Journal of Information Systems Engineering and Management
2025, 10(10s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Sentiments Analysis Prediction of The Arabic Stock-Market News Based on Machine- and Deep-Learning Approaches

Eman Alasmari[1], Fahd Saleh Alotaibi[1]
[1]*The Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia.*
Corresponding Author: Eman Alasmari: ealasmari0010@stu.kau.edu.sa
Fahd Saleh Alotaibi: fsalotaibi@kau.edu.sa

ARTICLE INFO

Received: 16 Nov 2024

Revised: 22 Dec 2024

Accepted: 10 Jan 2025

ABSTRACT

Stock market prediction of companies is a vital interest for financial analysts, investors, and other competitors. There is difficulty in predicting the future status of the companies' stocks. However, stock market behavior depends on the polarity prediction classification. Therefore, it is essential to use sentiment analysis to study attention indicators for stock market behavior in the news. Sentiment analysis (SA) can be used to extract public sentiments from stock news microblog platforms. Previous studies used machine learning (ML) algorithms to classify Arabic stock news into positive, negative, or neutral types. Recently, deep learning (DL) algorithms have provided good accuracy for Arabic SA. Motivated by such results; this study applies ML and DL techniques to classify sentiments of Arabic stock news. 30,098 articles were collected and preprocessed from the Saudi stock-market platform, Tadawul. For the sake of comparison, two ML and two DL techniques were performed for SA: Naive Bayes (NB), logistic regression, fast-text, and long short-term memory (LSTM). These algorithms were used to classify the sentiments of the collected data and help investors and stock analysts with decision-making. The results show that the DL techniques outperformed the ML algorithm. The experimental result of the LSTM model was 84%, which is the same as the reduced-features logistic regression model, but it has the lowest features over the same timeframe. Therefore, the LSTM model simultaneously has the best accuracy and the fewest features. On the other hand, the NB models achieve the worst performance. The Arabic SA models assist in decision-making based on the stocks news sentiments predicting the upcoming stock trends for the investors or analysts. These sentiment's models would limit the risks by supporting the decision-making analysts. Thus, this study would be a valuable resource to the stock market sector based on Arabic linguistic features.

Keywords: Sentiments Analysis Prediction, Deep-Learning

1. Introduction

There is no doubt that the association of stock-market news and stock price implies particular importance to investment value [1]. Stock articles possess sensitive variables based on their dynamic behaviors that support the investment sector [2–4]. The Arabic stock market represents a prominent advanced capital market for financial planning [1,3]. However, stock prices are dynamic, owing to the quick fluctuations in the domain. Sudden stock-market volatility creates a risk to investments and investors' credibility [1,2]. Foremost, analysts and investors face difficulty predicting company stocks' future statuses [3,5]. Therefore, the prediction problems of the stock market affect the decision-making of the investors. Whereas knowing the best time to buy or sell stocks is the main goal of prediction [5,6]. The stock news indicators provide an opportunity of analyzing the market through news sentiment, expectations, or behaviors. This fact produces the need for an accurate prediction model of stock behavior to reduce the trading risk [5]. The sentiment of stock news regarding companies provides primary variables that impact stock prices [6]. Thus, it is essential to use sentiment analysis (SA) to study attention indicators for stock-market behaviors in the stock news [3,5,6].

Language obstacles present challenges to the SA of textual news in the research community [7]. The Arabic language adds another layer of difficulty, owing to its complex language structure. Arabic is a Semitic language widely used in North Africa and Southwest Asia [7, 8]. Therefore, the Arabic stock news SA requires a developed model to increase

classification accuracy. This improved model requires linguistic preprocessing and labeling phases alongside classification techniques based on Arabic language restrictions [9,10].

The Arabic language further complicates the required natural language processing (NLP) [9]. Recent studies have used machine learning (ML) and deep learning (DL) for Arabic NLP [10], but none have used both for Arabic SA, per the literature review. There is a driving need to apply this type of NLP to the SA classification of Arabic textual stock news for prediction purposes [9,10]. Accordingly, this research seeks to develop ML and DL classifiers of Arabic stock sentiments based on supervised machine-learning techniques. To the extent of the authors' knowledge, this is the only study of its kind—showing the ML and DL classifiers on Arabic stock news. So, the developed models are valuable resources that classify the Arabic and Saudi Stock Market news based on their polarity. The sentiment models help to perceive the risks based on the Stock Market articles by the decision-making supporting analysts' evaluation and investing. Hence, it is a motivation to cover the lack of studying DL and ML in SA classification of Arabic stock news for prediction purposes. Thus, implementing this study would be an essential contribution to the stock market sector based on Arabic linguistic features.

2. Related Work

2.1 Arabic SA in Microblogs

In 2016, Arabic SA research articles spiked, owing to the increased demand and value to the Arab world. Globally, the SA of microblogs tends to focus on the English language. However, Arabic SA techniques are deficient, and English SA does not work well with translated Arabic. However, it is vital to use semantic SA to analyze Arabic news to understand the financial implications of microblogs [11].

The Arabic language exhibits a diglossic nature, with more than one version (i.e., dialectal), which makes research in this field quite complicated. The dialects include classical Arabic (CA), modern standard Arabic (MSA), and colloquial Arabic (CA) [12, 13]. CA is used in the Quran (i.e., Islam's Holy Book), classical texts, and other Islamic writings. Arabic speakers use the CA rarely. Therefore, texting and blogging do not usually use CA. MSA is used in media, educational materials, micro-blogs, and other formal communications. MSA is well understood, and most Arabic speakers understand it, even those with mediocre levels of education. Finally, colloquial Arabic includes several dialects used in various Arabic regions. Such dialects are often only understood by their native speakers in that region. Nevertheless, few dialects are understood by many Arabic speakers apart from MSA [13].

2.2 Sentiment Representation in Microblogs

Stock-market prediction relies on sentiment indices. Index vendors provide stock-market news as a service to analyze the importance of stock-price prediction [14]. The sentiment of a microblog is impacted by sequence, feeling, and other features [15]. These sentiments carry emotional color differences, such as negative or positive connotations [15–17]. Hence, microblogs' stock data are valuable for SA to determine negativity and positivity [15, 16]. From this perspective, SA is essential to financial fields, particularly for classifying stock news [15]. This sentiment classification is commonly used to interpret customer satisfaction trends, stock price prediction, and other variables [18].

2.3 Different Techniques of News Classification and Prediction

SA for domain-specific languages is difficult to manage because of the scarcity of labeled datasets. Thus, researchers need a combination of text representation approaches and ML classification to overcome these challenges. Researchers use SA lexicons with NLP transformers to improve SA contextual embeddings, which overcome standard lexicon-based methods [19]. The multiple layer stacked ML classification system with NLP approaches extracts valuable linguistic features for decision-making models [20]. These features are chosen by select algorithms, such as the DL-driven forest algorithm. These algorithms have an excellent performance when enhancing dynamic data (e.g., stock-market prediction) [21–23]. Furthermore, using finer-grained text representations and SA improves the prediction tasks [21]. On the other hand, ML algorithms can handle the complexity of the trading market by establishing a stock prediction model [9,24]. Results show that such strategies are superior to those based on other models when applying assessment indicators of yield performance, risk performance, and feasibility. Hence, an ML strategy is needed [24].

SA and NLP tasks have leveraged ML techniques in the past. However, DL approaches are currently more pervasive [10,25]. Strategic trading decision-making based on news analyses is commonplace. However, sentiment algorithms

are better for fast-paced decision-making. Therefore, DL models replace human analysis. DL can train complex nonlinear models in large parameter spaces over big datasets [26]. Most studies have focused on improving contemporary models, including SA, for accuracy enhancement goals [27]. However, some of these enhanced strategies overlook the meanings and order of words [27, 28]. Hence, recurrent neural networks (RNNs) have gained popularity in creating architectural challenges related to model training. Hence, researchers have developed long short-term memory (LSTM), gated recurrent units (GRUs), and their bidirectional methods [27–28]. These architectures can extract features from temporal and sequential data [27, 28]. For continuous data, such as stock-market reporting, RNNs and LSTMs are better than other prediction models. Additionally, DL approaches are the best for binary data [23]. Notably, bidirectional encoder representations from the transformer (BERT)-bidirectional LSTM models are superior to standard BERT and LSTM for sentiment prediction [29].

2.4 Sentiment Classification Through Machine - and Deep-learning Approaches

Recently, some English studies presented ML and DL approaches for sentiment classification purposes [3,30,31]. In a study of emotions classification into five classes: Joy, Sadness, Fear, Shame, and Guilt, baseline model and BiLSMT are compared. Neural Network Technique Backpropagation Neural Classifier (BPN) achieved 71.27; conversely, BiLSMT is obtained 87.66 in the test accuracy [30]. Moreover, Senti-eSystem predicts the sentiment using hybridized Fuzzy and BiLSMT to assess customers' satisfaction levels. BiLSMT is applied with an attention mechanism for the SA polarity: positive and negative. The DL system obtained an accuracy of 92.86%, outperforming the state-of-art lexicon-based techniques [31].

ML and DL models have been examined for Arabic SA prediction compared to English SA studies. Most of these studies show that DL approaches overcome ML approaches in the performance accuracy [32–35]. In Saudi dialects' study, the SA is predicted for sentiments' classes. In the ML accuracy performance, Logistic Regression model achieved 74.95%, and SVM gained 98.98% [32], which is a proper prediction performance. In the DL applications, Bi-LSTM model obtained 94%, which is higher than ML performance [33]. In the Arabic studies, NB, LSTM, BERT models are applied for solving several SA classification problems [34–36] A Supervised ML approach is used NB, which obtained 54.43% in the accuracy result for emotion icons of SA [34]. The LSTM approach got 90% in the negative class accuracy and 93% in the positive class accuracy in the hotel reviewing SA [35]. Moreover, the BERT model reached 67% and 77.1% in predicting positive and negative sentiments' categories. However, the Arab-BERT model is used as a transformer-based model for the Arabic language is reached 96.1 and 96.2 for positive and negative categories' prediction [36].

3. Materials and Methods

Researchers use ML classification with DL models to measure stock news sentiment based on feature extraction techniques. This study compares several models' accuracy levels according to feature, parameter, and hyperparameter tuning. Multinomial naive Bayes (MNB) is the first compared model because it is one of the essential ML baseline approaches. Then, supervised ML and DL models are improved via hyperparameter tuning and improved processes to increase accuracy.

3.1 Data Collection

Data for this study were collected from the Tadawul website as a comma-separated-value file [37]. The dataset included corporate news and historical stock data from 2011 through 2019. There were 34,386 rows for articles and 16 columns for variables. The dataset attributes included *Field*, *Company Name*, *Company Number*, *Article Title*, *Article Details*, *Opening Price*, *Highest Price*, *Lowest Price*, *Closing Price*, *Change Percentage*, *Change Price*, *Quantity Handled*, *Total Price*, *Total of Deals*, *Date*, and *Time*.

3.2 Preprocessing Data and Annotation

The dataset was pre-processed by removing noise using the pandas.DataFrame tool and unifying the data formats with Arabic tokenization [38]. Manual validation tests were performed to assess the correctness of the final cleansed 30,089 observations. The SA method adopts supervised ML. Therefore, the dataset was annotated for polarity training and testing goals. Native Arabic speakers who worked in the domain manually labelled article polarities. This research adopts multiclass classification with a neutral category to increase model performance [39]. Hence, article details (attributes) are categorized according to three polarities: positive, negative, and neutral.

Inter-annotator agreement calculates the reliability of the annotators by comparing the same annotation agreements for a specific class. Inter-annotator agreement is a required step of the classification validation outcomes to determine the annotators' understanding [40]. Based on Kappa statistics, Inter-annotator agreement measures some annotators' agreement by the Fleiss' kappa over multiple annotators. The annotators are three Arabic native speakers who were split into three categories of various data from all the classes. Each one of them annotated 10,032 unique texts and 1,500 overlapping texts to be annotated by all of them. This produced 1500 annotated texts by three separate annotators, and a single annotator annotated the other texts. Fleiss' kappa is applied to evaluate the agreement reliability between the three annotators [41]. Therefore, according to the Kappa statistic interpretation, the agreement outcome of (K) equals 96%, which is almost complete agreement.

3.3 Data Splits

Prior to DL, several techniques existed for handling data splits, such as cross-validation, k-folds, and grid search. These methods are beneficial, but they do not provide the best way to train, test, and validate. K-fold cross-validation involves training the model numerous times to ensure its generalizability [42], leading to slower training speeds and expensive evaluation because the process is repeated k times for each hyperparameter [43]. For instance, three different values and 10-fold cross-validation lead to a 30-time repeated training and validation cycle. Hence, k-fold cross validation takes too long to train the DL model. Also, it tweaks models such as the learning rate, changing features, trying a new model from scratch to pick the best model on the test set, but in the end. However, in this study, the train_validation_test function is used to split the data into training, validation, and testing sets [43] to reduce overfitting. It also optimizes the hyperparameters used to evaluate results from the training set. Hence, after validation, the testing re-evaluates the model to increase training and evaluation speeds [43]. For instance, three different values and a three-way split strategy repeat the training and validation cycle only three times. Thus, splitting the data set into the three subsets allows choosing the best model and features on the validation set and double-checking that model in the test set. Also, it helps decrease the mistake cost and set the best hyperparameter values `.

3.4 Proposed ML Approaches

This study applies a baseline model (BL) to evaluate the proposed model's accuracy of multi-class classification. Researchers train MNB and logistic regression models to compare accuracy levels and to improve them. Feature extraction algorithms transform text articles into numerical data by following these steps:

- Tokenization:
  - The bag-of-words (BoW) method transfers row text to numbers using several parameters [44, 45], creating a column for one word as a one-gram (unigram) token.
  - N-gram then makes a column for one word, every ordered-pair, or more ordered words as unigram, bi-gram, trigram, and n-gram tokens. The n-gram model reveals word dependencies via strings and individual sentences for SA [44, 45].
- Lemmatization finds the normalized forms of words [46]. The ARLSTem algorithm is used for Arabic lemmatization.
- The term-frequency–inverse document frequency (TFITF) method calculates each word's weight based on the number of times it appears in a document. Hence, less common words have the highest weight [47, 48].

In this study, 80,284 tokens accounted for the features extracted from the vocabulary: 72,517 tokens were used for training, 27,619 were used for validation, and 27,999 were used for testing. Note that tokenization sometimes removes necessary characters, such as punctuation. This is avoided with BoW and n_grams [44].

3.4.1 Naive Bayes (NB)

Starting with a simple model is a good approach. Hence, training a BL with split data requires only a few lines based on the auto parameters.

3.4.2 Logistic Regression

The logistic regression model uses the log-odds probability of an event comprising binary variables and estimates parameters. The binary variables are "0" or "1," representing "negative" and "positive," respectively. Notably, the logistic model is generalizable past binary via multi-class classification [49]. Regularization and weight balancing

terms are learned using the stochastic gradient descent (SGD) algorithm [50]. Table 1 shows the parameter settings used to perform multi-class logistic regression, where each parameter's value reaches the best result.

Table 1: Multiclass classification parameters.

Model	Parameter	Value
LogisticRegr	logisticregression__ loss	log
	logisticregression__ class_weight	balanced
	logisticregression__ alpha	0.001
	logisticregression__ l1_ratio	0.0
	logisticregression__ random_state	1

3.4.3 Retraining and Evaluation of Logistic Regression

Enhancing the model features requires the following steps:

- Remove noisy features via the lemmatization of the text (i.e., unifying Hamzas, such as "أ" to "ا" and removing preceding Waws "و" and diacritics).
- Add important features, such as bigrams and trigrams.
- Using a TFIDF vectorizer to add weight to more important words and to remove weight from less important (more frequent) ones.

The SGD algorithm is used to set the terms of regularization and for weight balancing. The regularization is decreased (Alpha = 0.0001) because the TFIDF performs the regularization, which controls the weights of features. Sometimes the TFIDF approach will prevent the model from giving the proper weights to the truly important words by over-controlling the feature weights. Imbalanced classes can be controlled during the training stage using class_weight=balanced to penalize losses on minority sections proportional to their underrepresentation. Table 2 shows the parameters used to perform logistic regression for multi-class classification; each parameter's value reaches the best result for retraining and evaluation.

Table 2: Multiclass classification parameters.

Model	Parameter	Value
LogisticRegr	CountVectorizer ngram_range	1,2
	CountVectorizer max_features	250,000
	TfidfTransformer()	-
	SGDClassifier __ loss	log
	SGDClassifier __ n_jobs	-1
	SGDClassifier __ class_weight	balanced
	SGDClassifier __ alpha	0.00001
	SGDClassifier __ l1_ratio	0.0
	SGDClassifier __ random_state	1

3.4.4 ML Interpretability

Model interpretability helps with understanding and correcting the model while explaining the predictions and applying the requirements [51, 52]. For example, the first article in the dataset is as follows:

- The article (in Arabic):

"تعلن شركة اسمنت ام القرى عن توقيع اتفاقية تسهيلات ائتمانية متوافقة مع احكام الشريعة الاسلامية مع بنك الرياض علي النحو التالي:

. - تم توقيع عقد التسهيلات الائتمانية من جانب الشركة بتاريخ 11 10 2017م١

. - بلغت قيمة التسهيلات 50 مليون ريال سعودي٢

- تنتهي فترة التسهيلات بتاريخ 2018 08 02مز٣

. - الهدف من التسهيلات تمويل راس المال العامل للشركة المشتريات الراسمالية٤

. - تم الحصول علي التسهيلات بضمان سند لامر مقدم من قبل الشركة بقيمة 50 مليون ريال سعودي لصالح بنك الرياض٥

" - لا يوجد اطراف ذات علاقة مع العلم بان توقيع هذه الاتفاقية غير ملزم للشركة باي مقابل في حال لم يتم استخدامها.٦

- The article (in English):

  "Umm Al-Qura Cement Company announces the signing of a credit facility agreement compatible with Islamic Sharia provisions with Riyad Bank. This agreement stipulates the following:

  1. The company signed the credit facilities contract on 10/11/2017.
  2. The value of the facilities amounted to 50 million Saudi riyals.
  3. The facility period ends on 02/08/2018.
  4. The facilities' objective is to finance the company's working capital for capital purchases.
  5. The facilities were obtained by securing a bond for an order submitted by the company amounting to 50 million Saudi riyals in favor of Riyadh Bank.
  6. There are no related parties, knowing that signing this agreement is not binding on the company for any consideration if it is not used."

The model categorized the article as positive, negative, or neutral. The actual polarity of this article is positive. Thus, the more and less important features for each class can be identified; the important features are highlighted with dark green, and the features having lower importance are highlighted with dark red. The most important features for each class impact the polarity of articles. The logistic regression model's probability value is 96% positive for this article based on the top important features (see Figure 1).

y=Negative (probability **0.015**, score **-4.177**) top features		**y=Neutral** (probability **0.022**, score **-3.803**) top features		**y=Positive** (probability **0.963**, score **3.181**) top features	
Contribution[7]	Feature	Contribution[7]	Feature	Contribution[7]	Feature
+0.791	مليون ريال	+0.417	2017م	+1.951	توقيع
+0.323	لصالح بنك	+0.281	بان توقيع	+0.725	اتفاقية
+0.298	ريال سعودي	+0.241	بتاريخ 2018	+0.564	الاتفاقية
+0.293	يتم استخدامها	+0.177	توقيع هذه	+0.487	ريال
+0.286	مقابل في	+0.165	11	+0.481	مع
... 59 more positive ...		+0.160	التسهيلات بضمان	+0.461	عقد
... 52 more negative ...		+0.155	استخدامها	+0.393	تمويل
-0.302	تمويل	+0.151	10 2017م	+0.299	راس
-0.337	تم	... 54 more positive ...		+0.202	المال
-0.341	ريال	... 35 more negative ...		+0.188	على
-0.343	مليون	-0.157	مليون	+0.177	التسهيلات
-0.343	بتاريخ	-0.240	المال	... 31 more positive ...	
-0.346	للشركة	-0.300	على	... 48 more negative ...	
-0.360	مع	-0.345	راس	-0.164	مقابل في
-0.363	الرياض	-0.358	تمويل	-0.169	التسهيلات بضمان
-0.381	سعودي	-0.385	مع	-0.171	لم يتم
-0.424	من	-0.480	عقد	-0.176	11
-0.427	<BIAS>	-0.538	الاتفاقية	-0.179	توقيع هذه
-0.458	50	-0.555	التسهيلات	-0.186	بتاريخ 2018
-0.512	بنك	-0.637	ريال	-0.187	استخدامها
-1.014	التسهيلات	-0.726	اتفاقية	-0.286	بان توقيع
-1.345	توقيع	-2.052	توقيع	-0.389	2017م

Figure 1: Logistic regression model probability value for the article based on the top important features.

Based on the visualization generated using the LIME method [53], multiple dataset examples can be generated for each article by deleting and adding the same words, such as deleting the word, ",توقيع" which means "signing," from this article. Then, the generated document is used to predict the new polarity after deleting the word to check the deleted word's impact on the polarity result. Supposing that the resulting probability prediction value is low, the word, "توقيع" or "signing," is found to have a high impact on the polarity, meaning that it is one of the most important features to be used to categorize the article. Therefore, the word has a high weight. Moreover, the deleted word is changed in each iteration, such as "اتفاقية" or "agreement," "المال" or "money," etc. Hence, there are multiple copies of the article with multiple probability prediction values, which produces a list of the most important features for each category based on their weights.

3.4.5 Word-embedding

NLP approaches represent words as indices in a vocabulary while ignoring the relationships between words. However, word embedding computes the distributed representation of any word, representing words in a pattern of

continuous vectors. Word embedding is the basis of applying DL approaches, where DL techniques are used to encode the semantic relationships, linguistic regularities, and forms into an embedding space [54].

In this study, the text is represented using BoW and TFIDF. The BoW method encodes words by representing the number of times these words appear in a document. TFIDF does the same, but it decreases words that are prevalent in the whole corpus. Therefore, BOW and TFIDF do not encode the semantic relationships between a document's words and encode the words' frequency. Thus, text representation captures semantics and meaning via word embedding (i.e., word vectors). Word embedding represents the word with vectors, where the vectors' entries represent sub-meanings [55].

Each word is represented with a vector of dimension 300 as d value, which is the best choice for training time and performance. Maximizing the probabilities is the main goal of this approach to estimate the relations between words. Moreover, the dot product determines the similarity between vectors, where the vectors' higher results are more similar.

The model repeats the steps at each level and stops when the probability stops improving or when the user-specified iterations limit is reached [56]. For instance, the word, "انخفاض" or “declining,” can be represented by the following factors:

array ([1.30537394e-02, -2.46280283e-01, 2.31229201e-01, 1.16268992e-02,

-2.35031307e-01, -3.73380601e-01, 8.87906998e-02, -3.92912924e-01,

-4.94547606e-01, 2.71383613e-01, 1.41343087e-01, 3.21934491e-01,

6.93546534e-02, 8.76310468e-02, -2.00803857e-02, 5.38746603e-02, ...

dtype=float32).

These numbers capture the meaning of words and the relationships between them and their neighbors. Additionally, the vectors that are most like “انخفاض” or “declining” and “ارتفاع” or “rise” are represented as follows:

- “انخفاض” or “Declining”:

[('0.7135729789733887, 'طفيف'), (0.7357336282730103, 'الارتفاع'), (0.8531400561332703, ارتفاع'),

('0.6781542301177979, 'لتحسن'), (0.6994649171829224, 'زياده'), (0.7058970928192139, الانخفاض'),

('0.6687246561050415, 'الانخفاض'), (0.6760700941085815, 'طفيفة'), (0.6767148971557617, يقابله'),

('0.6663761138916016, للارتفاع')].

- “ارتفاع” or “rise”:

[('0.7367196083068848, 'طفيف'), (0.7769777178764343, 'الارتفاع'), (0.8531400561332703, انخفاض'),

('0.7040438652038574, 'لتحسن'), (0.7065216302871704, 'طفيفة'), (0.7124515771865845, الانخفاض'),

('0.7018167972564697, 'تحسن'), (0.7019971609115601, 'زياده'), (0.7036209106445312, للتحسن'),

('0.6998525261878967, للارتفاع')].

In the example above, the Arabic words, "ارتفاع" and "انخفاض," are antonyms describing the upward and downward movement, respectively, of a stock. Word2Vec captures antonyms and synonyms simultaneously because it supposes that semantically related words frequently appear together. Hence, it maximizes the probabilities of predicting a word's context, regardless of anonymity, synonymity, and sentiment [55]. Thus, antonyms are highly related because they frequently appear in the same context.

Accordingly, more than 29,000 dot-product calculations are performed for each vocabulary to obtain the probability in one step, which wastes a long time. Thus, a skip-gram with negative sampling reduces the processing time by calculating a group of binary classification problems together [57].

Table 3 shows the parameter settings used to perform the word-vector model, where each parameter's value reaches the best result. The built vocabulary of the unique words is repeated to capture their relationships. Therefore, the minimum count of the word repetition is equal to three. The sliding window takes the center word as the feature for

each sliding step and the other four as targets. The probability sliding window for any word for prediction has a skip-gram equal to one, and five is the window's size. This word-vector size equals 300, which is the best choice for minimizing the training target and maximizing the performance result. Moreover, the complete dataset required 12 iterations.

Table 3: Multiclass classification parameters.

Model	Parameter	Value
LogisticRegr	Word2Vec__ min_count	3
	Word2Vec__ sg	1
	Word2Vec__ size	300
	Word2Vec__ window	5
	Word2Vec__ iter	3
	Word2Vec__ workers	12
	W2Vectorizer__ Self	-
	W2Vectorizer__ w2v	-
	W2Vectorizer__ pooling	np.mean
	W2Vectorizer__ tokenizer	None
	logisticregression__C	10,000.0
	logisticregression__max_iter	1,000
	logisticregression__multi_class	ovr

3.5 Proposed DL Approaches

For feature extraction and preprocessing, each document (article) is transformed to one vector as input to the neural network. The following steps apply:

1. Learn the vocabulary of all articles (maximum vocabulary size) and set a unique index for each word.
2. Transform each document (article) to a sequence of integers.
3. Unify the (max) length of all sequences.
    a. If all sequences are longer than the maximum length, data are truncated, causing loss of information.
    b. If all sequences are shorter than the maximum length, the data are padded with zeros.
4. Build a major embedding matrix for each word according to its index with dimensions of maximum vocabulary size and embedding vector size. This matrix contains weights randomly distributed based on a normal probability distribution.
5. Input one sequence of integers, where each represents a unique index of the original word in the document. Transform each document to a miner embedding matrix that includes all vocabulary weights based on their indices in the major matrix.
6. The miner matrixes change words' weights based on their contexts with dimensions of maximum length and embedding vector size. The major matrix weights embedding change from a random distribution to a learned distribution based on the document context.
7. Create a mini patch comprising 64 miner matrices of documents for each iteration for one update in the gradient descent to make a better decision until the maximum epoch is finished.
8. After average pooling, generate one matrix patch with dimensions of maximum length and mini patch size.

The lemmatized textual data are used for sequences based on specific parameters (e.g., length and embedding dimension). The maximum features are specified by testing the logistic regression model by changing the vocabulary size (i.e., 50,000, 40,000, 30,000, and 20,000), which decreases the 20,000 features limit. Hence, the maximum feature parameter equals 30,000; chosen because it reflects the BERT model authors select the number of features for experimentation [58]. The maximum length of the longest articles in the data equals 2,057 tokens. Hence, the sequence is not too long, but it overextends the training time. However, calculating the maximum length of most articles using a quantile (0.95) results in 95% of the documents being less than or equal to 512 tokens. Thus, a maximum length of 512 tokens is better than 2,000 because the truncation is too small to prevent information loss.

Moreover, 100 is specified as the embedding dimension size, where a vector of that size represents each word. Thus, preprocessing applications use these parameters during the training and testing stages. Training preprocesses the fitting vocabulary and transforms text into sequences. However, the preprocess validation and testing datasets use the same vocabulary learned from the training set without fitting.

3.5.1 Fast Text Model

The fast text model has a unique initial architecture for the input and output of the neural network, to which LSTM also applies. The input layer enters and produces unknown numbers of mini patches of sequences having a maximum length of 512 words, as is done for preprocessing. The embedding output includes unknown numbers of mini patches of sequences with a maximum length of words and embedding dimensions (vector sizes), which equal 64 according to the fast text design [59]. Then, the maximum pooling takes the maximum vector for one dimension, and each document is transformed to one vector with mini patches of sequences and vector sizes. These patches are normalized to increase the training speed and make the gradient descent work effectively. The first fully connected layer (i.e., dense) is generated using the same input and output vector size. Moreover, dropout and special dropout layers test the effectiveness of each word vector or neuron in the neural network by ignoring and counting some learners for regularization reasons. This process forces the model to learn from each word vector or neuron without relying on specific ones to prevent overfitting [60]. The last fully connected layer produces a vector using the mini patches of sequences and the SA class numbers.

3.5.1.1 Embeddings Learned

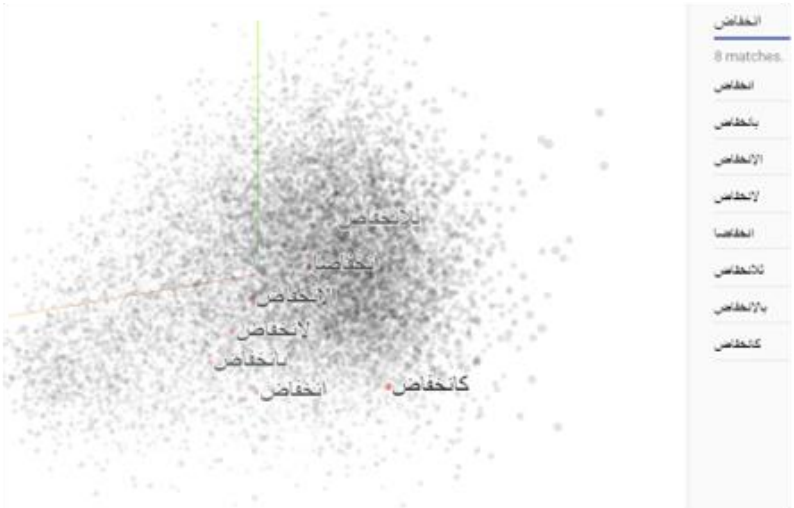

Figure 2: Fast text model TensorFlow embedding representing "انخفاض" or "Declining" related words.

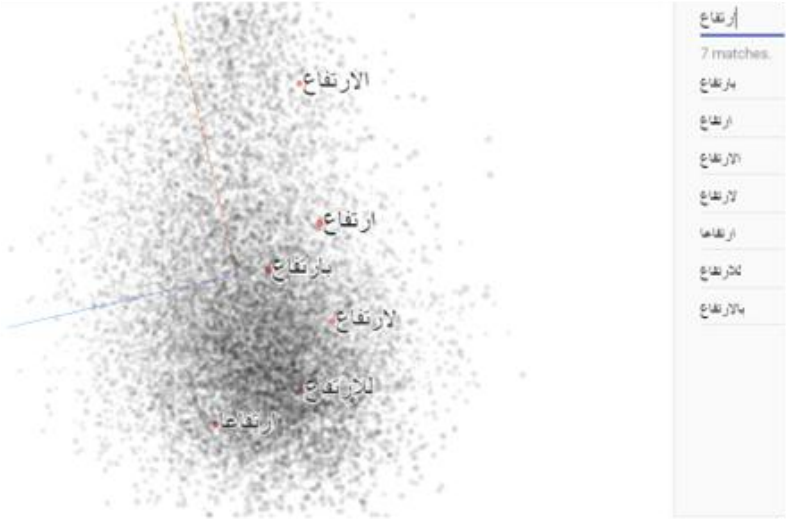

Figure 3: Fast text model TensorFlow embedding representing "ارتفاع" or "rise" related words.

The embedding process is learned jointly alongside classification problem-solving in the fast text model. Thus, the model maximizes the probability of predicting the correct classification of any document. For embedding, the model does not focus on the words in the same context as Word2vec [55, 61]. However, embedding is learned based on the context and the sentiment for frequent words of the same class. For instance, most stock news articles include positive words like "ارتفاع" or "ارتفاعا," which both mean "rise." There are also negative words, such as "انخفاض" or "انخفاضا," which both mean "declining." Hence, an article may contain words with similar meanings. Thus, by choosing a vector in a large space, such as 300 dimensions, the words shown together have similar vectors and clustered together. The labeling of the dataset gives words their meanings. The words, "ارتفاع" or "ارتفاعا," which both mean "rise," are presented as a negative sentiment because both articles have the same sentiments. They are also close to each other in the article. Hence, these two words have similar vectors (see Figures 2 and 3).

In the neural network, vectors are learned by associating them with sentiment. This is called "embedding." Doing so results in a two-dimensional array (max_len x d), where max_len is the article length and d is the vector's dimension.

3.5.1.2 Model Learning

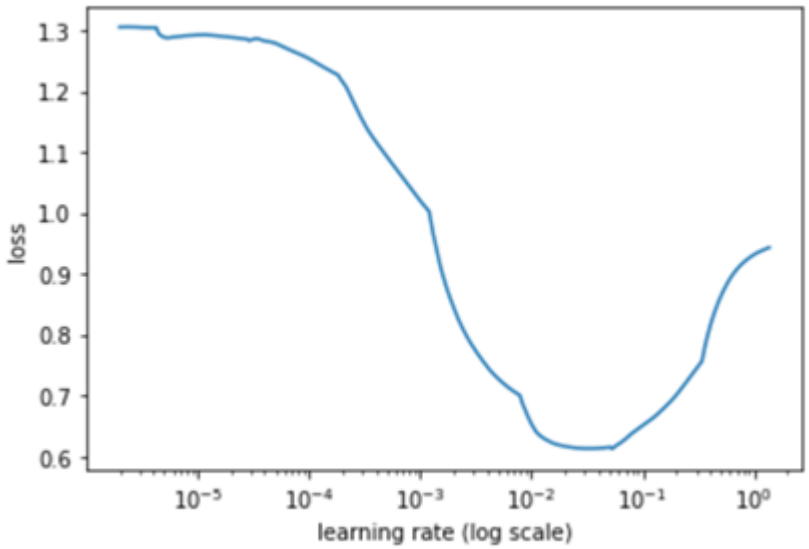

Figure 5: Identifying the maximal LR associated with falling loss.

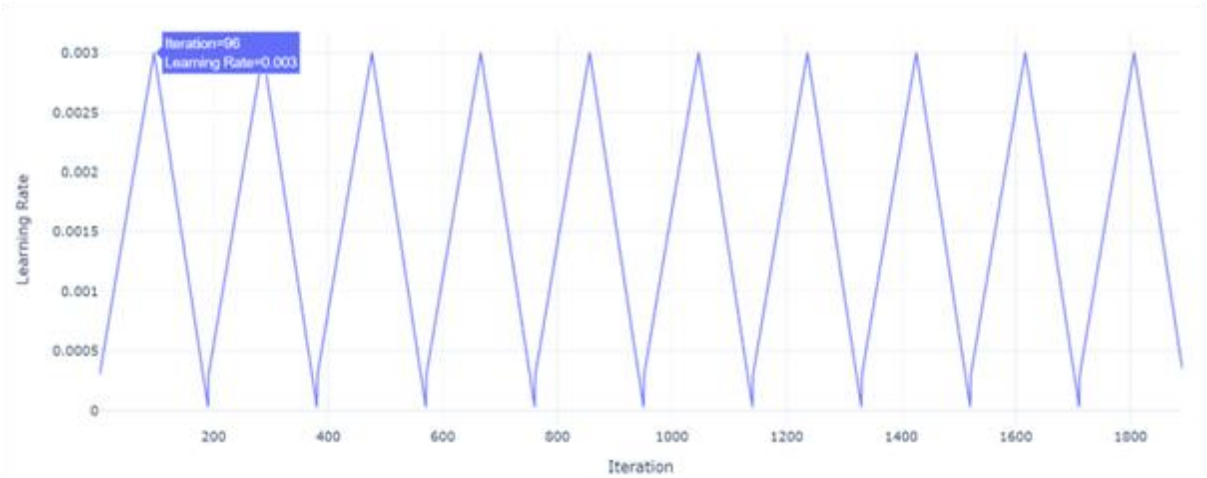

Figure 4: LR schedule (triangle LR policy).

It is optimal to use the ktrain Python library as a wrapper to add more functionality to the TensorFlow model for learning purposes. Hence, training and validation data are used in the learning step with a batch size of 128 [62–64]. Thus, choosing the best LR helps reach the top of the global weight decay from a local one. Therefore, 0.003 is the LR for all iterations, where every mini patch is an iteration. There are 24,078 articles divided by 128 (the mini patch

size), equaling 189 iterations per epoch. From the first iteration until iteration number 96, every new mini patch contains half training data. The LR value gets higher until the highest value is reached. Then, the LR decreases until the lowest value is found in iteration 190 by finishing one epoch to avoid conversion (see Figure 4). This method is called the “triangular LR policy” and is directly applied within ktrain [62].

Hence, “lr_find” is used to locate the maximum LR and track. the loss as the LR increases; it uses “lr_plot” to set the maximum LR related to a falling loss. This is applied by choosing LR with the curve still falling in the range of 10-3 to10-2 (see Figure 5) [62]. However, based on guidelines, the minimum LR (i.e., 3×10-2) is divided by 10. Therefore, 3×10-3 is used (see Figure 5).

3.5.1.3 Model Training

The epoch number is set based on trial and error; thus, any epoch number greater than ten results in better accuracy but worse loss, which leads to overfitting. Therefore, ten epochs are chosen to prevent overfitting and to prevent model performance from worsening during the validation stage. The best epoch based on validation loss turns out to be eight, and the best one based on validation accuracy is the last one. Hence, the validation loss increases until the weight of the last two steps worsens. However, the validation accuracy increases at each step.

3.5.1.4 Inspecting Misclassifications of Fast Text Model

The most misclassified sample in the validation set in the trained model can be identified using the learner—view_top_losses method, which displays the validation examples having the highest loss. Table 4 shows the top three in the validation set of the trained model are related to issues with predicting a neutral class as an actual or predicted class. Articles 434 and 290 are negative, but the model predicts them as neutral. However, article 90 is neutral, but the model predicts it as positive, which is correct. Moreover, some human-mislabeled examples were found, and the model caused other mislabeled examples.

Table 4: Top-three most misclassified examples in validation set from the trained fast text model.

• id:434 | loss:7.65 | true:Negative | pred:Neutral)	
Arabic Article	تعلن الشركة المتحدة للتامين التعاوني عن عدم تمكنها من نشر نتائجها المالية الاولية للفترة المنتهية في 31 12 2010 اثني عشر شهرا بسبب عدم حصولها علي موافقة مؤسسة النقد العربي السعودي ذلك نظرا لعدم اكتمال البيانات المسلمة للمؤسسة بناء علي خطابهم رقم م ت 341 بتاريخ 15 2 1432هـ الموافق 19 1 2011 م سيتم تزويد المؤسسة بالبيانات المطلوبة ليتم الحصول علي خطاب عدم الممانعة الاعلان عن هذه البيانات المالية لاحقا.
English Article	The United Cooperative Insurance Company announces that it was unable to publish its interim financial results for the period ending on 12/31/2010 twelve months due to its failure to obtain the approval of the Saudi Arabian Monetary Agency due to the incomplete data submitted to the institution based on their letter No. MT341 dated 15 2 1432 AH corresponding to 19 1 2011 AD The institution will be provided with the required data in order to obtain a letter of no objection to announce these financial statements later.
• id:290 | loss:6.68 | true:Negative | pred:Neutral)	
Arabic Article	تعلن الشركة الكيميائية السعودية عن انه تم يوم الاحد 29 6 1433هـ الموافق 20 5 2012م ايقاف مذكرة التفاهم الموقعة بتاريخ 15 9 1431هـ الموافق 25 8 2010م مع شركة يارا السويدية بخصوص رغبة الاخيرة في الدخول شريكا في شركة السويس العالمية للنترات سينكو الشركة التابعة في جمهورية مصر العربية ذلك لعدم التوصل الي اتفاق م رض للطرفين ي ذكر ان شركة سينكو تمد الكيميائية السعودية بالمواد الخام التي تستخدمها في منتجاتها مما اضاف للشركة قوة تنافسية كبيرة ساهمت في رفع هوامش الربح للشركة بالاضافة الي استقرار امان الامداد بالمواد الخام رغبة من الشركة في اطلاع المساهمين علي المستجدات تم هذا الاعلان الله الموفق.
English Article	The Saudi Chemical Company announces that on Sunday 29 6 1433 AH corresponding to 20 May 2012 AD, the memorandum of understanding signed on 15 9 1431 AH corresponding to 25 August 2010 AD with the Swedish Yara Company was suspended regarding the latter's desire to enter into a partner in the Suez International Nitrate Company SINCO, the subsidiary company in the Arab Republic of Egypt This is due to the failure to reach a mutually satisfactory agreement. It is mentioned that SINCO supplies the Saudi Chemical Company with the raw materials that it uses in its products, which added to the company a great competitive strength that contributed to raising the

profit margins of the company, also to stabilizing the security of supplying raw materials, as the company desires to inform the shareholders of the developments. God bless this announcement.

3.5.1.5 Interrogating the Model

It is essential to discover the reasons for the model's decisions regarding misclassified examples. According to the LIME method, the input is randomly chosen to discover how the prediction changes [53]. The input article class is examined based on the prediction and actual labeling. This is done to extract the relative importance of several words to the last forecast using a linear interpretable model. The GREEN words support model classification, and the RED ones do not support the final prediction. The shades of color indicate the strength effect of words having the linear model classification or agents it (see Figure 6).\

3.5.2 LSTM Model

The confusion matrix of the LSTM compares the prediction of the validation set and its true target data. It presents the evaluation metrics results of the classification report in the validation set's predicted data. The following figure shows the prediction results for the multiclass classification of LSTM (see Figure 7).

- True Neutral: 82%
- False Neutral: 18%
- True Positive: 88%
- False Positive: 12%
- True negative: 89%.
- False negative: 11%.

4. Results and Discussion

The models' performance can be evaluated in two ways: dataset evaluation and classification model evaluation. Dataset evaluation requires feature and target data, such as a training evaluation. The classification model evaluation uses several approaches for multiple datasets, requiring feature and target data of the dataset map, which contains the training, validation, and testing datasets. This type of evaluation for the classifier generates an evaluation report of multi-class classification, shown in Tables 5–7.

Applying a simple NB model as BL at the beginning of any study is vital for many reasons:

- Testing the dataset quality
- Estimating the first result
- Understanding the model task
- Clarifying the problem dimensions

However, the dataset is more significant than NB allows. This study uses big data, and the training-set size is vast. Thus, logistic regression should get better results. Additionally, discriminative models are better than generative models because they get low asymptotic errors. Therefore, if the number of training examples grows, logistic regression obtains a lower error rate [65].

Tables 5 and 6 compare all results to each other and the annotated and lemmatized datasets. After lemmatization, the feature number reaches 61,523 compared with 72,517 features of BoW. Thus, 10,994 tokens are removed, which helps simplify the model and avoid complexity when generalizing the model for any data type.

N-grams are a weak language representation that treats language as a BoW in any text string [66]. However, the language of the articles is not a set of separated words; each word has a relationship and an affectation to the others in the string. Therefore, BoW wastes the context, whereas TFIDF gives better results because it gives word weights based on their reputation. Hence, the more frequent words in the corpus weigh less. Thus, the least common words

have the highest weight of impact on model performance [9, 47, 48].

In Tables 5 and 6, although applying the N-grams and TFIDF may increase the feature number, it enhances logistic regression performance. The bigram approach generates 596,720 features, and the trigram approach generates 1,724,083. Thus, the model dimensionality must be reduced to maintain the same performance, and the maximum number of features must be limited. TFIDF gives important words greater weight. The highest-level features can be determined using the argument, max_feature=250,000, which achieves the closest result with the fewest features during the training stage. It arranges features based on their importance and chooses the most important ones by decreasing 1,500,000 features using the closest results.

Tables 5 and 6 contain the result of logistic regression with Word2Vec features is lower than its counterpart with BoW and TFIDF. However, this does not mean the latter has better text representation. This result represents an entire article with a single vector by taking the average of all word vectors comprising it. This results in the context being stripped. Hence, the semantic relationships between words cannot be obtained. However, vectors represent the relationships between words. For example, having an article of 100 words, the averaging method takes their average, which ignores their relationships. However, adjacent words have stronger relationships than words farther away. Thus, combining all word vectors shows the power generalization. There is identical accuracy in both training and validation sets, giving a strong indication of good generalization. That is, the model performs just as well with new unseen data as it does with trained data. Hence, by comparing this result to the best-performing model using traditional text representation methods (BoW or TFIDF), there is more than a 10% difference between their accuracy on the training set and validation. This is a clear sign of overfitting of training data, which leads to less generalization. Thus, more advanced neural networks can achieve better accuracy and generalization.

By comparing accuracy results, all results are similar, and there is no overfitting in the data prediction. Overfitting is detectable if the model performs better on the training set than on the testing set [56]. This model avoids overfitting by setting up a train–validation–test split in the scikit-learn library. Thus, the applied logistic regression model with trigrams, TFIDF, and the reduced features parameter can be performed on any random data with good accuracy. Furthermore, the results can be generalized to data outside the sample.

With Arabic classification and prediction, artificial neural networks can achieve high performance [67, 68]. Fast text classifiers are often ranked similarly to DL classifiers in accuracy, fast training, and evaluation. Fast text trains millions of tokens in less than 10 min on a standard multicore processor and classifies half-a-million documents with many classes in less than 1 min. It also scores well with prediction tasks and SA problems [59]. The fast text model was chosen in this study for prediction purposes, owing to its efficacy in solving classification problems. With this model, the embedding process is learned in parallel to the classification process. The model not only focuses on words of the same context, as with the Word2Vec feature, but it also increases the probability of predicting the true classification of any article [55, 61]. Fast text also replaces SoftMax as an activation method for labels having a hierarchical SoftMax activation where each node is a label. This minimizes computation because computing all label probabilities is unnecessary, and the limited number of parameters reduces the training time [59]. These representations are trained in a supervised model with a large dataset to elicit high performance and good embeddings in a short period for classification. As a result, the fast text model achieves 82% accuracy with 30,000 features compared with 84% with the logistical regression model and 250,000 features. Thus, the fast text model achieves a lower accuracy value than the reduced-features logistic regression model with trigrams and TFIDF by almost 2%, but with 220,000 fewer features.

LSTMs use forward propagation alongside backward propagation. It and the RNN backpropagation algorithm update the parameters using the GDS method. However, LSTMs calculate parameters based on the loss function. RNNs enable the design of time-dependent and sequential data functions, such as stock-market classification. However, RNNs suffer from the vanishing gradient problem, which prevents learning long data sequences [69, 70]. The LSTM makes it easier for the RNN to preserve overall information steps. It does not guarantee the elimination of the vanishing gradient, but it provides a simple means of learning long-term dependencies [69, 70]. The LSTM model obtains the same accuracy as the reduced-features logistic regression model with trigrams and TFIDF but with 220,000 fewer features. Therefore, the LSTM model achieves the highest accuracy with the lowest features over the same period, as shown in tables 5 and 6.

Table 7 shows the best classification model evaluation for multiple datasets that requires the feature and target data

from the dataset map, which contains training, validation, and testing sets. This type of classifier evaluation generates an evaluation report of the multi-class classification for the LSTM model, where the positive class gets the highest prediction result.

Table 5: Models performance as features change over the three data splits.

Model	Features	Features' No.	Dataset	Accuracy	Precision	Recall	F1score
NB	-	72,517	Train	0.726929	0.761358	0.726929	0.730638
			Validation	0.710299	0.742229	0.710299	0.715460
			Test	0.683389	0.714801	0.683389	0.686385
Logistic Regression	BoW	72,517	Train	0.904269	0.905350	0.904269	0.904413
			Validation	0.819934	0.822184	0.819934	0.820559
			Test	0.793023	0.803697	0.793023	0.794309
Logistic Regression (Lemmatization)	Lemmatization	61,523	Train	0.898829	0.901909	0.898829	0.899056
			Validation	0.819269	0.824208	0.819269	0.820124
	Bigrams	596,720	Train	0.971260	0.972029	0.971260	0.971285
			Validation	0.836213	0.840327	0.836213	0.836803
	Bigrams with TFIDF	-	Train	0.953692	0.954615	0.953692	0.953785
			Validation	0.840864	0.843669	0.840864	0.841482
	Trigrams with TFIDF	1,724,083	Train	0.966692	0.967336	0.966692	0.966771
			Validation	0.844850	0.847767	0.844850	0.845506
			Test	0.835548	0.837976	0.835548	0.836158
	(Trigrams with TFIDF) Reduced Features	250,000	Train	0.949747	0.950671	0.949747	0.949844
			Validation	0.842193	0.845228	0.842193	0.842853
			Test	0.837542	0.839635	0.837542	0.8381
Logistic Regression (W2vectorize)	W2vec	Matrix of 29,527 words * 300 vector size	Train	0.776020	0.776162	0.776020	0.775908
			Validation	0.767442	0.768661	0.767442	0.767816
Fast Text	Randomly initialized and Jointly Trained Embeddings.	30,000	Validation	0.931244	0.829568	0.831677	0.829568
			Test	0.921928	0.821262	0.822746	0.821262
LSTM		30,000	Validation	0.944712	0.856478	0.858488	0.856478
			Test	0.940196	0.838538	0.839518	0.838538

Table 6: Model performance during the testing stage.

Model Name	Dataset	Accuracy	Precision	Recall	F1-score
Naïve Bayes	Test	0.793023	0.803697	0.793023	0.794309
Logistic Regression	Test	0.793023	0.803697	0.793023	0.794309
Logistic Regression (Lemmatization, Trigrams & TFIDF)	Test	0.837542	0.839635	0.837542	0.8381
Fast Text	Test	0.821262	0.822746	0.821262	0.821396
LSTM	Test	0.847176	0.847087	0.847176	0.847112

Table 7: Model evaluation metrics report of multi-class classification.

Model Name	Class	Precision	Recall	F1-score	Support
LSTM	Negative	0.756831	0.890675	0.818316	311
	Neutral	0.867403	0.820557	0.843330	1148
	Positive	0.872272	0.876209	0.874236	1551
	Accuracy	-	-	0.856478	3010
	Macro avg	0.832169	0.862481	0.845294	3010
	Weighted avg	0.858488	0.856478	0.856671	3010

5. Comparative Analysis

Although the used dataset in different studies approaches is not the same. However, it needs to show an analysis and comparison session over other methods in the closest field and scope with the current study. Different datasets are used in the same domain of Arabic sentiment classification through ML and DL. Therefore, the comparative analysis focuses on the same approaches and basics used by some recent studies regardless of the dataset. Thus, the studies in comparison should focus on the sentiment analysis of MSA textual news based on the polarity classification.

The baseline model is used in comparative analysis because it is widely used in the research community, and it does not need any high technical demands. So, the NB algorithm is implemented to build the study's model and compare all the other studies according to its result. On the one hand, the NB in this study achieved 68.33%, as mentioned in the testing accuracy result. On the other hand, some recent studies used the NB in sentiment classification based on Arabic linguistic features.

The supervised ML is applied for the NB model in the emotion icons in SA for MSA textual news. The supervised NB model obtained 63.79% accuracy results to classify the Twitter data into Positive, Negative, and Neutral classes [34]. Also, in the same domain, the supervised NB model achieved 76.78 in the Arabic classification tweets [71] and 98.2% in the Arabic articles of a balanced dataset [72], (See Table 8).

Table 8: The NB model evaluation comparison analysis of Arabic SA studies.

Study Scope	Dataset Source	No. of Articles	Targeted Language	Used Model	Polarity	Result
Sentiment of Stock Market news.	Tadawul.	30,098 articles.	MSA.	Super-vised ML: Naïve Bayes.	Positive, Negative, and Neutral.	68.33%
SA of Arabic emotions [34].	Twitter's API.	3,000 tweets.				63.79%
SA of Arabic tweets [71].	Twitter's API.	25,000 tweets.				76.78
SA of authorship authentication of Arabic articles [72].	Articles of 14 authors as a balanced dataset.	14,039 articles.				98.2%

Moreover, some studies in the same domain used the LSTM model for SA in the DL of MSA textual data [35,74]. The LSTM approach got 90% in the negative class accuracy and 93% in the positive class accuracy in the hotel reviewing SA [35]. However, the LSTM model in predicting Positive, Negative, and Neutral categories got 64.75 in accuracy outcome [73] (See Table 9).

Table 9: The LSTM model evaluation comparison analysis of Arabic SA studies.

Study Scope		No. of Articles	Targeted Language	Used Model	Polarity	Result
Sentiment of Stock Market news.	Tadawul.	30,098 articles.	MSA.	Super-vised DL: LSTM.	Positive, Negative, and Neutral.	84.71%. Positive: 87%.

				Negative: 82%. Neutral: 84%.
SA of Arabic tweets [73].	Twitter's API.	3,315 tweets.		64.75
SA of Hotel's Arabic reviews [35].	TripAdvisor.	15.000 Reviews.	Positive, and Negative.	Positive: 93%. Negative: 90%.

There are some highlights notes in comparing the used approaches in the closest domain. Firstly, Table 8 shows that the current stock market study has the most significant number of articles and the longest. The maximum features number equals 250,000 words, greater than 280 words for the tweets limit. Although, the NB model obtains a high accuracy in the face of the massive challenges of many articles, high maximum of features, and the complex nature of the stock market news.

Furthermore, Table 9 shows that the current stock market study obtains an above-average result. Although, it is implemented based on the dataset, which has the most massive number of articles with the maximum length of features. Whereas, Twitter's comments do not exceed 280 features, and TripAdvisor comments' limit equals 200 characters for the hotel reviews. Also, predicting the Neutral class category adds a layer of difficulty to the sentiment prediction of Negative and Positive types only. So, the complexity of the Neutral sentiment class may decrease the classification prediction accuracy percentage. However, ignoring the Neutral sentiment class does not result in accurate classification for generalization goals [74].

Thus, the current NB model proves its ability in the generalization for predicting the large and long textual articles of the stock market SA. i.e., applying the training phase based on many features would develop the accuracy results to generalize the dataset's accuracy out of the sample. Also, adding the Neutral class for SA increases the approach efficiency to result in accurate classification.

6. Conclusion

This study compared several supervised model performances for the SA of textual Arabic stock news items. Different features and parameters are applied with hyperparameter tuning. MNB achieved the lowest accuracy value with auto parameters. The logistic regression model with lemmatization, BoW, n-grams, and TFIDF vectorizer enhanced the accuracy with fewer tokens. The logistic regression model achieved a high outcome by determining the lemmatized feature values of 250,000 tokens while using TFIDF and trigrams. Word2Vec counted the relationships between words, but it did not achieve a high result. Thus, some improved processes were implemented to increase accuracy, such as fast text and LSTM. Both models have unique randomly initialized architectures and are jointly trained with learning and training parameters. The fast text model does not surpass the best outcome of the logistic regression model. Still, the number of features is much lower, which results in better performance and good embeddings over a shorter period for the classification task. Therefore, interpretability was assessed, misclassifications were inspected, and models were interrogated to understand prediction errors. Finally, the LSTM model achieved the best result with the highest accuracy and the fewest features. The developed models are useful resources for the Arabic Stock Market news classification prediction. The sentiment models help perceive the risks based on the Stock Market articles by the decision-making supporting analysts' evaluation and investing.

7. Data Availability

The data is available through this link according to the method of data available on the request: www.kaggle.com/dataset/bf24521a3898714597a13efa27a85fa208c96ad49620c787c52720861ddd1e6c.

8. Conflicts of Interest

The authors declares that there is no conflict of interest regarding the publication of this paper.

9. Funding Statement

The authors received no specific funding for this study.

References

[1] Hadi, Sarah K., and Shabbir Ahmad. "Investor sentiment effect on stock returns in Saudi Arabia stock market." *PalArch's Journal of Archaeology of Egypt/Egyptology* 18, no. 13 (2021): 1096-1103.

[2] Zhang, Mr Zhongxia. *Stock returns and inflation redux: an explanation from monetary policy in advanced and emerging markets*. International Monetary Fund, 2021.

[3] Carosia, A. E. D. O., da Silva, A. E. A., & Coelho, G. P. (2024). Predicting the Brazilian Stock Market with Sentiment Analysis, Technical Indicators and Stock Prices: A Deep Learning Approach. *Computational Economics*, 1-28.

[4] Mariani, M. C., Bhuiyan, M. A. M., Tweneboah, O. K., Beccar-Varela, M. P., & Florescu, I. (2020). Analysis of stock market data by using Dynamic Fourier and Wavelets techniques. *Physica A: Statistical Mechanics and its Applications*, *537*, 122785.

[5] Peivandizadeh, A., Hatami, S., Nakhjavani, A., Khoshsima, L., Qazani, M. R. C., Haleem, M., & Alizadehsani, R. (2024). Stock market prediction with transductive long short-term memory and social media sentiment analysis. *IEEE Access*.

[6] BL, S., & BR, S. (2023). Combined deep learning classifiers for stock market prediction: integrating stock price and news sentiments. *Kybernetes*, *52*(3), 748-773.

[7] K. C. Ryding, "A reference grammar of modern standard Arabic.," no. 1, p. 2005, 2005.

[8] AL-dihaymawee, D. T. M., Merzah, A. A., & Ridha, H. M. A. (2024). The Story of Arabic Language: Historical Linguistics Study. *Tasnim International Journal for Human, Social and Legal Sciences*, *3*(1), 572-582.

[9] Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El-Hajj, W., & Shaban, K. (2015, July). Deep learning models for sentiment analysis in Arabic. In *Proceedings of the second workshop on Arabic natural language processing* (pp. 9-17).

[10] Al-Ayyoub, M., Nuseir, A., Alsmearat, K., Jararweh, Y., & Gupta, B. (2018). Deep learning for Arabic El-Masri, M., Altrabsheh, N., & Mansour, H. (2017). Successes and challenges of Arabic sentiment analysis research: a literature review. *Social Network Analysis and Mining*, *7*, 1-22.

[12] Kiadan, J. (2024). The effect of linguistic medium on metaphor directionality: written standard Arabic versus oral colloquial Arabic. *Journal of Cultural Cognitive Science*, *8*(1), 65-78.

[13] Al-Ayyoub, M., Alwajeeh, A., & Hmeidi, I. (2017). An extensive study of authorship authentication of Arabic articles. *International Journal of Web Information Systems*, *13*(1), 85-104.

[14] Das, N., Sadhukhan, B., Ghosh, R., & Chakrabarti, S. (2024). Developing Hybrid Deep Learning Models for Stock Price Prediction Using Enhanced Twitter Sentiment Score and Technical Indicators. *Computational Economics*, 1-40.

[15] Qiu, L., Lei, Q., & Zhang, Z. (2018). Advanced sentiment classification of tibetan microblogs on smart campuses based on multi-feature fusion. *IEEE Access*, *6*, 17896-17904.

[16] Hassanein, A., Mostafa, M. M., Benameur, K. B., & Al-Khasawneh, J. A. (2024). How do big markets react to investors' sentiments on firm tweets?. *Journal of Sustainable Finance & Investment*, *14*(1), 1-23.

[17] Alslaity, A., & Orji, R. (2024). Machine learning techniques for emotion detection and sentiment analysis: current state, challenges, and future directions. *Behaviour & Information Technology*, *43*(1), 139-164.

[18] Pai, P. F., & Liu, C. H. (2018). Predicting vehicle sales by sentiment analysis of Twitter data and stock market values. *IEEE Access*, *6*, 57655-57662.

[19] Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L. T., & Trajanov, D. (2020). Evaluation of sentiment analysis in finance: from lexicons to transformers. *IEEE access*, *8*, 131662-131682.

[20] De Arriba-Perez, F., García-Méndez, S., Regueiro-Janeiro, J. Á., & González-Castaño, F. J. (2020). Detection of financial opportunities in micro-blogging data with a stacked classification system. *IEEE Access*, *8*, 215679-215690.

[21] Bouktif, S., Fiaz, A., & Awad, M. (2020). Augmented textual features-based stock market prediction. *IEEE Access*, *8*, 40269-40282.

[22] Liu, W. J., Ge, Y. B., & Gu, Y. C. (2024). News-driven stock market index prediction based on trellis network and sentiment attention mechanism. *Expert Systems with Applications*, *250*, 123966.

[23] Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S., & Mosavi, A. (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *Ieee Access*, *8*, 150199-150212.

[24] Killeen, P., Kiringa, I., Yeap, T., & Branco, P. (2024). Corn grain yield prediction using UAV-based high spatiotemporal resolution imagery, machine learning, and spatial cross-validation. *Remote Sensing*, *16*(4), 683.

[25] Yadav, A., & Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, *53*(6), 4335-4385.

[26] Rajan, K. V. (2024). Sentiment Analysis of Social Media Using Artificial Intelligence. In *Advances in Sentiment Analysis-Techniques, Applications, and Challenges*. IntechOpen.

[27] Sachin, S., Tripathi, A., Mahajan, N., Aggarwal, S., & Nagrath, P. (2020). Sentiment analysis using gated recurrent neural networks. *SN Computer Science*, *1*, 1-13.

[28] Tembhurne, J. V., & Diwan, T. (2021). Sentiment analysis in textual, visual and multimodal inputs using recurrent neural networks. *Multimedia Tools and Applications*, *80*(5), 6871-6910.

[29] Cai, R., Qin, B., Chen, Y., Zhang, L., Yang, R., Chen, S., & Wang, W. (2020). Sentiment analysis about investors and consumers in energy market based on BERT-BiLSTM. *IEEE access*, *8*, 171408-171415.

[30] Asghar, M. Z., Lajis, A., Alam, M. M., Rahmat, M. K., Nasir, H. M., Ahmad, H., ... & Albogamy, F. R. (2022). A deep neural network model for the detection and classification of emotions from textual content. *Complexity*, *2022*(1), 8221121.

[31] Asghar, M. Z., Subhan, F., Ahmad, H., Khan, W. Z., Hakak, S., Gadekallu, T. R., & Alazab, M. (2021). Senti-eSystem: a sentiment-based eSystem-using hybridized fuzzy and deep neural network for measuring customer satisfaction. *Software: Practice and Experience*, *51*(3), 571-594.

[32] Mostafa, A. O., & Ahmed, T. M. (2024). Enhanced Emotion Analysis Model using Machine Learning in Saudi Dialect: COVID-19 Vaccination Case Study. *International Journal of Advanced Computer Science & Applications*, *15*(1).

[33] Al-Qerem, A., Raja, M., Taqatqa, S., & Sara, M. R. A. (2024). Utilizing Deep Learning Models (RNN, LSTM, CNN-LSTM, and Bi-LSTM) for Arabic Text Classification. In *Artificial Intelligence-Augmented Digital Twins: Transforming Industrial Operations for Innovation and Sustainability* (pp. 287-301). Cham: Springer Nature Switzerland.

[34] Almurqren, L., Hodgson, R., & Cristea, A. (2024). Arabic Text Sentiment Analysis: Reinforcing Human-Performed Surveys with Wider Topic Analysis. *arXiv preprint arXiv:2403.01921*.

[35] Nejjari, M., & Meziane, A. (2020, November). SAHAR-LSTM: an enhanced model for sentiment analysis of hotels' Arabic reviews based on LSTM. In *2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech)* (pp. 1-7). IEEE.

[36] Obied, Z., Solyman, A., Ullah, A., Fat'hAlalim, A., & Alsayed, A. (2021, February). Bert multilingual and capsule network for arabic sentiment analysis. In *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)* (pp. 1-6). IEEE.

[37] "The Saudi Stock Market Tadawul." [Online]. Available: https://www.tadawul.com.sa/wps/portal/tadawul/home/.

[38] "The official home of the Python Programming Language." [Online]. Available: https://www.python.org/.

[39] Szu, W. M., Wang, Y. C., & Yang, W. R. (2015). How does investor sentiment affect implied risk-neutral distributions of call and put options?. *Review of Pacific Basin Financial Markets and Policies*, *18*(02), 1550010.

[40] https://towardsdatascience.com/inter-annotator-agreement-2f46c6d37bf3

[41] https://pubmed.ncbi.nlm.nih.gov/15883903/

[42] T. S. Ng. (2016) "Machine learning," *Stud. Syst. Decis. Control*, vol. 65, pp. 121–151.

[43] Arif, H., Munir, K., Danyal, A. S., Salman, A., & Fraz, M. M. (2016). Sentiment analysis of roman urdu/hindi using supervised methods. *Proc. ICICC*, *8*, 48-53.

[44] Dai, S., Li, K., Luo, Z., Zhao, P., Hong, B., Zhu, A., & Liu, J. (2024). AI-based NLP section discusses the application and effect of bag-of-words models and TF-IDF in NLP tasks. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, *5*(1), 13-21.

[45] Nisa, H. L., & Ahdika, A. (2024). Hybrid Method for User Review Sentiment Categorization in ChatGPT Application Using N-Gram and Word2Vec Features.

[46] Kundu, S. (2024, May). 31 An overview of Stemming and Lemmatization Techniques. In *Advances in Networks, Intelligence and Computing: Proceedings of the International Conference On Networks, Intelligence and Computing (ICONIC 2023)* (p. 308). CRC Press.

[47] Mohammed, M. T., & Rashid, O. F. (2023). Document retrieval using term frequency inverse sentence frequency weighting scheme. *Indonesian Journal of Electrical Engineering and Computer Science*, *31*(3), 1478-1485.

[48] Holla, L., & Kavitha, K. S. (2024). An improved fake news detection model using hybrid time frequency-inverse document frequency for feature extraction and adaboost ensemble model as a classifier. *Journal of Advances in Information Technology*, *15*(2), 202-211.

[49] Jónsdóttir, H. B., & Thorsø, L. W. (2022). *Sentiment Analysis in the Norwegian Stock Market: Predicting Stock Price Movements Using Media Sentiment* (Master's thesis).

[50] Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, *8*(8), 832.

[51] Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

[52] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).

[53] Jain, R., Kumar, A., Nayyar, A., Dewan, K., Garg, R., Raman, S., & Ganguly, S. (2023). Explaining sentiment analysis results on social media texts through visualization. *Multimedia Tools and Applications*, *82*(15), 22613-22629.

[54] Ono, M., Miwa, M., & Sasaki, Y. (2015). Word embedding-based antonym detection using thesauri and distributional information. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 984-989).

[55] Goodfellow, I., Bengio, Y., & Courville, A. (2017). deep learning English version.

[56] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[57] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[58] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

[59] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

[60] Al-Twairesh, N., & Al-Negheimish, H. (2019). Surface and deep features ensemble for sentiment analysis of arabic tweets. *IEEE Access*, *7*, 84122-84131.

[61] Karakaya, O., & Kilimci, Z. H. (2024). An efficient consolidation of word embedding and deep learning techniques for classifying anticancer peptides: FastText+ BiLSTM. *PeerJ Computer Science*, *10*, e1831.

[62] Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

[63] Vinayak, V., & Jyotsna, C. (2023, July). Consumer Complaints Classification using Deep Learning & Word Embedding Models. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-5). IEEE.

[64] Saifullah, K., Khan, M. I., Jamal, S., & Sarker, I. H. (2024). Cyberbullying Text Identification based on Deep Learning and Transformer-based Language Models. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, *11*(1), e5-e5.

[65] Gaydhani, A., Doma, V., Kendre, S., & Bhagwat, L. (2018). Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. *arXiv preprint arXiv:1809.08651*.

[66] Naseem, U., Razzak, I., Khan, S. K., & Prasad, M. (2021). A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Transactions on Asian and Low-Resource Language Information Processing*, *20*(5), 1-35.

[67] Anthony, M., Bartlett, P. L., & Bartlett, P. L. (1999). *Neural network learning: Theoretical foundations* (Vol.

9, p. 8). Cambridge: cambridge university press.

[68] Zaghoul, F. A., & Al-Dhaheri, S. (2013, April). Arabic text classification based on features reduction using artificial neural networks. In *2013 UKSim 15th International Conference on Computer Modelling and Simulation* (pp. 485-490). IEEE.

[69] Ramesh, M. R., Chandrakala, B., Varalakshmi, B., & Shashidhar, M. (2024). Deep Learning-Based Predictive Analytics for Soil Strength and State Forecasting in The Construction Domain.

[70] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), 2222-2232.

[71] Alahmari, S., & Buckley, J. (2015). Sentiment analysis in Arabic tweets. In *IMSCI 2015-9th Int. Multi-Conference Soc. Cybern. Informatics, Proc* (pp. 105-108).

[72] Al-Ayyoub, M., Alwajeeh, A., & Hmeidi, I. (2017). An extensive study of authorship authentication of Arabic articles. *International Journal of Web Information Systems*, *13*(1), 85-104.

[73] Heikal, M., Torki, M., & El-Makky, N. (2018). Sentiment analysis of Arabic tweets using deep learning. *Procedia Computer Science*, *142*, 114-122.

[74] Koppel, M., & Schler, J. (2006). The importance of neutral examples for learning sentiment. *Computational intelligence*, *22*(2), 100-109.