

Retraction

Retracted: News Text Classification Method Based on the GRU_CNN Model

International Transactions on Electrical Energy Systems

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

Copyright © 2023 International Transactions on Electrical Energy Systems. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] L. Deng, Q. Ge, J. Zhang et al., "News Text Classification Method Based on the GRU_CNN Model," *International Transactions on Electrical Energy Systems*, vol. 2022, Article ID 1197534, 11 pages, 2022.

Research Article

News Text Classification Method Based on the GRU_CNN Model

Lujuan Deng,¹ Qingxia Ge,¹ Jiaxue Zhang,¹ Zuhe Li ^{1,2}, Zeqi Yu,¹ Tiantian Yin,¹
and Hanxue Zhu¹

¹School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, Henan, China

²Henan Key Laboratory of Food Safety Data Intelligence, Zhengzhou University of Light Industry, Zhengzhou 450002, Henan, China

Correspondence should be addressed to Zuhe Li; zuheli@zzuli.edu.cn

Received 25 June 2022; Revised 25 July 2022; Accepted 16 August 2022; Published 31 August 2022

Academic Editor: Raghavan Dhanasekaran

Copyright © 2022 Lujuan Deng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The convolutional neural network can extract local features of text but cannot capture structure information or semantic relationships between words, and a single CNN model's classification performance is low, whereas GRU can effectively extract semantic information and global structure relationships of text. To address this problem, this paper proposes a news text classification method based on the GRU_CNN model, which combines the advantages of CNN and GRU. The model first trains word vectors as the embedding layer with the Word2vec model and then extracts semantic information from text sentences with the GRU model. Following that, this model employs the CNN model to extract crucial semantic information features and finally completes the classification through the Softmax layer. The experimental results reveal that the GRU_CNN hybrid model outperforms single CNN, LSTM, and GRU models in terms of classification effect and accuracy.

1. Introduction

With the rise of various news clients such as NetEase News, Toutiao, and Tencent News, news resources have become readily available. However, how to accurately locate information from existing news resources and find information that can meet their own needs is becoming a problem worth studying at present. In addition, news classification is the basis of news website classification navigation, personalized recommendation, etc. Therefore, efficient and accurate screening and classification of news content have very important practical significance, so how to achieve efficient and accurate classification has become a nodus [1].

Text classification is indeed the fundamental work among many natural language processing applications, and it can support tasks including topic classification [2], sentiment analysis [3], and question answering system [4]. At present, there have been many studies on text classification. The early research was mainly based on manual rules and knowledge engineering methods, which require manual processing of data and the participation of experts. The rule-

makers need to have enough knowledge and understanding of a certain category field, but this method necessitates a significant amount of manpower, material wealth, and financial support. What is more, the subjectivity of manual rule design will also cause differences in classification. Text categorization in natural language processing is gradually being substituted by automatic classification by machine rather than manual classification since the emergence of machine learning. Support vector machine (SVM) [5], K-nearest neighbor (KNN) [6], Naive Bayes (NB) [7], logistic regression (LR) [8], and decision tree [9] are a few machine learning algorithms that are commonly employed in text classification. However, such traditional machine learning models usually require manually designed features and feature combinations, which are highly subjective and task-specific. At the same time, these models can only extract shallow features. The model is prone to overfitting and inefficiency caused by the large amount of data, which ultimately affects the classification accuracy. Due to the popularity of deep learning, many neural network models have been extensively applied in the production of text

categorization in recent years. Deep neural network models have superior feature extraction ability to typical machine learning algorithms, making them ideally suited for applications in the area of text categorization [10].

The primary goal of this study is to address the problem of Chinese news text classification using deep learning models. As GRU can efficiently extract the semantic information and global structural information of text, and as CNN cannot effectively extract the structural information of text and the semantic relationship of sentences, the classification accuracy of a single CNN model is low. Therefore, this paper combines the benefits of the CNN model and the GRU model to propose the GRU_CNN model for the classification of news text.

2. Related Work

Among deep learning classification models, there are two main types of common categories, namely, convolutional neural network and recurrent neural network, which have many applications in text categorization.

Kim [11] first proposed a text categorization model based on CNN in 2014, which converts words into fixed-length word vectors through Word2vec as the input of CNN and then uses multisize convolution to check word vector convolution. Finally, pooling and classification are performed. The key advantage of CNN is that it has a quick training speed and can efficiently extract local text features, but the pooling layer will lose a lot of vital information and overlook the association between the local and the whole.

After Kim proposed the model, Zhang and Wallace [12] also proposed a text categorization method based on CNN and performed numerous comparative experiments under various hyperparameter settings. Besides, they also offered parameter tuning advice and have some experience with hyperparameter configuration.

Johnson and Tong proposed the deep pyramid convolutional neural network (DPCNN) [13], which has low complexity and excellent categorization power. This model primarily investigates word-level CNN and improves accuracy by deepening the depth of the network. DPCNN can extract long-distance text dependencies by constantly deepening the network, but the computational complexity will increase as well, which makes practical applications problematic. Yao et al. proposed a graph-based convolutional neural network (GCN) [14], which is more effective for small datasets categorization. GCN's limitations of flexibility and scalability are its main drawbacks.

Mikolov et al. [15] employed the recurrent neural network model to achieve text categorization. RNN can accommodate inputs of arbitrary length, and the model size does not grow with the input length. However, the RNN model relies on learning features for a long time, which is prone to the problem of gradient dispersion.

Hochreiter and Schmidhuber proposed a long short-term memory network (LSTM) [16] based on RNN, which makes up for the classic RNN's poor learning performance for long-distance sentences as well as the problem of gradient disappearance or gradient explosion. However, LSTM

requires a lot of calculations during training due to its numerous parameters and the complexity of the calculations between each gate.

Chung et al. proposed the GRU model [17] on this basis, which simplified the structure and training parameters of LSTM and improved the efficiency of training. However, GRU is unable to compute in parallel and is still unable to resolve the gradient vanishing problem entirely.

Graves and Schmidhuber [18] applied the bidirectional LSTM model for the first time to solve the categorization problem and achieved better classification results than the unidirectional LSTM model. However, compared with LSTM, BiLSTM has more parameters and is more difficult to calculate. Cao et al. [19] utilized the BiGRU model to categorize Chinese text by synthesizing the context of the article. The model is simple, with a few parameters and fast convergence.

Li and Dong [20] employed CNN to extract text's local features and the BiLSTM to extract text's global features in order to fully capitalize on the benefits of the features obtained by the two models and enhance the model's ability to classify data.

In conclusion, a single neural network model often has the problem of low classification accuracy. This research provides a news text categorization strategy based on the GRU_CNN model in order to get a better classification performance. This model extracts more accurate text features, and its effectiveness has been demonstrated through experiments.

3. GRU_CNN Model

The whole process of the GRU_CNN hybrid model is as follows: Firstly, input the news text data, preprocess the news text, and train the text using the Word2vec model to produce a word vector comprising the text's overall information. Secondly, to extract contextual semantic information from the text, the vector representation of words obtained by Word2vec is fed into the GRU model for semantic information extraction. Then, the output results of the GRU model are input into the CNN model for further semantic feature extraction. Finally, the final word vector is input to the Softmax layer for text classification. Figure 1 depicts the GRU_CNN model's structure.

3.1. Word2vec Embedding Layer. This paper uses the classic Word2vec [21] model for word embedding, and the Word2vec of the Gensim toolkit is used to train word vectors, which is simple and fast. The Word2vec model is divided into three layers: input, hidden, and output. The model's input is a one-hot vector; that is, the textual information is represented by $\{x_1, x_2, \dots, x_V\}$. Assuming the size of the vocabulary is V and the dimension of each hidden layer is N , the input to the hidden layer is represented by a matrix W of size $V \times N$. Similarly, word vectors can be obtained by connecting the hidden and output layers via a $N \times V$ matrix W' . Figure 2 depicts the Word2vec model's structure.

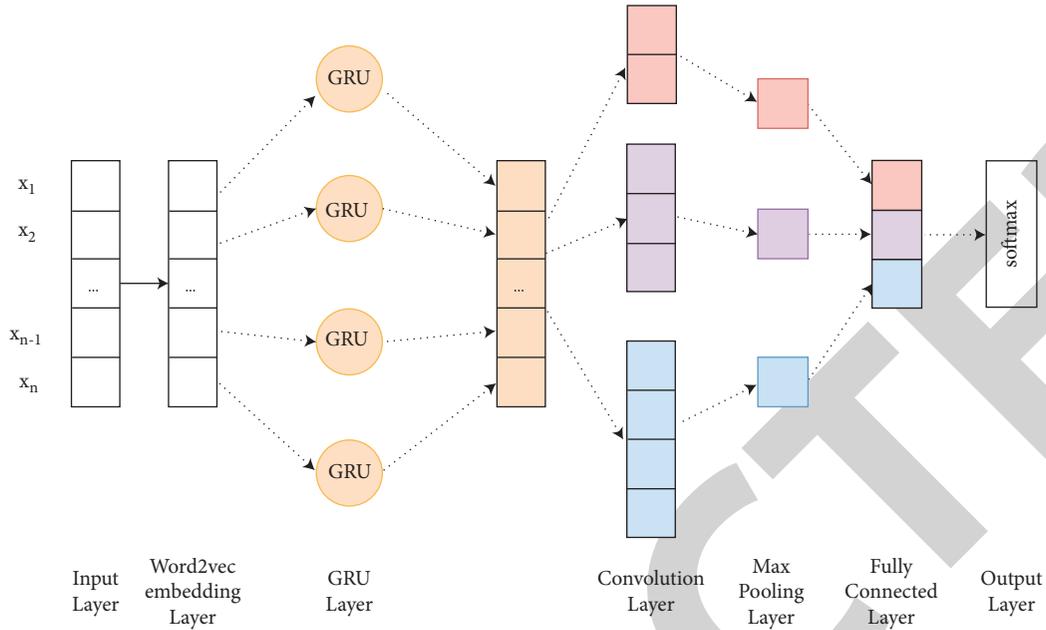


FIGURE 1: GRU_CNN model.

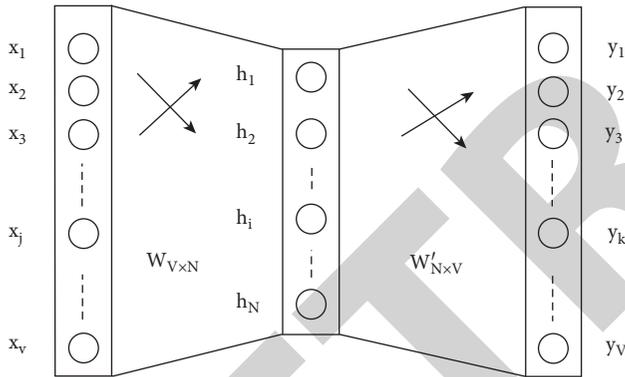


FIGURE 2: Word2vec model.

The Word2vec model is analogous to a simple neural network that mainly includes CBOW and Skip-Gram models. The CBOW model primarily foretells the present value based on the context, which is equivalent to taking a word out of context and letting you infer its meaning. The Skip-Gram model primarily forecasts the context using the current word, which is the same as giving you a word and asking you to predict which words might come before and after it. CBOW is better suited to smaller datasets, whereas Skip-Gram performs better in large corpora. Figure 3 depicts the structure of the CBOW and Skip-Gram models. This paper uses the Skip-Gram model for training, and Skip-Gram is more sensitive to low-frequency words.

3.2. GRU Layer. Recurrent neural network (RNN) [22] has been proved by many scholars to be suitable for deep learning tasks of text processing. But when faced with longer texts, it is difficult for the gradient of the letter sequence to backpropagate to the earlier semantic information, and it is

prone to the problem of gradient dispersion. In order to overcome the shortcomings of RNN, many variant models of the recurrent neural network have been proposed, among which the two most popular models are LSTM and GRU.

The input gate, memory unit, forget gate, and output gate are the four main components of an LSTM. The important features of the text are retained through the “memory” and “forgetting” of the input vector, while the largely pointless stuff is removed. However, as the volume of texts grows, the issue that LSTM network training takes a long time to complete is gradually revealed. This is because of the high number of parameters and the relatively difficult calculation between the gates. To address the shortcomings of the LSTM model, a simpler neural network model GRU is presented. The GRU model combines the forget gate and the input gate into an update gate based on LSTM.

As a variant of RNN, gated recurrent unit network (GRU) uses a reset gate and update gate to decide how to discard and update information, which successfully solves the problem of long-term dependence. GRU contains fewer parameters and is easier to calculate than LSTM, which can enhance training efficiency to a certain level while achieving the same effect as LSTM. This paper uses the GRU model to obtain the context semantic information of text. Figure 4 depicts the GRU model’s structure.

The update gate is used to determine the impact of the previously hidden layer state on the current layer. The larger the value in the update gate, the greater the impact on the current layer at the previous moment. Formula (1) illustrates the update gate’s computation process.

$$U_m = \sigma(W_U \bullet [h_{m-1}, x_m]). \quad (1)$$

The reset gate is used to remove the invalid information at the last moment. The smaller the reset gate value, the more invalid information is deleted. Formula (2) illustrates the reset gate’s computation process.

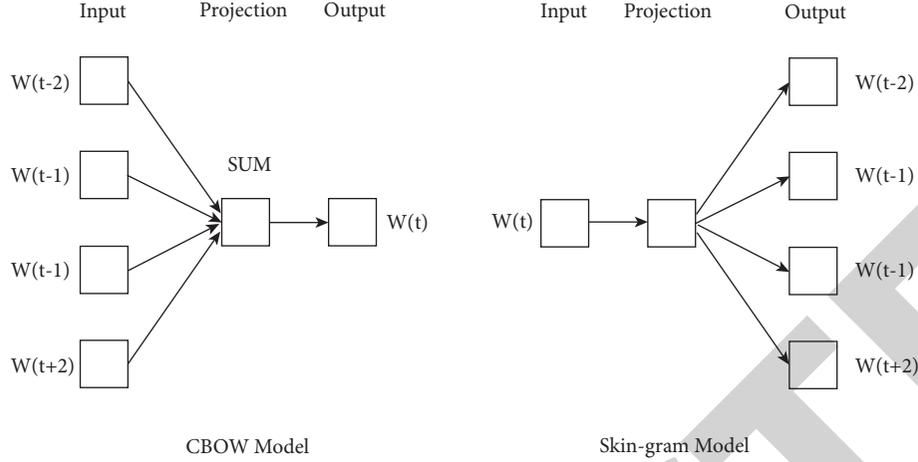


FIGURE 3: CBOV model and Skip-Gram model.

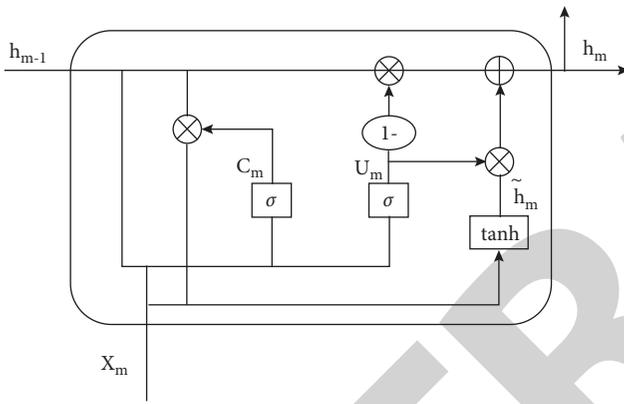


FIGURE 4: GRU model.

$$C_m = \sigma(W_C \bullet [h_{m-1}, x_m]). \quad (2)$$

The current state computation method is presented in the following formulas:

$$\tilde{h}_m = \tanh(W \bullet [C_m \odot h_{m-1}, x_m]), \quad (3)$$

$$h_m = U_m \odot \tilde{h}_m + (1 - U_m) \odot h_{m-1}. \quad (4)$$

Among them, W_U , W_C , and W represent the weight matrix of the GRU, x_m is the input data, h_m is the current hidden state of the model, the input of the previous state is h_{m-1} , \tilde{h}_m is the candidate active state, U_m and C_m represent update and reset gates, respectively, \odot represents the Hadamard product, that is, the elements of the corresponding position are multiplied, and σ represents the sigmoid function.

3.3. CNN Layer. Convolutional neural network (CNN) was originally employed to solve image processing problems and later applied to natural language processing. It has grown in popularity as among the most widely utilized deep learning models. Among text categorization models based on deep learning, the convolutional neural network is a shallow

neural network model with good performance and fast training speed. Figure 5 depicts the CNN model's structure.

The function of the convolution layer is to extract local features within a set window range and use the convolution kernel to convolute the input vectors to obtain the feature output.

The pooling layer further integrates the local features of the text extracted by the convolution layer, shrinks the feature map, and enhances the computation speed. At the same time, it obtains global information from the feature values, improves the robustness of the extracted features, and controls overfitting issues. Generally, two approaches are used: Mean pooling and Max pooling. Mean pooling is to calculate the average value of column vectors, and Max pooling is to directly extract the maximum level of column vectors. This article uses the Max pooling operation.

The fully connected layer further abstracts its eigenvalues, taking all the outputs of the pooling layer as input, where each neural unit is connected to each unit of the pooling layer and transforms the pooling layer vector into a long vector by activating function ReLu, mapping the text from the feature space to the marker space.

The CNN model is utilized to extract semantic information features in this paper. After obtaining the contextual semantic information extracted from the GRU layer, it is input to the CNN layer for extracting semantic features.

4. Experiment and Analysis

4.1. Dataset Introduction. The news text dataset used in this article is Cnews, which is a subset of the THUCNews dataset. 10 of these categories were used, each with 6,500 entries, for a total of 65,000 news data. The specific categories and divisions of Cnews dataset are shown in Table 1.

4.2. Comparison Experiment Design. In order to test the hybrid neural network model's effectiveness, several classical models are selected for comparative experiments.

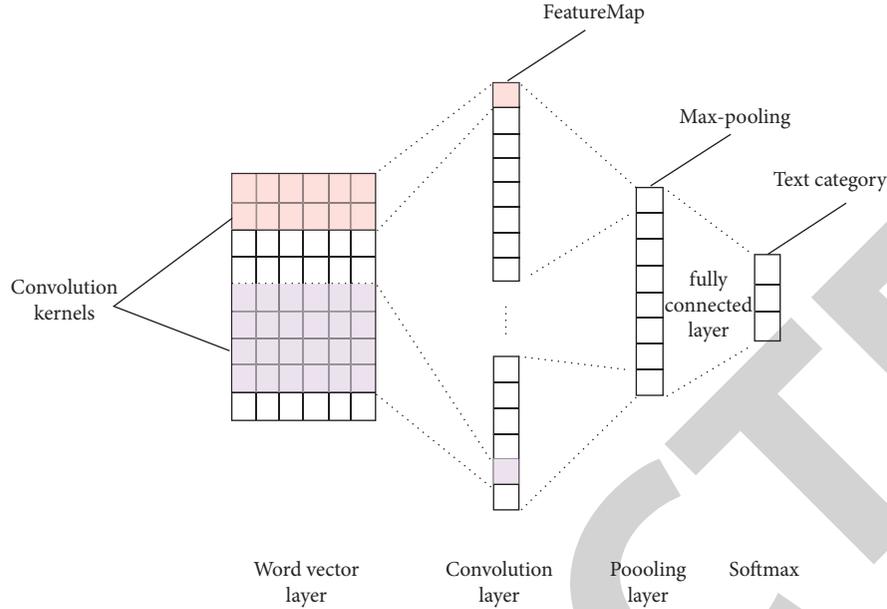


FIGURE 5: CNN model.

TABLE 1: Distribution of Cnews dataset.

Data category	Training set	Test set	Validation set
Sports	5000	1000	500
Finance	5000	1000	500
Real estate	5000	1000	500
Home	5000	1000	500
Education	5000	1000	500
Technology	5000	1000	500
Fashion	5000	1000	500
Politics	5000	1000	500
Games	5000	1000	500
Entertainment	5000	1000	500

TABLE 2: Parameters.

Parameter names	Values
Convolution kernel size	(2, 3, 4)
Word vector dimension	100
The number of hidden units	128
Dropout	0.5
learning_rate	0.001
Optimizer	Adam
Batch size	64
Epoch	6

- (1) LR: logistic regression is a machine learning categorization method for estimating the likelihood of something.
- (2) NB: Naive Bayes is a machine learning categorization algorithm that leverages Bayes' theorem.
- (3) CNN: it is the most basic convolutional neural network for text categorization.
- (4) LSTM: it is the most basic long and short memory network for text categorization.
- (5) GRU: it is the most basic gated recurrent unit network for text categorization.
- (6) LSTM + CNN: firstly, the LSTM model is utilized to extract the contextual semantic information from the input text, and then the CNN model is used to extract the feature of the semantic information. Finally, complete classification through the Softmax layer.

4.3. *Parameters.* The experiment presented in this paper is based on the TensorFlow framework, and the model's parameter settings determine how well the model trains. Table 2 displays the GRU_CNN model's specific parameter settings.

4.4. *Evaluation Index.* The experiment employs the accuracy of text categorization as its evaluation index in order to examine the effectiveness of the model put forth in this study. The accuracy rate indicates the proportion of correctly predicted samples to the total samples, and the calculation formula is shown in

$$\text{accuracy} = \frac{1}{K} \sum_{i=1}^K |y_i = \tilde{y}_i|. \quad (5)$$

In formula (5), K represents the sample capacity of the sample set, y_i represents the label of the sample x_i , and \tilde{y}_i is used as the prediction result. When y_i and \tilde{y}_i are consistent, the value of the precision rate is 1; otherwise, the value is 0.

4.5. *Analysis of Experimental Results.* The error and accuracy of GRU_CNN model obtained by training on the training set are shown in Figure 6. During the experiment, the accuracy and loss values are output on the training set and validation set every 100 batches and recorded and saved. Continuously train the model, saving the best-performing model for testing on the test set. We can see

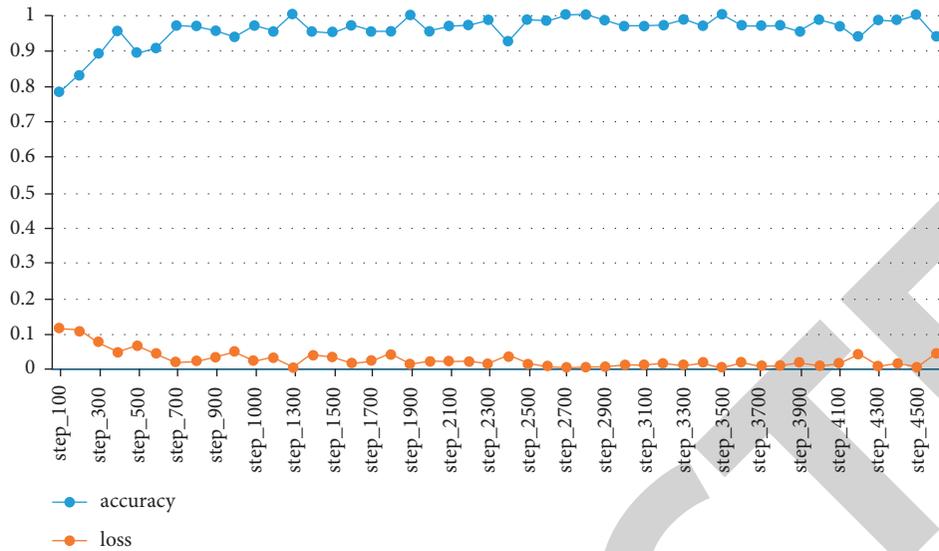


FIGURE 6: Error and accuracy results of model training.

that the model's accuracy on the training set has achieved good results.

In the experiment, the number of epochs can affect the model's performance. This study discusses the model classification effect under different epoch numbers through experiments. Table 3 and Figure 7 illustrate the experimental outcomes. The experimental results show that the number of epochs required by different models to achieve the optimal effect varies. The GRU_CNN model reaches the highest accuracy rate when the epoch number is 6. The numbers of epochs at which CNN, LSTM, GRU, and LSTM+CNN achieve the best results are 3, 2, 4, and 5, respectively. Figure 6 shows that the model's performance first rises and then falls with the increase of the number of epochs, which means that, before reaching the best effect of the model, the ability of model learning characteristics will continue to increase as the number of epochs increases. After the best results, the model will show an overfitting phenomenon so that the model's performance will decrease.

During the training process of the model, the dropout method is introduced into the model in this experiment. The value of dropout is also an important parameter, and an appropriate value can enable the model to converge better, avoid overfitting, and improve its performance. Therefore, we choose different dropout values for training. The dropout values set in this experiment are [0.2, 0.3, 0.4, 0.5, 0.6, 0.7], respectively. The optimal dropout values are selected based on the training results of the model. Table 4 and Figure 8 illustrate the experimental outcomes. The findings show that only the LSTM model has the best effect when the dropout value is 0.7, and the other models have the best effect when the dropout value is 0.5. At this time, the dropout value can effectively prevent the model from overfitting on the premise of ensuring accuracy. Therefore, the dropout value of the model in this paper is set to 0.5.

During the course of updating parameters by gradient backpropagation in the neural network, the optimizer used

in this experiment is Adam. The Adam optimization algorithm has high computational efficiency and fast convergence speed. In order to make the algorithm more efficient, different learning rate values are selected for experiments. Table 5 and Figure 9 illustrate the experimental outcomes. As shown by the experimental results, the model's accuracy is highest when Adam's corresponding learning rate is 0.001. Therefore, the learning rate of the Adam optimizer in this case is 0.001.

Word vector dimension is also an important parameter. The larger the dimension, the more feature information the model can learn, and the greater the risk of the model overfitting. However, the lower the word vector dimension, the greater the risk of the model underfitting. Therefore, the appropriate word vector dimension is also important for the model's training effect. This paper selects different dimensions of the word vectors to train. Table 6 and Figure 10 illustrate the experimental outcomes. As can be observed from the experimental findings that all models' accuracy reaches the highest in dimension 100, this paper chooses 100 dimensions to experiment.

Figure 11 illustrates the experimental results of all models on the Cnews dataset. For proving the performance of the GRU_CNN model, several classical models are selected for comparative experiments. In the classification model based on machine learning, logistic regression (LR) and Naive Bayes (NB) are selected for comparative experiments. In the classification model based on deep learning, this paper selects a single CNN, LSTM, and GRU model, as well as a hybrid model LSTM_CNN for comparative experiments. The experimental results are selected to compare the best effect of each model.

As can be noticed from the outcomes of the experiments, the GRU_CNN model has the best classification effect and the highest accuracy on the Cnews dataset. Among the traditional machine learning classification algorithm models, the classification accuracy of logistic

TABLE 3: Table of epoch experimental results.

Epoch	CNN (%)	LSTM (%)	GRU (%)	LSTM + CNN (%)	GRU + CNN (%)
1	93.96	94.42	95.5	95.28	96.48
2	95.26	96.3	94.62	95.56	96.94
3	96.92	96.06	94.54	97.26	97.22
4	96.24	95.42	95.92	97.24	97.04
5	96.84	96	94.62	97.78	97.1
6	96.22	96.24	95.88	97.18	97.86
9	96.91	95.68	94.3	97.4	97.2
10	94.26	94.8	94.28	97.56	96.84

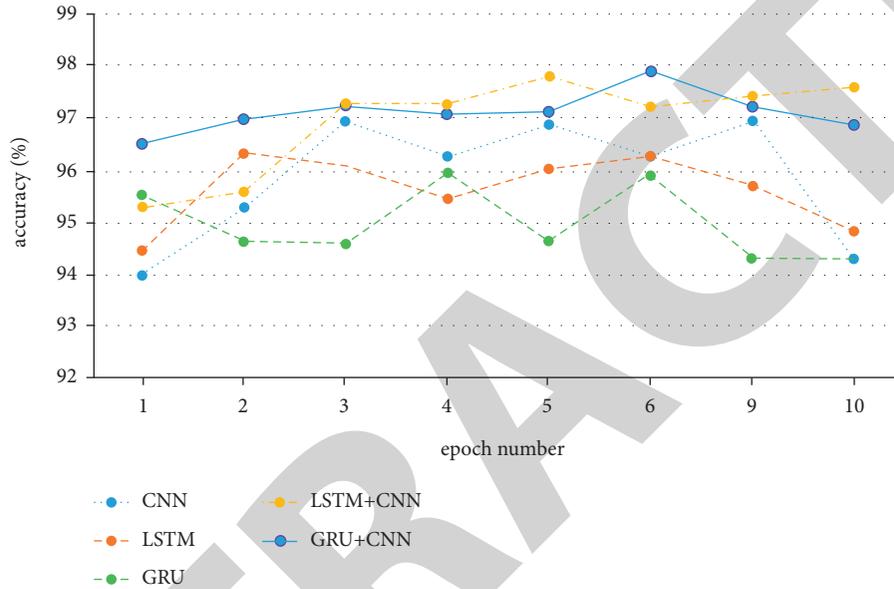


FIGURE 7: The accuracy curve of each model under different epoch numbers.

TABLE 4: Table of dropout experimental results.

Dropout	CNN (%)	LSTM (%)	GRU (%)	LSTM + CNN (%)	GRU + CNN (%)
0.2	95.94	95.72	95.88	96.32	94.32
0.3	96.42	94.3	95.54	97.28	95.92
0.4	95.16	95.3	94.64	95.98	95.94
0.5	96.92	96.3	95.92	97.78	97.86
0.6	95.14	95.38	93.18	97.42	97.38
0.7	96.44	96.78	93.4	97.38	97.52

regression (LR) on the Cnews dataset is superior to that of the Naive Bayes model (NB), and the classification accuracy of all deep learning models is higher than that of traditional machine learning classification models, which is also the reason why deep learning models are popular nowadays. In the deep learning model, the hybrid neural network model has a greater categorization accuracy than the single deep learning model, and the CNN model has a better categorization effect than the LSTM and GRU models, as shown by the results. Although the classification accuracy of the GRU_CNN model and the LSTM_CNN model is not significantly different, the GRU model has fewer parameters and is easier to calculate than the LSTM model. As a result, when everything is taken into

account, the GRU_CNN model this study proposes performs better.

4.6. Ablation Experiment. In order to verify the effectiveness of using Word2vec to train word vectors as the embedding layer, this paper first selects CNN, LSTM, and GRU as the comparative experimental model and compares the experimental results of Word2vec_CNN, Word2vec_LSTM, and Word2vec_GRU. The experimental results are shown in Table 7.

It can be seen that using Word2vec to train the word vector as the embedding layer can effectively optimize the vector representation of the input text, so as to obtain a better training

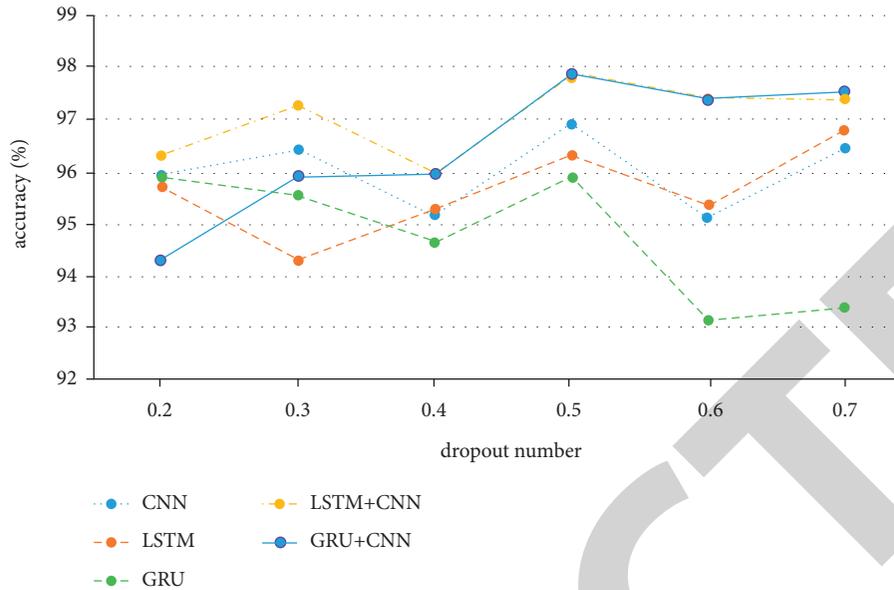


FIGURE 8: The accuracy curves of each model under different dropout values.

TABLE 5: Table of learning rate experimental results.

learning_rate	CNN (%)	LSTM (%)	GRU (%)	LSTM + CNN (%)	GRU + CNN (%)
0.1	92.1	10	10	10.02	10
0.01	94.44	93.94	10	96.18	10
0.001	96.92	96.78	95.92	97.78	97.86
0.0001	90.58	92.06	91.88	95.46	95.08
0.00001	20.92	80.86	82	85.16	83.06

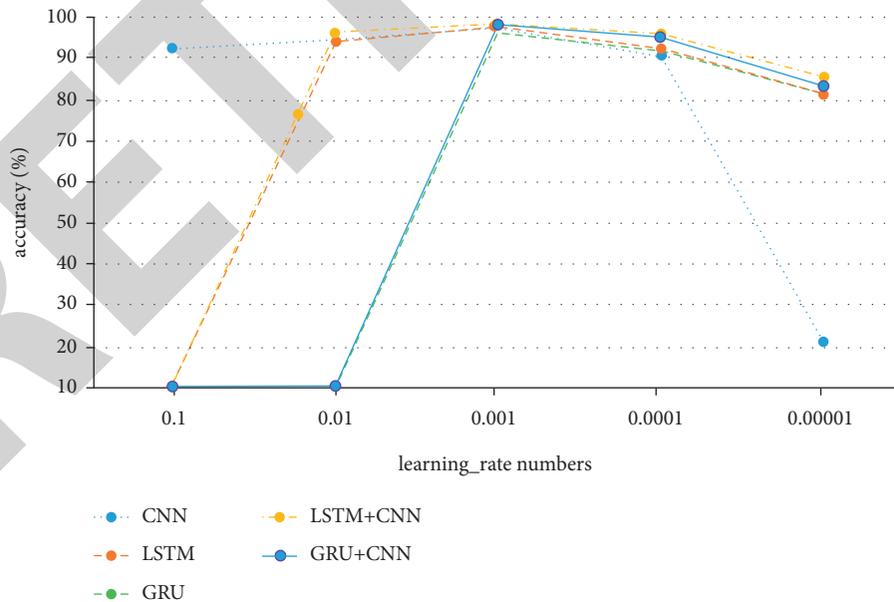


FIGURE 9: The accuracy curves of each model under different learning rates.

effect. Compared with CNN, the accuracy of Word2vec_CNN has increased by 1.23%. Compared with LSTM, the accuracy of Word2vec_LSTM has increased by 3.32%. Compared with GRU, the accuracy of Word2vec_GRU has increased by 2.33%.

Next, in order to verify the effectiveness of the Word2vec_GRU_CNN model proposed in this paper in the text classification task, the Word2vec_CNN and Word2vec_GRU models that performed relatively well in the above

TABLE 6: Table of embedding dimension experimental results.

embedding_dim	CNN (%)	LSTM (%)	GRU (%)	LSTM + CNN (%)	GRU + CNN (%)
50	95.56	94.68	93.89	95.86	95.98
100	96.92	96.78	95.92	97.78	97.86
150	94.30	94.54	91.68	95.36	95.34
200	94.88	93.78	92.68	94.46	95.10
250	95.80	95.42	92.76	95.74	95.54
300	94.78	93.68	92	96.30	92.06

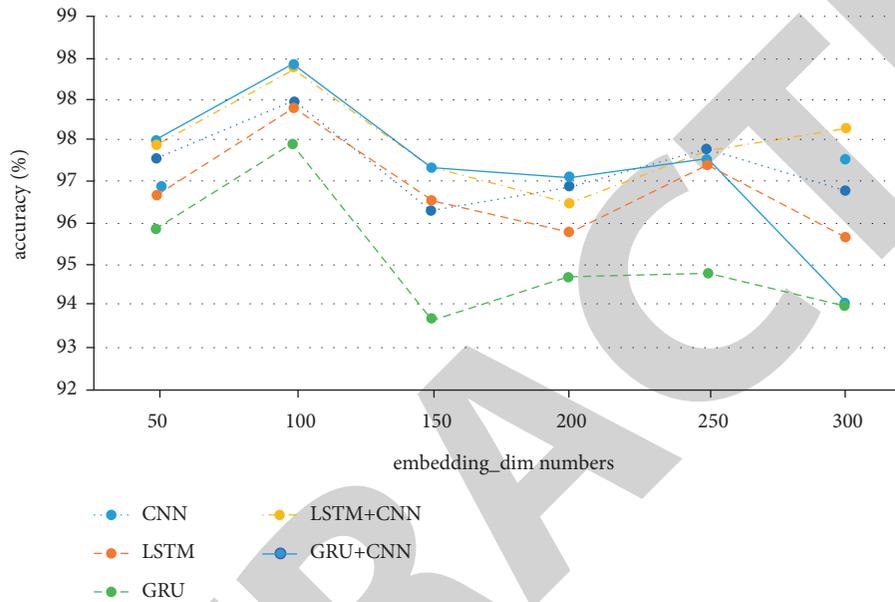


FIGURE 10: The accuracy curves of each model under different word vector dimensions.

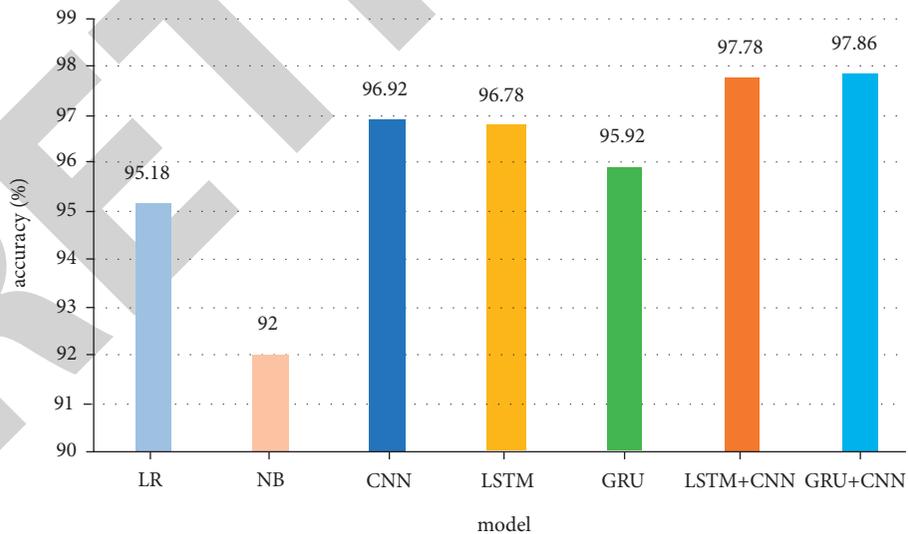


FIGURE 11: Classification accuracy of each model.

experiments were selected for comparison. Experiment with the same parameter settings and the experimental results are shown in Table 8.

From the experimental results in Table 8, it can be seen that, compared with Word2vec word embedding followed by only one layer of training network, the GRU_CNN model

TABLE 7: The first group of ablation experiment results.

Model	Accuracy (%)
CNN	95.69
LSTM	92.98
GRU	93.59
Word2vec_CNN	96.92
Word2vec_LSTM	96.3
Word2vec_GRU	95.92

TABLE 8: The second group of ablation experiment results.

Model	Accuracy (%)
Word2vec_CNN	96.22
Word2vec_GRU	95.88
Word2vec_GRU_CNN	97.86

proposed in this paper effectively improves the accuracy of text classification. Compared with the Word2vec_CNN and Word2vec_GRU models, the accuracy of Word2vec_GRU_CNN is increased by 1.64% and 1.98%, respectively.

5. Conclusions

This paper uses the GRU_CNN hybrid model to classify news texts. The findings of the experiment reveal that the GRU_CNN hybrid model possesses a strong classification performance on the Cnews dataset, with an accuracy of 97.86%, which is better than the single CNN, LSTM, and GRU models and the hybrid model LSTM_CNN. The GRU_CNN hybrid model first preprocesses the dataset and then uses the Word2vec model of the Gensim toolkit to train word vectors. The word's vector representation is fed into the GRU model for semantic information extraction to gain a sea of contextual information about the text, and then the output results are input into the CNN model for semantic feature extraction. Finally, text classification via Softmax layer.

Although compared with the traditional machine learning classification models logistic regression and Naive Bayes, single CNN, LSTM, and GRU models as well as hybrid model LSTM_CNN, this paper proposes that the hybrid model GRU_CNN has a better classification effect on the Cnews dataset. However, there is still a lot of optimization space in this experiment, such as further optimizing the neural network model, optimizing the word vector method, and trying to introduce an attention mechanism into the model.

Data Availability

The data that support the findings of this study can be obtained from the corresponding author upon reasonable request.

Ethical Approval

There are no animal studies conducted by any of the authors in this study.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China, under Grants 61702462 and 61906175, the Henan Province Scientific and Technological Research Project, under Grants 222102210010 and 222102210064, the Research and Practice Project of Higher Education Teaching Reform in Henan Province, under Grants 2019SJGLX320 and 2019SJGLX020, the Undergraduate Universities Smart Teaching Special Research Project of Henan Province, under Gran Jiao Gao [2021] no. 489-29, and the Academic Degrees & Graduate Education Reform Project of Henan Province, under Grant 2021SJGLX115Y.

References

- [1] J. Zhang, Q. Wang, Y. Li, D. Li, and Y. Hao, "A method for Chinese text classification based on three-dimensional vector space model," in *Proceedings of the 2012 International Conference on Computer Science and Service System*, pp. 1324–1327, Nanjing, China, August 2012.
- [2] X. Cheng, C. Zhang, and Q. Li, "Improved Chinese short text classification method based on ERNIE_BiGRU model," *Journal of Physics: Conference Series*, vol. 1993, no. 1, Article ID 012038, 2021.
- [3] L. Zhou and X. Bian, "Improved text sentiment classification method based on BiGRU-Attention," *Journal of Physics: Conference Series*, vol. 1345, no. 3, Article ID 032097, 2019.
- [4] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences. Multimedia systems," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 655–665, Baltimore, MD, USA, June 2014.
- [5] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," in *Proceedings of the Machine Learning: ECML-98*, pp. 137–142, Chemnitz, Germany, April 1998.
- [6] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [7] S. B. Kim, K. S. Han, H. C. Rim, and S. Hyon Myaeng, "Some effective techniques for naive Bayes text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 11, pp. 1457–1466, 2006.
- [8] S. J. Yen, Y. S. Lee, J. C. Ying, and Y. C. Wu, "A logistic regression-based smoothing method for Chinese text categorization," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11581–11590, 2011.
- [9] B. A. Wahiba and B. Ahmed, "New fuzzy decision tree model for text classification," in *Proceedings of the 1st International Conference on Advanced Intelligent System and Informatics*, pp. 309–320, Beni Suef, Egypt, November 2015.
- [10] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: a comprehensive review," 2021, <https://arxiv.org/abs/2004.03705>.

- [11] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, Doha, Qatar, October 2014.
- [12] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, <https://arxiv.org/abs/1510.03820>.
- [13] R. Johnson and Z. Tong, "Deep Pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 562–570, Vancouver, Canada, July 2017.
- [14] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7370–7377, 2019.
- [15] T. Mikolov, I. Sutskever, K. Chen, C. Kai, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3111–3119, Red Hook, NY, USA, December 2013.
- [16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [17] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, <https://arxiv.org/abs/1412.3555>.
- [18] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 2047–2052, Montreal, QC, Canada, July 2005.
- [19] Y. Cao, T. R. Li, Z. Jia, and C. F. Yin, "BGRU: a new method of Emotion analysis based on Chinese text," *Computer Science and Exploration*, vol. 13, no. 6, pp. 973–981, 2019.
- [20] Y. Li and H. B. Dong, "Text sentiment analysis based on feature fusion of convolution neural network and bidirectional long short-term memory network," *Journal of Computer Applications*, vol. 38, no. 11, pp. 3075–3080, 2018.
- [21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the International Conference on Learning Representations*, pp. 1–12, Scottsdale, ARI, USA, May 2013.
- [22] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the 14th Python in Science Conference*, pp. 2267–2273, Austin, TX, USA, January 2015.