selected for speed and accuracy is trained and tested on the same data split utilized in ML algorithms. The number of epochs varied from 5 to 100 and a batch size of 32 is maintained for CNN training. In the case of LSTM, the batch size is kept at 32, epochs are set to 5, while the main layers that constitute LSTM are embedding, dense and SoftMax layers. Apart from providing the word embeddings to these ML and DL models, the preprocessed dataset is given as an input to BERT, which derives its encodings from the preprocessed D1, takes as input the preprocessed D1, extracts embedding representations from it, and maps transformations on it. Finally, it decodes the representations back into vocabulary-based representations and uses its deep layers to perform classification. The same data split is maintained for BERT as well while the number of epochs is kept at 5 epochs and a batch size of 16 is maintained. Table 3 shows the results of CNN, LSTM, and BERT when word embeddings and prepared datasets are given to them respectively.

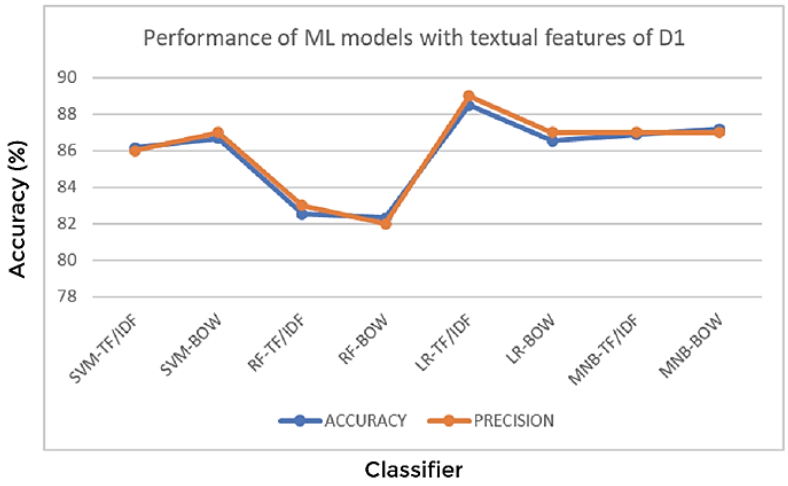


Figure 2. Results comparison of ML models with textual features of D1

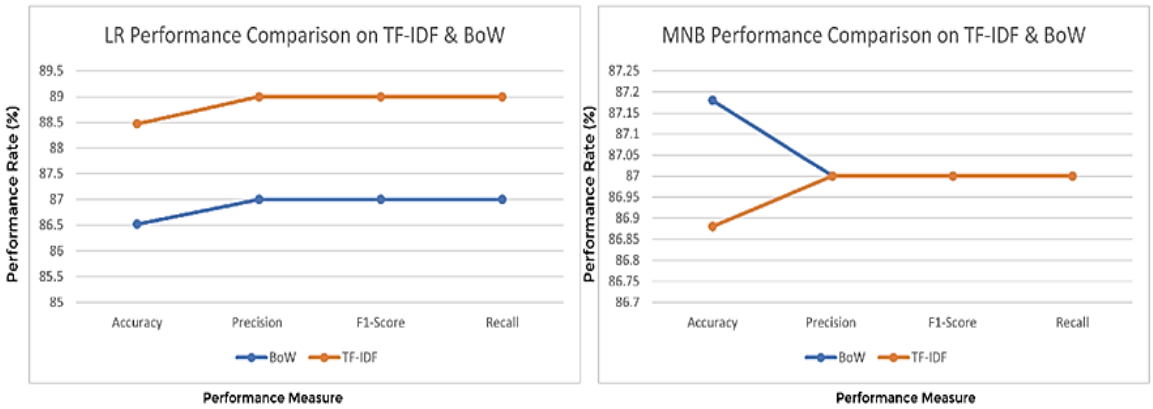


Figure 3. Performance comparison of best performing LR and MNB models on D1

As evident from Table 3 that CNN performs better as compared to LSTM in terms of all PEMs when applied to word embeddings derived from D1. CNN excels in terms of accuracy and other PEMs for both GloVe and Word2vec features. On the other hand, BERT outperforms both CNN and LSTM and achieves the highest performance rates with an accuracy of 90%. BERT is the best-performing model on D1 as its performance dominates that of ML and DL model's performance concerning accuracy and other PEMs. Figure 4 shows the graphical visualization of the accuracy and loss of CNN in Figure 4(a) and Figure 4(b) respectively.

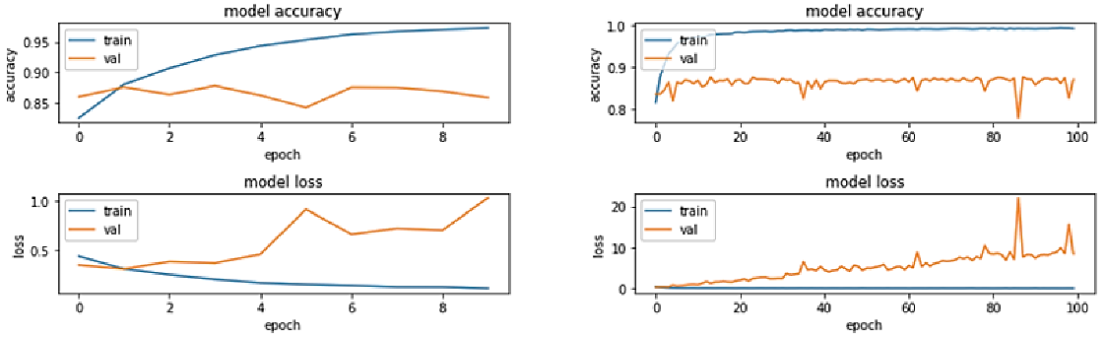
Table 3. Classification results of DL algorithms with word embedding features of D1

PEM	CNN-GloVe (%)	CNN-Word2vec (%)	LSTM-GloVe (%)	LSTM-Word2vec (%)	BERT
Accuracy	87.75	86.46	86.37	86.28	90
Precision	88	87	86	86	90
F1 score	88	87	86	86	89
Recall	88	86	86	86	89

As shown in Figure 5 CNN initiates with less accuracy and a higher loss rate while training on both GloVe and Word2vec features but then goes on to achieve a considerably higher accuracy rate. The reason for that is that CNN gradually trains on the input data, starts learning deep features from the data using its deep layers. As time progresses and layers get more and more trained, the predictions start becoming better and loss rate significantly falls. DL models perform better on larger datasets as they have much input to learn their features from so the more data is given to them, better prediction starts showing up. Figure 5 shows the accuracy and loss ratio for the LSTM model when trained and evaluated on word embeddings in Figure 5(a) and Figure 5(b) respectively.

LSTM also shows an uprising curve with the time when the accuracy rate increases, and the loss rate decreases. Figure 6 shows the accuracy graph for BERT for epochs when trained and tested on D1 in Figure 6(a)

and Figure 6(b) respectively. The reason for that is that LSTM gradually trains on the input data, starts learning deep features from the data using its deep layers. As time progresses and layers get more and more trained, the predictions start becoming better and loss rate significantly falls. DL models perform better on larger datasets as they have much input to learn their features from so the more data is given to them, better prediction starts showing up.



(a) (b)

Figure 4. Accuracy and loss ratio visualization of CNN with (a) GloVe features of D1 and (b) Word2vec embedding features of D1

After the implementation of all steps included in the proposed methodology on D1, the same steps are repeated for D2. In the first experiment conducted on D2, textual features are given to four ML classifiers SVM, RF, LR, and MNB. The results evaluated by performance evaluation metrics accuracy, precision, recall, and f1-score, are shown in Table 4. These experiments are also carried out in Python with the "sklearn" ensemble integration package. All the models are trained and evaluated using 90% and 10% of the dataset, respectively.

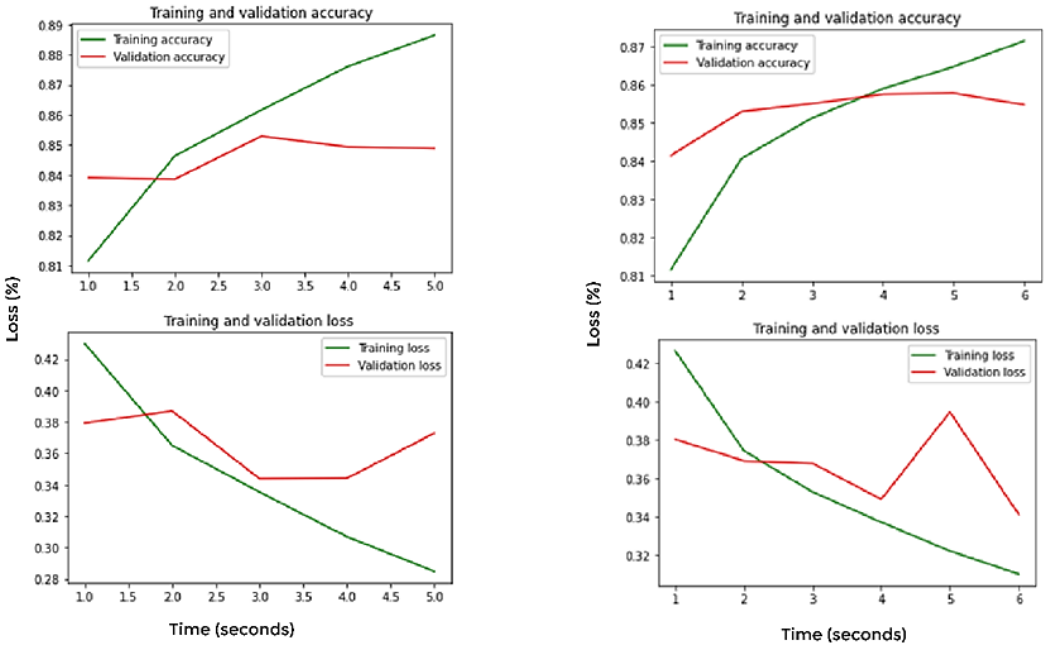
Figure 7 shows a graphical performance comparison of all four ML models with textual features derived from D2. The SVM and LR classifiers provide the highest accuracy rates of 94.02% and 93.99% with Figure 8 shows the performance comparison of both LR and SVM when applied to textual features of D2. TF-IDF and BoW features, respectively and hence are the best-performing models. All the ML models perform a lot better in general in the case of D2 as compared to D1 as can be observed by comparing Table 2 and Table 3. The reason could be that D2 is better prepared and engineered as compared to D1.

Same to the experiments conducted on D1, the two DL models including CNN and LSTM are provided with the word embedding features extracted by both GloVe and Word2vec models from D2. In the case of CNN, the number of epochs is increased from 5 to 10 while a batch size of 32 is maintained. In the case of LSTM, the batch size is kept at 32, epochs are set to 5, and the same deep layers are maintained. In parallel to that, the preprocessed D2 is fed into BERT for the derivation of transformer-based representations while the same number of epochs and batch size are maintained. Table 5 shows the results of CNN, LSTM, and BERT when word embedding features and processed D2 are given to them, respectively.

As evident from Table 5 CNN outperforms both LSTM and BERT for both GloVe and Word2vec features. It achieves an accuracy of 97.12% for GloVe and 96.66% for Word2vec features which is considerably superior to its counterparts. The reason for such an increment in performance over D2 could be the preparation and feature engineering of D2 as compared to D1. This better preparation and engineering led to an increment in the performance of all ML models, DL-based CNN, LSTM as well as BERT in the case of D2 whereas all the models were limited to a maximum of 90% accuracy in case of D1. Figure 9 shows the graphical visualization of the accuracy and loss of CNN in Figure 9(a) and Figure 9(b). CNN started with a high loss rate, eventually learns deep features and improves its performance over time and epochs.

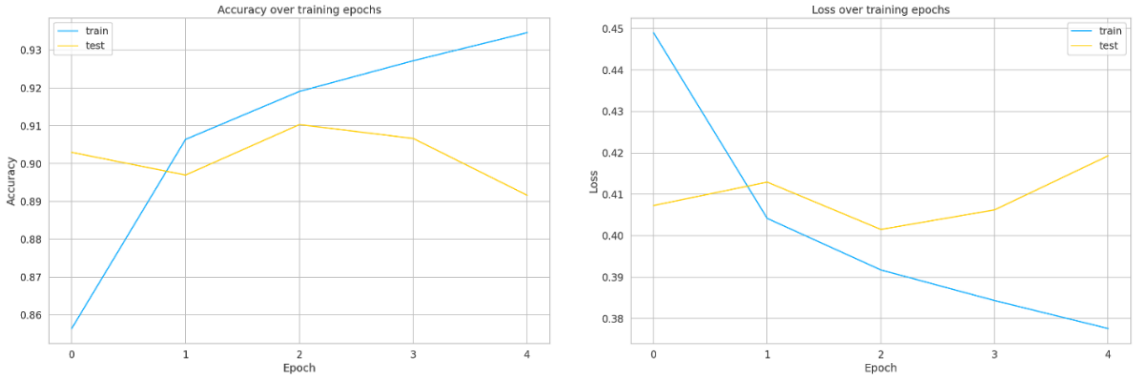
Table 4. Classification results of ML models with textual features of D2

PEM	SVM-TF-IDF (%)	SVM-BoW (%)	RF-TF-IDF (%)	RF-BoW (%)	LR-TFIDF (%)	LR-BoW (%)	MNB-TF-IDF (%)	MNB-BoW (%)
Accuracy	94.02	93.07	92.98	93.47	93.73	93.99	93.53	92.93
Precision	93	92	86	94	93	93	89	92
F1 score	94	93	93	93	94	94	94	93
Recall	93	93	90	90	91	93	91	92



(a) (b)

Figure 5. Accuracy and loss ratio visualization of LSTM with (a) GloVe features of D1 and (b) Word2vec embedding features of D1



(a) (b)

Figure 6. BERT model with (a) training accuracy on D1 and (b) loss graph on D1

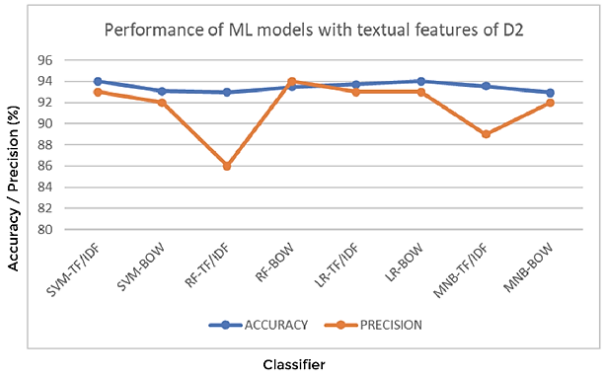


Figure 7. Results comparison of ML models with textual features of D2

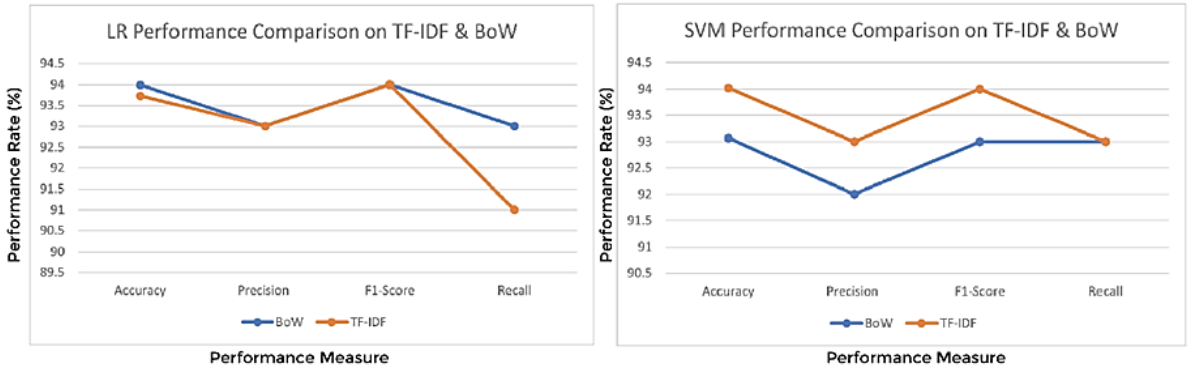
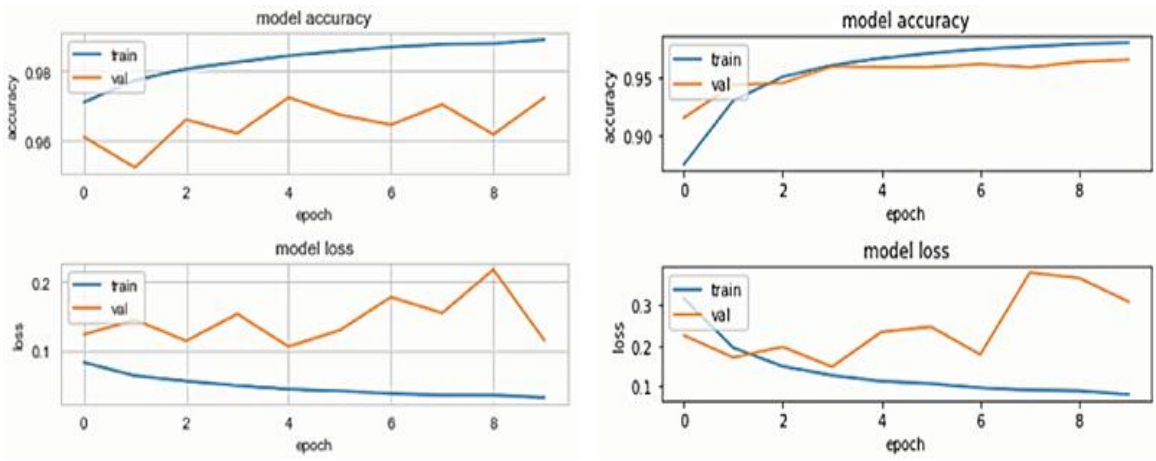


Figure 8. Performance comparison of best performing LR and SVM models on D2

Table 5. Classification results of DL models and BERT with word embedding features of D2

PEM	CNN-GloVe (%)	CNN-Word2vec (%)	LSTM-GloVe (%)	LSTM-Word2vec (%)	BERT
Accuracy	97.12	96.66	93.91	92.26	95.17
Precision	97	97	94	93	95
F1 score	97	97	94	92	85
Recall	97	97	94	92	85

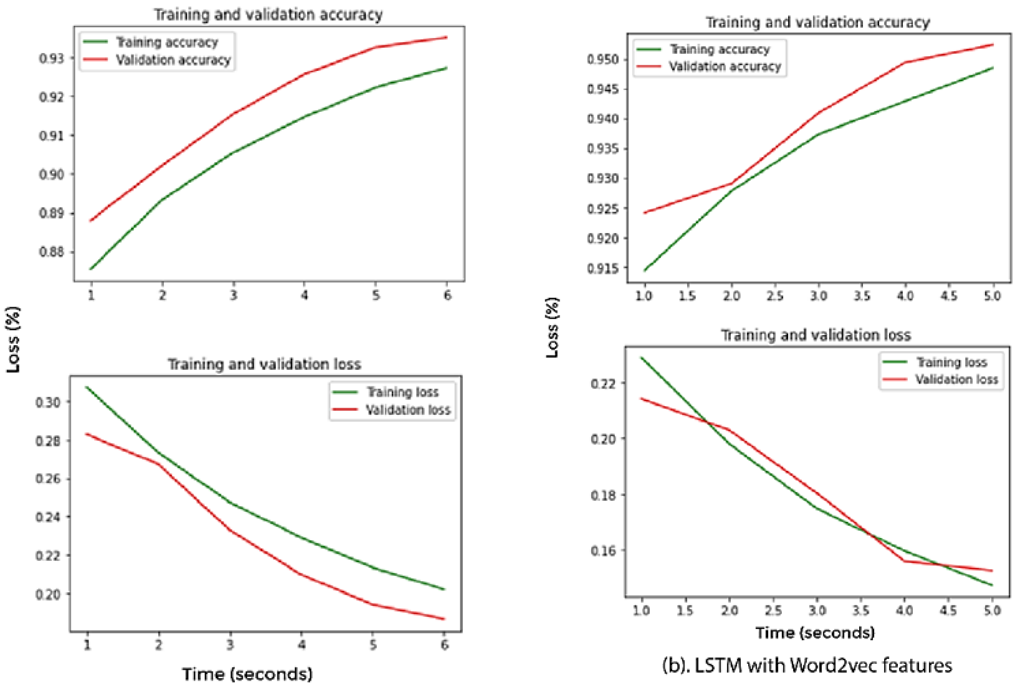
Figure 10 shows the accuracy and loss ratio for the LSTM model in Figure10(a) and Figure 10(b) respectively, when trained and evaluated on word embedding features derived from D2 while maintaining the same settings as in the case of D1. Here again LSTM starts with a high loss rate and minimum accuracy and eventually excels in terms of performance. The reason for that is that LSTM gradually trains on the input data, starts learning deep features from the data using its deep layers. As time progresses and layers get more and more trained, the predictions start becoming better and loss rate significantly falls. DL models perform better on larger datasets as they have much input to learn their features from so the more data is given to them, better prediction starts showing up.



(a) (b)

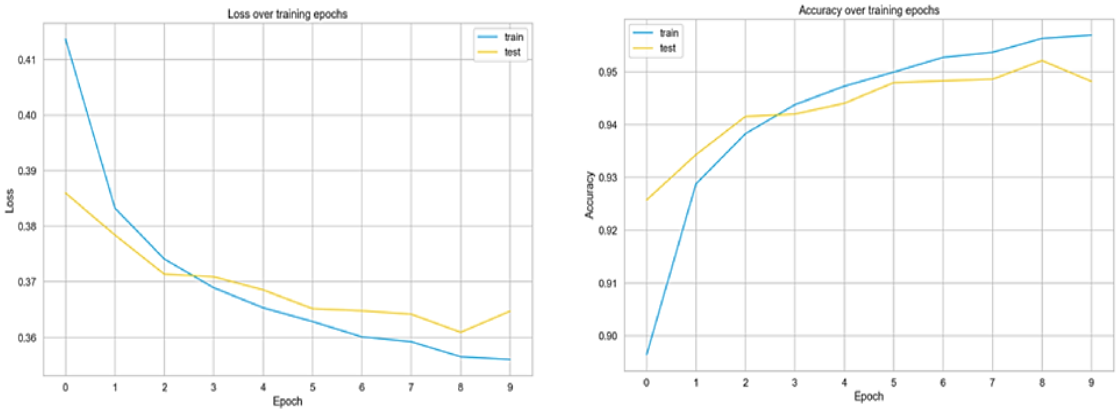
Figure 9. Accuracy and loss ratio visualization of CNN with (a) GloVe features of D2 and (b) Word2vec embedding features of D2

After the implementation of CNN and LSTM upon D2's word embedding features. The preprocessed D2 is given to BERT to compare the performance of the ML and DL models performance with it. BERT takes in the data and performs self-driven classification results which are discussed in Table 5. Figure 11 shows the accuracy to lose ratio graph for BERT for epochs when trained and tested on D2 in Figure 11(a) and Figure 11(b) respectively.



(a) (b)

Figure 10. Accuracy and loss ratio visualization of LSTM model with (a) GloVe features of D2 and (b) Word2vec embedding features of D2



(a) (b)

Figure 11. BERT model for (a) training accuracy on D2 and (b) loss graph on D2

5. DISCUSSION

All the experiments performed for the proposed work are discussed in detail in the preceding section along with the results. It is quite evident from experiments conducted on D1 that ML models perform better in general as compared to DL models in terms of accuracy and other PEMs. Although BERT outperforms both ML and DL models regarding the accuracy of 90% as well as all other PEMs. The accuracy comparison of ML, DL models, and BERT when applied to D1, is visualized in Figure 12.

In the case of D2, DL-based CNN outperforms all the ML models along with BERT. It achieves the highest accuracy rates of 97.12% for GloVe and 96.66% for Word2vec features which is considerably superior to its counterparts. The accuracy comparison of ML, DL models, and BERT when applied on D2, is visualized in Figure 13.

The fact to be noticed here is that experiments conducted on D1 show considerably fewer performance rates in general as the highest accuracy, in this case, turn out to be 90% by the BERT. After merging D1 with Amazon's general product reviews dataset to form D2 and performing over-sampling on it, the performance of the model is increased to a huge extent. This phenomenon is demonstrated in Figure 14 where ML, DL, and BERT models are compared based on accuracy upon textual features, word embedding features, and datasets themselves, respectively, in the case of both D1 and D2. The merged dataset D2 produces better results in general, as compared to D1. The proposed model excels in terms of accuracy and other PEMs as compared to D1. It is evident from Figure 14, Figure 15, and Figure 16 that the performance of ML models, DL models and BERT is better on D2 as compared to D1.

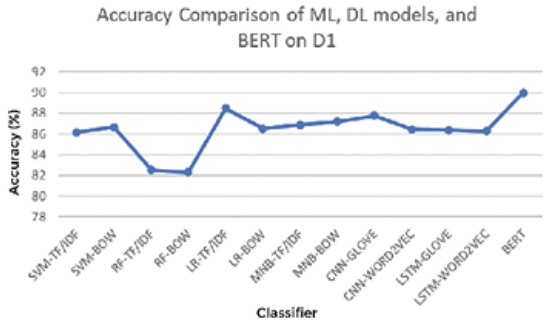


Figure 12. Accuracy comparison of ML, DL models, and BERT on D1

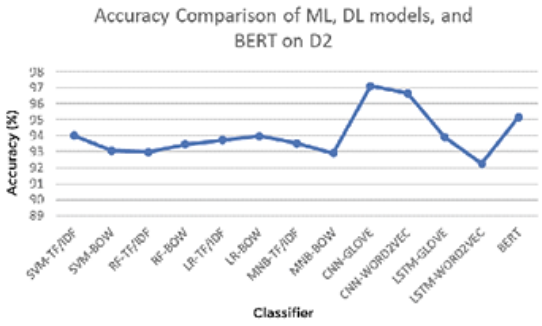


Figure 13. Accuracy Comparison of ML, DL models, and BERT on D2

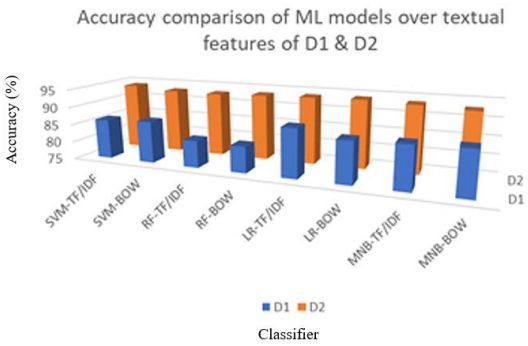


Figure 14. Accuracy comparison of ML models over D1 and D2

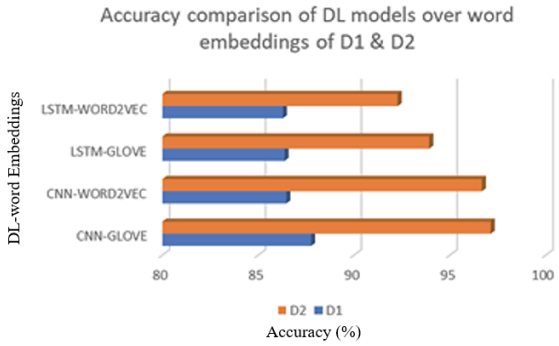


Figure 15. Accuracy comparison of DL models over D1 and D2

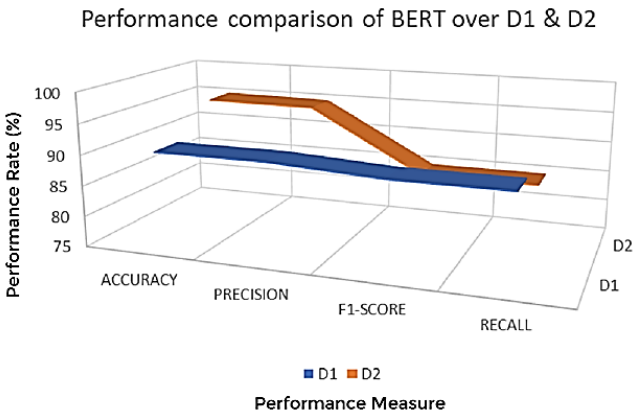


Figure 16. Accuracy comparison of BERT over D1 and D2

6. CONCLUSION

The ever-growing trend of online shopping through e-commerce platforms has resulted in a massive surge of reviews posted regarding various product categories on them. Analyzing these reviews helps product and platform owners to improve their services and maintain higher standards. Since the reviews are present in bulk, contain opinions of people around the globe, and are based on positive and negative sentiments, manually sorting, and analyzing them is not possible. Therefore, an automated system is presented in this work that takes input from the two product review datasets gathered from Amazon. The implication of several preprocessing steps ensures that the data is well-balanced and ready for the feature extraction phase. The feature extraction is carried out with textual feature derivation techniques, such as TF-IDF, BoW, and word embedding feature extractors GloVe, and Word2vec. Multiple ML models including SVM, RF, LR, MNB, and a couple of DL models CNN, and LSTM are applied to textual and word embedding features, respectively. BERT is also applied to both datasets to compare its performance with ML and DL models. BERT turns out to be the best-performing model in the case of D1 with an accuracy of 90% on features derived by word embedding models while RF provides the minimum accuracy rate of 86% on textual features. In the case of D2, CNN provides the best accuracy of 97% upon word embedding features while RF and MNB show the minimum accuracy rates of 92%. The proposed model shows better overall performance on D2 as compared to D1. The model shows better overall results on D2 with the increase in data when compared with several performance metrics. The proposed model uses two datasets to perform preprocessing, feature engineering, textual feature extraction, deep feature extraction, classification based on ML as well as DL models and BERT-based classification. When textual features are classified using ML models, The LR and MNB classifiers provide the highest accuracies of 88.47% and 87.18% with TF-IDF and BoW features, respectively whereas in case of deep features, BERT has the highest accuracy of 90% as compared to both ML and DL models. This causes the limitation in case of D1 as the overall accuracy is not exceeding 90%. In future we might have to apply much better methodologies for data cleaning, pruning and feature engineering and also optimize ML, DL models and BERT according to them to increase our overall accuracy.

In case of D2, the DL-based CNN achieves an accuracy of 97.12% on deep features extracted by GloVE and Word2vec, outperforming ML models by a large margin and BERT by a significant margin. This proves that D1 and D2 have provided different results indicating that their preparation or feature extraction has a lot of difference. We need to study the difference closely and apply the best result deriving methodology in future works. To derive much better results, we can further enhance our CNN model, BERT models and look to apply generative pre-trained transformer (GPT) for much better results.

ACKNOWLEDGEMENTS

The authors would like to thank Arab Open University research group number: "AOURG-2023-020", Saudi Arabia for supporting this study.

REFERENCES

[1] R. Liang and J. -Q. Wang, "A linguistic intuitionistic cloud decision support model with sentiment analysis for product selection in E-commerce," *International Journal of Fuzzy Systems*, vol. 21, pp. 963–977, 2019, doi: 10.1007/s40815-019-00606-0.

[2] Y. Basani, H. V. Sibuea, S. I. P. Sianipar, and J. P. Samosir, "Application of sentiment analysis on product review e-commerce," *Journal of Physics: Conference Series*, 2019, doi: 10.1088/1742-6596/1175/1/012103.

[3] D. Coppola, *Amazon - Statistics & Facts*, Statista, Jun. 2023. [Online]. Available: https://www.statista.com/topics/846/amazon/#topicOverview

[4] Y. Ma, *Alibaba Group - Statistics & Facts*, Statista Key Figures of E-Commerce, Nov. 2022. [Online]. Available: https://www.statista.com/topics/2187/alibaba-group/

[5] *Revenue of Flipkart Private Limited between financial year 2014 and 2021*, Statista Key Figures of E-Commerce, 2022. [Online]. Available: statista.com/statistics/1053314/india-flipkart-revenue/

[6] R. Ireland and A. Liu, "Application of data analytics for product design: Sentiment analysis of online product reviews," *CIRP Journal of Manufacturing Science and Technology*, vol. 23, pp. 128–144, 2018, doi: 10.1016/j.cirpj.2018.06.003.

[7] B. S. Rintyarna, R. Sarno, and C. Fatichah, "Evaluating the performance of sentence level features and domain sensitive features of product reviews on supervised sentiment analysis tasks," *Journal of Big Data*, vol. 6, no. 84, 2019, doi: 10.1186/s40537-019-0246-8.

[8] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, pp. 685–695, 2021, doi: 10.1007/s12525-021-00475-2.

[9] A. M. Hoyle, P. Goel, and P. Resnik, "Improving neural topic models using knowledge distillation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1752–1771, doi: 10.18653/v1/2020.emnlp-main.137.

[10] D. A. J. Daniel and M. J. Meena, "A Novel Sentiment Analysis for Amazon Data with TSA based Feature Selection," *Scalable Computing: Practice and Experience*, vol. 22, no. 1, 2021, doi: 10.12694/scpe.v22i1.1839.

[11] X. Li, X. Sun, Z. Xu and Y. Zhou, "Explainable Sentence-Level Sentiment Analysis for Amazon Product Reviews," 2021 *5th International Conference on Imaging, Signal Processing and Communications (ICISPC)*, 2021, pp. 88-94, doi: 10.1109/ICISPC53419.2021.00024.

[12] N. Shrestha and F. Nasoz, "Deep learning sentiment analysis of amazon. com reviews and ratings," *International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI)*, vol. 8, no. 1, 2019, doi: 10.48550/arXiv.1904.04096.

[13] E. I. Elmurngi and A. Gherbi, "Unfair reviews detection on amazon reviews using sentiment analysis with supervised learning techniques," *Journal of Computer Science*, vol. 14, no. 5, pp. 714–726, 2018, doi: 10.3844/jcssp.2018.714.726.

[14] M. V. Rao and Sindhu C., "Detection of Sarcasm on Amazon Product Reviews using Machine Learning Algorithms under Sentiment Analysis," *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2021, pp. 196-199, doi: 10.1109/WiSPNET51692.2021.9419432.
[15] S. Wassan, X. Chen, T. Shen, M. Waqar, and N. Jhanjhi, "Amazon product sentiment analysis using machine learning techniques," *Revista Argentina de Clínica Psicológica*, vol. 30, no. 1, pp. 695-703, 2021. [Online]. Available: https://www.researchgate.net/publication/349772322_Amazon_Product_Sentiment_Analysis_using_Machine_Learning_Techniques
[16] M. Hawlader, A. Ghosh, Z. K. Raad, W. A. Chowdhury, M. S. H. Shehan, and F. B. Ashraf, "Amazon Product Reviews: Sentiment Analysis Using Supervised Learning Algorithms," *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, 2021, pp. 1-6, doi: 10.1109/ICECIT54077.2021.9641243.
[17] N. M. Alharbi, N. S. Alghamdi, E. H. Alkhammash, and J. F. Al Amri, "Evaluation of sentiment analysis via word embedding and RNN variants for Amazon online reviews," *Mathematical Problems in Engineering*, vol. 2021, 2021, doi: 10.1155/2021/5536560.
[18] A. Dadhich and B. Thankachan, "Sentiment Analysis of Amazon Product Reviews Using Hybrid Rule-based Approach," *International Journal of Engineering and Manufacturing (IJEM)*, vol. 11, no. 2, 2021, doi: 10.5815/ijem.2021.02.04.
[19] U. Norinder and P. Norinder, "Predicting Amazon customer reviews with deep confidence using deep learning and conformal prediction," *Journal of Management Analytics*, vol. 9, no. 1, pp. 1–16, 2022, doi: 10.1080/23270012.2022.2031324.
[20] P. Bhuvaneshwari, A. N. Rao, Y. H. Robinson, and M. Thippeswamy, "Sentiment analysis for user reviews using Bi-LSTM self-attention based CNN model," *Multimedia Tools and Applications*, vol. 81, pp. 12405–12419, 2022, doi: 10.1007/s11042-022-12410-4.
[21] N. Nandal, R. Tanwar, and J. Pruthi, "Machine learning based aspect level sentiment analysis for Amazon products," *Spatial Information Research*, vol. 28, pp. 601–607, 2020, doi: 10.1007/s41324-020-00320-2.
[22] S. Dey, S. Wasif, D. S. Tonmoy, S. Sultana, J. Sarkar, and M. Dey, "A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews," *2020 International Conference on Contemporary Computing and Applications (IC3A)*, 2020, pp. 217-220, doi: 10.1109/IC3A48958.2020.233300.
[23] H. Zhao, Z. Liu, X. Yao, and Q. Yang, "A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach," *Information Processing & Management*, vol. 58, no. 5, 2021, doi: 10.1016/j.ipm.2021.102656.
[24] P. Mukherjee, Y. Badr, S. Doppalapudi, S. M. Srinivasan, R. S. Sangwan, and R. Sharma, "Effect of negation in sentences on sentiment analysis and polarity detection," *Procedia Computer Science*, vol. 185, pp. 370–379, 2021, doi: 10.1016/j.procs.2021.05.038.
[25] M. Sivakumar and S. R. Uyyala, "Aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic," *International Journal of Data Science and Analytics*, vol. 12, pp. 355–367, 2021, doi: 10.1007/s41060-021-00277-x.
[26] A. Huang, "A risk detection system of e-commerce: researches based on soft information extracted by affective computing web texts," *Electronic Commerce Research*, vol. 18, pp.143-157, 2018, doi: 10.1007/s10660-017-9262-y.
[27] S. Vanaja and M. Belwal, "Aspect-Level Sentiment Analysis on E-Commerce Data," *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2018, pp. 1275-1279, doi: 10.1109/ICIRCA.2018.8597286.
[28] M. E. Alzahrani, T. H. H. Aldhyani, S. N. Alsubari, M. M. Althobaiti, and A. Fahad, "Developing an Intelligent System with Deep Learning Algorithms for Sentiment Analysis of E-Commerce Product Reviews," *Computational Intelligence and Neuroscience*, vol. 2022, 2022, doi: 10.1155/2022/3840071.
[29] K. L. Tan, C. P. Lee, K. M. Lim, and K. S. M. Anbananthen, "Sentiment Analysis With Ensemble Hybrid Deep Learning Model," in *IEEE Access*, vol. 10, pp. 103694-103704, 2022, doi: 10.1109/ACCESS.2022.3210182.
[30] M. A. Qureshi *et al.*, "Sentiment Analysis of Reviews in Natural Language: Roman Urdu as a Case Study," in *IEEE Access*, vol. 10, pp. 24945-24954, 2022, doi: 10.1109/ACCESS.2022.3150172.
[31] E. Cambria, Q. Liu, S. Decherchi, F. Xing, and K. Kwok, "SenticNet 7: A Commonsense-based NeurosymbolicAI Framework for Explainable Sentiment Analysis," *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, 2022, pp. 3829–3839. [Online]. Available: https://sentic.net/senticnet-7.pdf
[32] A. Zhao and Y. Yu, "Knowledge-enabled BERT for aspect-based sentiment analysis," *Knowledge-Based Systems*, vol. 227, 2021, doi: 10.1016/j.knosys.2021.107220.
[33] *Google news vector*, kaggle, Apr. 2022. [Online]. Available: https://www.kaggle.com/datasets/adarshsng/googlenewsvectors
[34] L. Khreisat, "Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study," *International Conference on Data Mining*, 2006, pp. 78–82. [Online]. Available: https://dblp.uni-trier.de/rec/conf/dmin/Khreisat06.html
[35] P. Wang, J. Hu, H. -J. Zeng, and Z. Chen, "Using Wikipedia knowledge to improve text classification," *Knowledge and Information Systems*, vol. 19, pp. 265–281, 2009, doi: 10.1007/s10115-008-0152-4.
[36] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF* IDF, LSI and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, 2011, doi: 10.1016/j.eswa.2010.08.066.
[37] Z. Y. -Tao, G. Ling, and W. Y. -Cheng, "An improved TF-IDF approach for text classification," *Journal of Zhejiang University-Science A*, vol. 6, pp. 49–55, 2005, doi: 10.1007/bf02842477.

BIOGRAPHIES OF AUTHORS

Saman Iftikhar received her M.S and Ph.D. degrees in Information Technology in 2008 and 2014, respectively, from National University of Sciences and Technology (NUST), Islamabad, Pakistan. Currently she is serving Arab Open University, Saudi Arabia as an Assistant Professor. Her research interests include information security, cyber security, machine learning, data mining, distributed computing, and semantic web. On her credit, several research papers have been published in various reputed journals and in prestigious conferences. She can be contacted at email: s.iftikhar@arabou.edu.sa.

Bandar Alluhaybi received the B.S. degree in computer science from King Abdulaziz University, Saudi Arabia, in 2009, the M.S. degree in engineering system management from St. Mary's University, San Antonio, TX, USA, in 2012, and the Ph.D. degree in computer science from King Abdulaziz University. He is a Lecturer at FCS, Arab Open University, KSA. His current research interests include information security, computer networks, networks security, big data, and high-performance computing. He can be contacted at email: b.alluhaybi@arabou.edu.sa.

Mohammed Suliman received the BCA and MCA degree in computer applications from Bangalore University, India, in 2004, 2007, respectively. He is a Lecturer at FCS, Arab Open University, KSA. His current research interests include information security, network security, cloud computing, big data, and software engineering. He can be contacted at email: msuliman@arabou.edu.sa.

Ammar Saeed did his Bachelor in Computer science from COMSATS University Islamabad, Pakistan in 2019. Currently, he is pursuing Masters in Computer Science from COMSATS University Islamabad, Pakistan. His major areas of research interest are Machine Learning, Natural language processing and Data Analytics. He can be contacted at email: ammarsaeed1997@gmail.com.

Kiran Fatima Completed PHD (CS) in 2018 from National University of Computer and Emerging Sciences Islamabad Pakistan. Majors are Artificial intelligence, machine learning and image processing. Currently working in TAFE NSW Australia as Network and Web Programming trainer. She can be contacted at email: kiran.fatima4@tafensw.edu.au.