Advances in Artificial Intelligence Research (AAIR)
ISSN 2757-7422
www.dergipark.com/aair/

Vol. 3 (No. 2), pp. 96-107, 2023
doi: 10.54569/aair.1329048
Published online: Oct 23, 2023

Analyzing the Impact of Augmentation Techniques on Deep Learning Models for Deceptive Review Detection: A Comparative Study

Anusuya Krishnan [1,*], Kennedyraj Mariafrancis [2]

[1] United Arab Emirates University, UAE
[2] Noorul Islam University, India

Abstract

Deep Learning has brought forth captivating applications, and among them, Natural Language Processing (NLP) stands out. This study delves into the role of the data augmentation training strategy in advancing NLP. Data augmentation involves the creation of synthetic training data through transformations, and it is a well-explored research area across various machine learning domains. Apart from enhancing a model's generalization capabilities, data augmentation addresses a wide range of challenges, such as limited training data, regularization of the learning objective, and privacy protection by limiting data usage. The objective of this study is to investigate how data augmentation improves model accuracy and precise predictions, specifically using deep learning-based models. Furthermore, the study also conducts a comparative analysis between deep learning models without data augmentation and those with data augmentation. Our proposed method, combining RoBERTa with data augmentation, achieves a remarkable 94% accuracy, underscoring the significant effectiveness of this approach in improving NLP model performance.

Keywords: *Deep learning techniques; deceptive review detection; augmentation.*

1. Introduction

Text augmentation techniques play a vital role in Natural Language Processing (NLP) tasks by expanding the diversity and quantity of training data. With the increasing availability of large text corpora and the advancements in deep learning models, researchers and practitioners have recognized the significance of text augmentation in improving the performance of various NLP applications. Text augmentation involves generating new instances of text by applying a series of linguistic transformations while preserving the original meaning and context. These transformations can range from simple operations such as synonym replacement and random word deletion to more complex techniques like paraphrasing and back-translation. By augmenting the training data, models can learn to generalize better, capture a wider range of language patterns, and become more robust to variations in input [1].

In recent years, text augmentation techniques have gained considerable attention and have been successfully applied to a wide range of NLP tasks, including sentiment analysis, text classification, machine translation, and named entity recognition, among others. Researchers have explored various augmentation strategies, leveraging linguistic rules, pre-trained language models, and domain-specific knowledge to generate augmented data that mimics the characteristics of real-world text [2].

The benefits of text augmentation extend beyond the augmentation of model performance. It can also help mitigate data scarcity challenges, particularly in low-resource domains, where collecting a large, annotated dataset is often impractical or expensive. Furthermore, text augmentation can address issues related to data bias, as it can help balance the representation of different classes and reduce the risk of overfitting to specific patterns in the training data. Despite the widespread use of text augmentation in NLP, there exists a compelling need for a comprehensive understanding of its impact on model performance. Diverse factors, including the selection of augmentation techniques, the extent of augmentation, and the intricate interplay between augmentation and model architecture, can collectively shape overall effectiveness. Additionally, a critical examination of the constraints and potential risks linked with text augmentation, such as the introduction of synthetic artifacts or inadvertent amplification of inherent biases in the original data, remains imperative [3-6].

In this study, we aim to provide a comprehensive analysis of text augmentation techniques in the context of NLP tasks. We will explore the existing augmentation methods, categorize them based on their underlying principles, and discuss their advantages and limitations. Furthermore, we will conduct a comparative evaluation of without augmentation and with augmentation techniques on NLP tasks, investigating their impact on model performance, generalization, and robustness.

The following sections of this paper provide a comprehensive examination of the research findings. Section 2 presents a concise overview of relevant works in the field, drawing insights from existing literature. Moving forward, Section 3 delves into the background and intricacies of our proposed machine learning approach

*Corresponding author
E-mail address: anusuyababy18@gmail.com

Received: 18/Jul/2023; Received in revised form: 06/Sep/2023; Accepted: 18/Oct/2023.

designed for detecting deceptive reviews. The section elaborates on the methodology and underlying principles of our approach. Section 4 focuses on presenting the detailed results and analysis conducted to evaluate the accuracy of our model in identifying deceptive reviews. These experiments offer valuable insights into the efficacy of our approach. Finally, in Section 5, the paper concludes by summarizing the key findings and contributions of our work, while also outlining potential avenues for future research and development in this domain.

2. Related works

The development of universal data augmentation techniques in the field of natural language processing (NLP) has encountered challenges due to the complexity of devising generalized rules for transforming languages. Our survey introduces various methodologies for implementing data augmentation in textual data. While some prior research has proposed methods for augmenting data in NLP, there remains a noticeable gap in a comprehensive exploration of this area. Notably, one study generated new data by translating sentences into French and then back into English [7]. Furthermore, alternative approaches encompass introducing noise to the data to enhance smoothness and employing predictive language models to replace synonyms [8-10]. However, despite the validity of these techniques, their practical adoption is limited due to substantial implementation costs in relation to the performance improvements they yield. Another study laid the groundwork for incorporating formal causal language into data augmentation, involving the use of structured causal models and the process of abduction, action, and prediction to generate counterfactual instances. Their experiments encompass aligning phrases within sequences in neural machine translation to extract counterfactual substitutions [10].

One of the prominent challenges when applying machine learning methods, including artificial neural networks, to small datasets is the issue of learning stability. This instability can manifest in a strong dependence on parameter selection, training batch order, and other factors [11]. It also encompasses challenges such as overfitting and the inability to achieve effective generalization. As a result, the volatility inherent in small datasets can lead to inconsistent outcomes when employing models of similar architecture. This, in turn, can limit generalization and accuracy, impeding overall performance [12]. Furthermore, ensuring the reproducibility of results can become problematic, even when employing the same architecture and dataset, making comparisons, enhancements, and optimizations more challenging [13]. Previous studies have made efforts to address the stability challenge in machine learning with small datasets using diverse techniques [14]. These methods include k-fold cross-validation, ensemble methods, Radial-Basis Function (RBF) neural networks, and other approaches [15-17]. While many of these methods have demonstrated success in specific applications, their applicability across various datasets and problems remains uncertain [18-20]. This uncertainty arises from the specific architectural requirements and assumptions concerning the data distribution.

In a previous study, the author introduced an innovative approach to ensemble learning using augmentation for the detection of stance and fake news [22]. Their method involves data augmentation, which entails creating new training instances sharing the same true labels as their source instances [23]. Although data augmentation is widely embraced in computer vision (CV) as a cornerstone of robust predictive performance, its exploration in the realm of natural language processing (NLP) has been comparatively limited. In the context of NLP, it is often seen as an incremental improvement, affording modest but consistent performance enhancements. This distinction can be attributed to the inherent characteristics of textual data, such as polysemy, which renders the formulation of label-preserving transformations notably more intricate [24]. Another author proposed augmentation rules rooted in syntactic heuristics including strategies like inversion, where subject and object are swapped in sentences, and passivization, which transforms hypothesis sentences in premise-hypothesis natural language inference (NLI) pairs into their passive counterparts [25]. Another noteworthy approach by researchers involves constructing a knowledge graph from the extensive input context for abstractive summarization. This graph facilitates semantic swaps that uphold overall coherence [26].

In recent developments, some studies highlighted the limited benefits of supervised syntactic parsing in contemporary pre-training and fine-tuning pipelines with large language models [27]. Some researchers extended these ideas to domains such as molecules, genomics, therapeutics, and healthcare, proposing the integration of such structures with text data to potentially enhance text representations [28]. They have introduced an edge augmentation technique that exposes graph neural networks (GNNs) to likely yet nonexistent edges while limiting exposure to existing but improbable ones [29]. This augmentation results leads to a 5% average accuracy enhancement across six prominent node classification datasets. Building upon this, researchers demonstrate the effectiveness of adversarially controlled node feature augmentation for graph classification [30]. Similarly, another author created an embedding graph to enforce coherence between predictions from strongly and weakly augmented data [31].

Another author delves into the application of data augmentation (DA) for the detection of stance and fake news. In the initial segment of their study, they examine the impact of diverse DA techniques on the efficacy

of standard classification algorithms. Their research capitalizes on the insights gleaned from this analysis to introduce a novel ensemble learning methodology rooted in augmentation. Their approach harnesses text augmentation to enrich the diversity and accuracy of the base learners, thereby elevating the predictive capabilities of the ensemble [32]. Furthermore, they investigated to address the challenge of class imbalance, a prevalent issue in the realm of stance and fake news detection that often leads to biased models. Also, they empirically demonstrate how text augmentation can be instrumental in mitigating moderate and severe class imbalances, elucidating its potential in rectifying this problem. Another study conducted an analysis of small datasets poses several significant challenges, primarily due to the limited sampling of characteristic patterns. As a result, drawing confident conclusions about the unknown distribution becomes elusive, leading to reduced statistical confidence and increased errors. However, small datasets can be crucial in scenarios involving novel or rare conditions where large amounts of data are unavailable or yet to be accumulated [33-34].

On the other hand, unsupervised machine learning methods have demonstrated effective capabilities in reducing dimensionality and eliminating redundancy in the observable parameter space [34]. These methods have played a crucial role in analyzing and identifying characteristic patterns and trends in complex data, including constrained datasets. Importantly, these methods do not rely on labeled data with known outcomes and can be applied with smaller sample sizes [35]. We believe that these characteristics make unsupervised machine learning methods suitable for analyzing early and rare conditions, scenarios, and situations where large amounts of confidently labeled data have not yet been accumulated. Furthermore, these methods enable the aggregation of data for later stages of statistical analysis using conventional techniques.

To overcome the challenges associated with analyzing small datasets, a proposed solution involves the utilization of different augmentation techniques. These techniques aim to identify characteristic structures within the input data, effectively addressing both the issues of limited labels and training instability when working with minimal datasets. By examining the latent representations of the augmented data, it becomes possible to identify underlying structures that can be used to generate new data points.

3. Methodology

In this section, we present a comprehensive exposition of our proposed framework, drawing comparisons with existing methodologies. The primary objective of this research is to delve into the impact of data augmentation on enhancing model accuracy and the precision of predictions, with a specific focus on deep learning-based models. Additionally, this study undertakes a comparative analysis between deep learning models that do not employ data augmentation and those that integrate this technique. The overall architecture of the proposed methodology can be observed in **Figure 1**.

3.1. Data Collection

In our research, we utilized the "Deceptive opinion spam corpus" dataset, which is publicly available on Kaggle, comprises a total of 1600 reviews [1]. These reviews are divided into two categories: 800 truthful reviews and 800 fake reviews. The dataset focuses specifically on the top twenty hotels in Chicago. The reviews within this dataset were sourced from platforms such as TripAdvisor and Amazon Mechanical Turk. Each review entry in the dataset contains the following information: the review label indicating its authenticity (whether it is fake or truthful), the corresponding hotel name, the sentiment or polarity of the review classified as positive or negative, the review source, which can either be TripAdvisor or Mechanical Turk, and lastly, the actual review text itself.

3.2. Data Preprocessing

In machine learning tasks, data preprocessing plays a critical role, especially when dealing with unstructured data. It involves a diverse set of techniques aimed at cleaning and refining the data, which includes removing punctuation, URLs, stop words, converting to lowercase, tokenization, stemming, and lemmatization. These techniques effectively eliminate irrelevant information and prepare the data for feature extraction. Tokenization, a fundamental technique in natural language processing, breaks down the text into smaller units referred to as tokens. These tokens can encompass alphanumeric characters, punctuation marks, or special characters. For example, the sentence "the desert is tasty" would be tokenized into "the," "desert," "is," and "tasty." Stop words, such as articles, conjunctions, prepositions, and pronouns, are commonly used words in everyday English. These words lack significant meaning and are typically removed during the preprocessing stage. Lemmatization is the process of transforming tokenized words into their base or root forms to enhance human comprehension. It reduces words to their common root form, effectively eliminating variations in tense or form. For instance, words like "dancing," "danced," and dancer" would all be reduced to "dance." In this paper, we utilize lemmatization as an integral part of our data preprocessing approach.

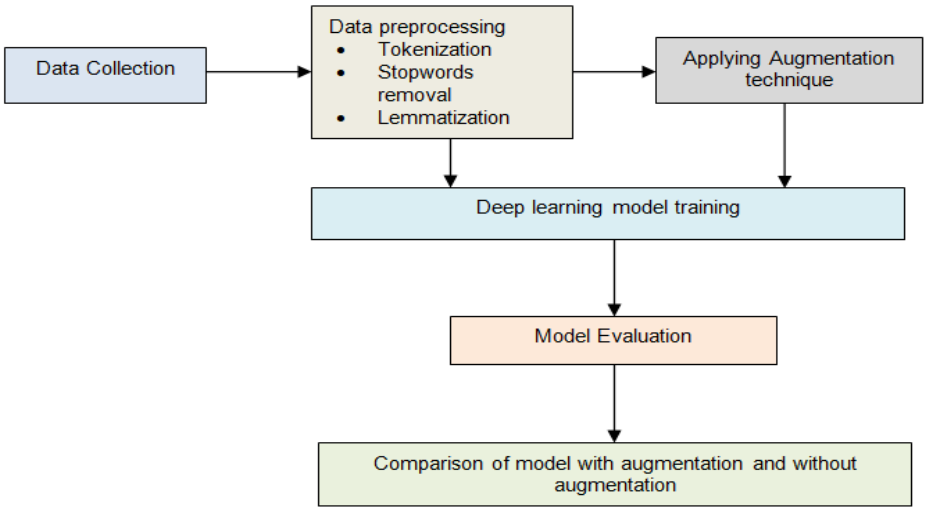

Figure 1. *Proposed Methodology*

3.3. Augmentation Technique

Text augmentation is an effective method utilized to amplify the diversity and volume of a text dataset. It entails employing various transformations to the text, resulting in augmented versions that can improve the performance of machine learning models. Commonly employed techniques for text augmentation include synonym replacement, random insertion, random deletion, and random swap. The synonym replacement technique entails substituting words in the text with their synonyms. This expands the vocabulary and introduces semantic variations, thereby enriching the text. On the other hand, random insertion involves the random insertion of words at different positions within the text. This technique increases the length of the text and introduces noise, which enhances the model's ability to handle variations in input length. Random deletion, as the name suggests, involves the removal of random words from the text. This simulates missing or incomplete information, compelling the model to learn more resilient representations. Lastly, random swap entails swapping the positions of two randomly selected words within the text. This introduces different word orderings, thereby enhancing the model's capacity to accommodate variations in sentence structure. The example of four methods in data augmentation is provided in the **Table 1**.

Table 1. *Example of data augmentation of four methods*

Example	Sentences
Original data	I love to eat pizza.
Synonym replacement	I adore indulging in pizza.
Random insertion	I absolutely love to eat delicious pizza every weekend.
Random swap	I pizza to eat love.
Random deletion	I to eat pizza.

Following data cleaning, we partitioned the dataset into an 80:20 ratio using the train-test split method. Subsequently, we exclusively applied all four augmentation techniques to the training data.

3.4. Deep learning model training

Then we employed different deep learning techniques. Deep learning model training for text data using word embedding is a powerful technique that enables the effective representation and analysis of textual information. By employing word embeddings, which are dense vector representations of words, the model can capture semantic relationships and contextual information. The process of training a deep learning model for text data using word embedding involves several steps. First, the text data is preprocessed by removing unnecessary characters, tokenizing the text into individual words or subword units, and performing other necessary preprocessing steps. Next, we split the dataset into 80:20 using train test split. Following that, augmentation technique applied only on training dataset. Then word embedding technique is chosen, such as Word2Vec or GloVe. These techniques create word embeddings by considering the co-occurrence statistics of words in a large corpus. Alternatively, pretrained word embeddings can be utilized, which have been trained

on extensive datasets and capture general language semantics. Once the word embeddings are obtained, they are used to represent the text data. Each word in the input text is mapped to its corresponding word embedding vector. The resulting sequence of word embeddings forms the input to the deep learning model. In this study, we utilized different deep learning models like recurrent neural networks (RNNs) such as LSTM and transformer-based models like BERT, DistilBERT, XLNET, RoBERTa [36-40]. These models take the word embeddings as input and employ layers to capture the hierarchical structure and dependencies within the text data.

During training, the model parameters are optimized by backpropagation and gradient descent methods. The model is presented with the input word embeddings, and the predicted outputs are compared to the true labels or targets. The difference between the predictions and the targets is measured using a suitable loss function, such as categorical cross-entropy for multi-class classification or binary cross-entropy for binary classification. The model parameters are updated iteratively based on the gradients of the loss function with respect to the model's parameters. This process continues for multiple epochs, where each epoch represents a complete pass through the training data. The model learns to adjust its parameters to minimize the loss and improve its performance on the given task. Hyperparameter tuning is an essential step to optimize the model's performance. Parameters such as learning rate, batch size, number of layers, and regularization techniques can be adjusted to improve the model's generalization ability and prevent overfitting. In this paper, we configured the hyperparameters for transformer models as follows: learning rate = 5e-5, weight decay = 0, number of epochs = 20, and batch size = 64. Additionally, for LSTM models, we utilized dropout rate = 0.2, number of layers = 64, activation = sigmoid, optimizer = Adam, and epochs = 20.

3.5. Model Evaluation

The evaluation of a deep learning model is a vital stage in determining its performance and effectiveness. It revolves around assessing the model’s ability to generalize to new and unseen data, as well as its accuracy in accomplishing the designated task. In this research, we have selected accuracy, F1-score, and the loss function as our performance metrics. Accuracy is a widely used metric for assessing classification models, representing the percentage of correctly predicted labels. Additionally, the loss, exemplified by cross-entropy loss, quantifies the degree of prediction error in the model. In this study, we used binary cross-entropy loss function for binary classification problems, where the goal is to assign one of two classes to each input sample. By monitoring the loss throughout the training process, we gain insights into the model's convergence and progress.

3.6. Hardware and software used:

We utilized a high-performance Linux Ubuntu 18.04 server with 40 CPU cores, powered by an Nvidia DGX-1, and equipped with 8 NVIDIA Tesla V100 GPUs, each boasting 32 GB of memory. This server also featured a web-based multi-user concurrent job scheduling system. All experiments and training were carried out using Python 3.8.16. The libraries used for Deep Learning, Data Processing, and Data Visualization, including tensorflow-gpu v2.3.1, keras v2.4.3, SciPy v1.16.4, NumPy, Pandas, Matplotlib, and Seaborn, were integrated into the environment.

4. Experimental results

In this section, we delve into the experimental assessment of our proposed approach and provide an analysis of the acquired results. To conduct this evaluation, our primary focus during the comparative performance analysis was the utilization of the Opinion spam dataset, comprising a modest 1,600 reviews. Despite its relatively small size, this dataset presented a formidable challenge when implementing deep learning models.

4.1. Original Data without Augmentation

The original data undergoes a preprocessing phase to ensure cleanliness and standardization. This involves various techniques such as tokenization, lowercasing, punctuation removal, handling special characters, and employing methods like stopword removal and lemmatization. Once the data is preprocessed, we split the dataset into 80:20 using train test split. Then it is transformed into word embeddings, which are compact vector representations capturing the semantic and contextual information of words. Each word in the preprocessed data is associated with its corresponding word embedding vector, resulting in sequences or matrices of word embeddings. These word embeddings act as input features for deep learning models. The models leverage their layers and parameters to make predictions and conduct tasks such as text classification. By utilizing the learned representations from the word embeddings, the deep learning models effectively interpret and analyze the textual data, enabling accurate predictions and effective text classification. We have used accuracy and loss metrics for model evaluation. **Table 2** shows the deep learning model performance without augmentation.

Table 2. *Deep learning models without Augmentation.*

Deep learning model	Library	Training		Testing	
		Accuracy	Loss	Accuracy	Loss
LSTM	Word2Vec	92%	0.05	86%	0.49
LSTM	GloVe	93%	0.08	82%	0.82
BERT	Transformer	91%	0.09	78%	0.77
DistilBERT	Transformer	96%	0.03	85%	0.76
XLNET	Transformer	95%	0.06	88%	0.43
RoBERTa	Transformer	97%	0.01	91%	0.31

The table provides an overview of various deep learning models trained on a specific task, including their architecture, word embedding library, training accuracy, training loss, testing accuracy, and testing loss. The LSTM models with Word2Vec and GloVe libraries achieved accuracies ranging from 92% to 93% during training and 86% to 82% during testing. The BERT model with the Transformer library achieved a training accuracy of 91% but dropped to 78% on the testing dataset. The DistilBERT model achieved a high training accuracy of 96% and 85% accuracy on the testing dataset. The XLNET model achieved accuracies of 95% during training and 88% during testing. Finally, the RoBERTa model outperformed all others with a training accuracy of 97% and a testing accuracy of 91%. **Figure 2** shows the performance analysis of training accuracy and testing accuracy of the deep learning models. From this **Figure**, we observed that RoBERTa model performed well compared to other deep learning models.

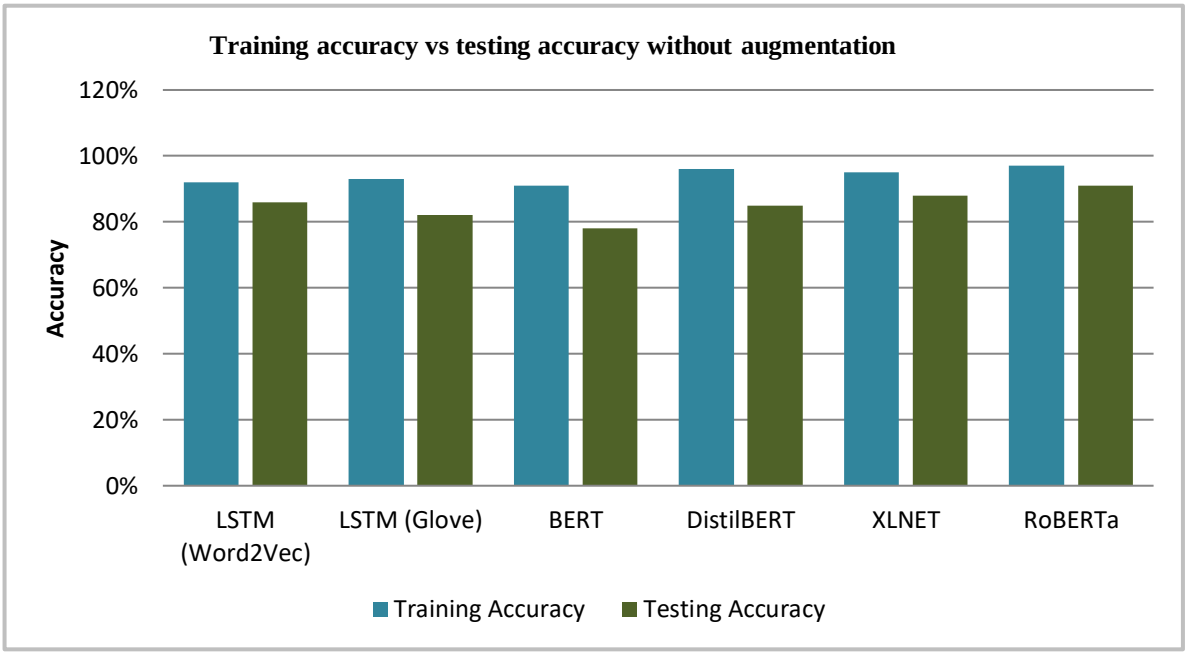

Figure 2. *Training vs testing accuracy without augmentation technique*

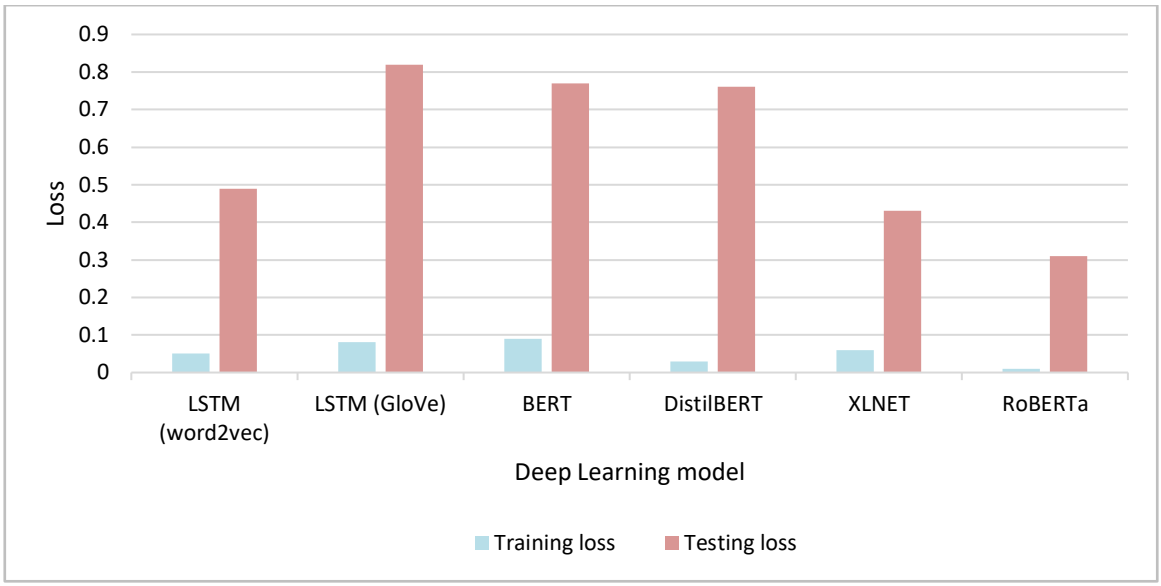

Figure 3. *Training vs testing loss without augmentation technique*

Figure 3 shows the results of a deep learning model with a training loss and testing loss. The training loss is a measure of how well the model fits the training data, while the testing loss is a measure of how well the model fits new data. The lower the loss, the better the model is performing. The chart explains that the RoBERTa model has the lowest training loss and testing loss, followed by DistilBERT, XLNET, and LSTM (Word2Vec). This suggests that RoBERTa is the best-performing model out of the ones tested. The LSTM models (Glove) and BERT have higher training and testing losses than the other models. This suggests that LSTM and BERT models are not as good at fitting the data as the other models. Overall, the table shows that the RoBERTa model is the best-performing model out of the ones tested. However, it is important to note that the results of this table are based on a small dataset, so more testing is needed to confirm these results.

4.2. Applying augmentation in training data

Due to the limited size of the dataset, training deep learning models directly may not yield satisfactory results. To address this, we employed various techniques to augment the data and improve its size and diversity. The data was cleaned and split into an 80:20 ratio for training and testing. We applied the synonym replacement technique to augment the training data, expanding each sentence by a factor of ten, resulting in an augmented dataset of 14,400 samples from the original 1,600. For augmentation, we utilized the nlpaug library's wordnet parameter in the nlpaug.augmenter.word module. However, the test data remained unaltered, with augmentation exclusively applied to the training data. The augmented data was preprocessed by removing punctuation and stop words.

Table 3. *Deep leaning models with Augmentation.*

Deep learning model	Library	Training		Testing	
		Accuracy	Loss	Accuracy	Loss
LSTM	Word2Vec	95%	0.01	88%	0.39
LSTM	GloVe	96%	0.005	89%	0.32
BERT	Transformer	98%	0.009	91%	0.27
DistilBERT	Transformer	97%	0.001	90%	0.26
XLNET	Transformer	98%	0.02	89%	0.29
RoBERTa	Transformer	99%	0.001	94%	0.15

Subsequently, we transformed the augmented data into word vectors using either the Word2Vec technique from the gensim library or the word embeddings for transformer method. Tokenization was applied to further process the vector data, preparing it for input into the deep learning models. The **Table 3** summarizes the performance of different deep learning models on a specific task. We have evaluated our model using accuracy and loss metrics. Each model is associated with a specific word embedding library, and the table provides information on their training and testing accuracy as well as training and testing loss. The LSTM model, when

combined with the Word2Vec library, achieved a training accuracy of 95% with a training loss of 0.01. When evaluated on a separate testing dataset, it attained an accuracy of 88% with a testing loss of 0.39. Similarly, when the LSTM model was paired with the GloVe library, it achieved a slightly higher training accuracy of 96% and a lower training loss of 0.005. On the testing dataset, it achieved an accuracy of 89% with a testing loss of 0.32. **Table 3** shows the deep learning model performance with augmentation.

The BERT model achieved a higher training accuracy of 98% with a training loss of 0.009. During testing, it reached an accuracy of 91% with a testing loss of 0.27. The DistilBERT model achieved a slightly lower training accuracy of 97% with a training loss of 0.001. On the testing dataset, it obtained an accuracy of 90% with a testing loss of 0.26. The XLNET model achieved a training accuracy of 98% with a higher training loss of 0.02. When tested, it achieved an accuracy of 89% with a testing loss of 0.29. Finally, the RoBERTa model, using the Transformer library, achieved the highest training accuracy of 99% with a training loss of 0.001. On the testing dataset, it achieved an impressive accuracy of 94% with the lowest testing loss of 0.15. Overall, we observed from the table that the RoBERTa model achieved the highest accuracy, followed closely by the BERT model.

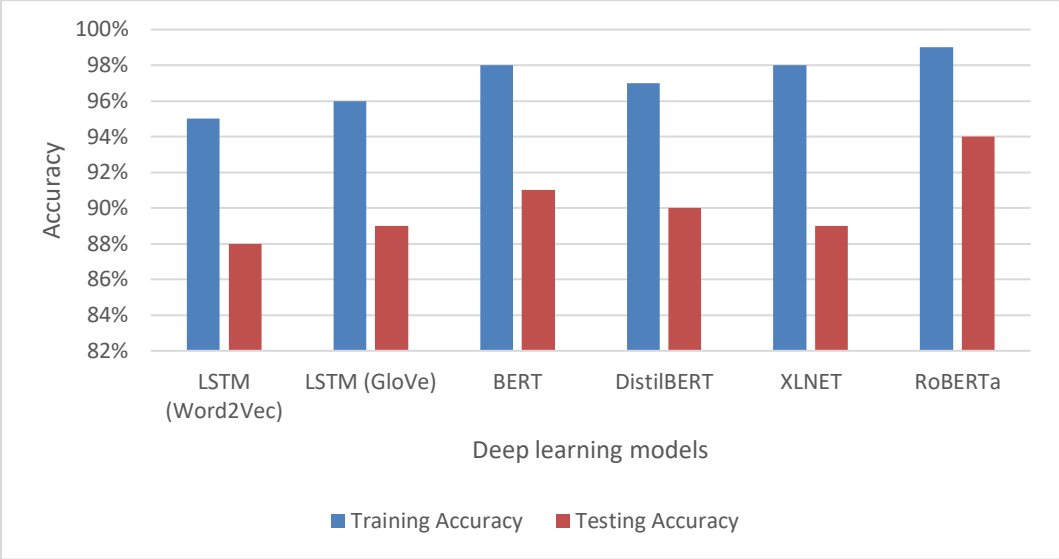

Figure 4. *Training vs testing accuracy with augmentation technique*

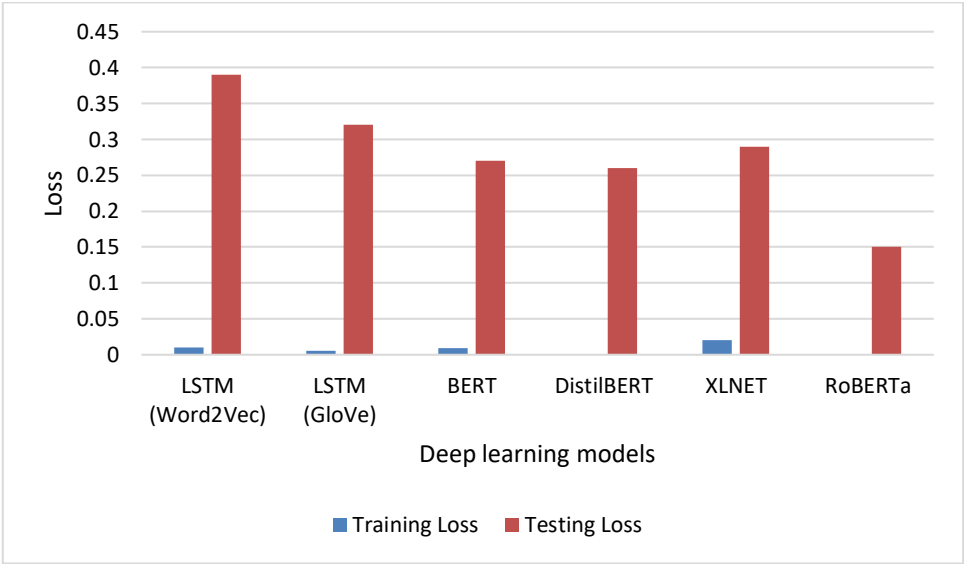

Figure 5. *Training vs testing loss with augmentation technique*

Figure 4 showcases the training and testing accuracy of deep learning models. Also, **Figure 5** shows that the RoBERTa model has the lowest training loss and testing loss, followed by DistilBERT, LSTM (GloVe), and BERT. This suggests that BERT is the best-performing model out of the ones tested. The LSTM model (Word2Vec) and XLNET have higher training and testing losses than the other models. This suggests that

LSTM model (Word2Vec) and XLNET are not as good at fitting the data as the other models. Overall, the barchart shows that the RoBERTa model is the best-performing model out of the ones tested.

4.3. Performance comparison of proposed model with machine learning models and other existing works

In **Table 4**, we present a comparative analysis of our model with existing machine learning approaches. Our model has demonstrated superior performance when compared to these methods. Table 3 provides an overview of different models along with their associated libraries and accuracy scores. The passive aggressive classifier, utilizing the TF-IDF library, achieved an accuracy of 92.5%. This model employs an online learning algorithm that makes updates based on the observed data. The linear support vector machine (LSVM), using the bag of words approach, obtained an accuracy of 91.8%. SVM is a supervised learning algorithm that separates data points using hyperplanes in a high-dimensional feature space. RoBERTa, utilizing the transformer library, achieved an accuracy of 91% without augmentation and 94% with augmentation. RoBERTa utilizes self-attention mechanisms to capture contextual dependencies in the input text. These results demonstrate the performance of the models on the given task, using RoBERTa with augmentation showing the highest accuracy, followed by the passive aggressive classifier and the linear SVM.

Table 4. *Performance comparison of proposed methodology with machine learning approach*

Input data	Model	Library	Accuracy
Original data	Passive aggressive classifier	TF-IDF	92.5%
Original data	Linear support vector machine	Bag of words	91.8%
Original data	RoBERTa	Transformer	91%
Original data (augmentation only on training data)	RoBERTa	Transformer	94%

It's essential to consider specific factors when assessing the performance of our proposed methods in comparison to other existing studies. Firstly, our results can only be directly compared to studies that have employed the same dataset, namely OpSpam. Secondly, we adopted well-established performance metrics for binary classification problems, consistent with those found in the scikit-learn library. Lastly, for the sake of ensuring a fair basis for comparison, we have exclusively presented the best results from each of the studies used in our evaluation. **Table 5** provides a comprehensive comparison between our proposed methods and existing research work that utilized the OpSpam dataset for fake review detection. Most of these studies have employed machine learning classifiers as their modeling approach. Regarding feature extraction, methods such as TF-IDF, LIWC, unigrams, and bigrams have been widely used in existing works. Notably, our proposed methods, leveraging transformer model with augmentation, have achieved the highest levels of performance in these comparisons.

Table 5. *Performance comparison of proposed methodology with existing works on the opinion spam dataset*

Model	Augmentation Techniques	Feature vectorization	Accuracy	F1-score	References
Linear SVM	No	TF-IDF	84%	83.6%	[9]
Spam GAN	Yes	Bag of words	86.8%	87.8%	[29]
Naïve Bayes	No	Bigram	93%	93%	[5]
SVM	No	Unigram + Bigram	90.9%	91%	[2]
SVM	No	LIWC + Bigram	91.2%	91%	[28]
Linear SVM	No	LDA + WSM	86%	86%	[24]
RoBERTa	Yes	Word embeddings	94%	94%	Our work

5. Discussion

The paper introduces a novel method for detecting fake reviews, particularly tailored for small datasets, through the application of augmentation techniques. The proposed method outperforms state-of-the-art approaches on the Deceptive Opinion Spam Corpus (OpSpam) dataset. It utilizes four distinct text augmentation techniques, including synonym replacement, random deletion, random insertion, and random swap. These techniques are applied to the training data, which is subsequently converted into vector representations for input into deep learning models. The proposed method offers several notable strengths. First, it excels in identifying subtle fake review activities with a higher degree of accuracy than existing state-

of-the-art methods. Second, the incorporation of augmentation techniques has notably enhanced the precision of fake review detection. Third, the research underscores the significance of introducing artificial data into the training set, a factor that greatly improves the overall performance of the method.

Detecting fake reviews continues to pose a formidable challenge, particularly when dealing with smaller datasets. Artificial intelligence (AI) has exhibited promising outcomes within this domain, but the deficiency in interpretability and transparency of machine learning models, especially in the context of smaller datasets, raises doubts about the credibility of these proposed models. In alignment with the objective of developing models that not only excel in performance but also prioritize interpretability and transparency in their decision-making processes, the proposed model attains superior results and effectively tackles the constraint of working with smaller datasets by integrating augmentation techniques.

Overall, the proposed model presents a promising solution to the challenge of detecting fake reviews in smaller datasets, with the potential to make a significant impact on the realms of marketing and e-commerce. By enhancing the accuracy and dependability of online reviews using limited data, the approach addresses a critical need. Despite the promising results of this study, several limitations warrant further investigation in future research. One limitation pertains to the reliance on a single dataset, which may not adequately represent all categories of fake reviews. Additionally, the method primarily focuses on textual features, leaving room for exploration of other feature types such as images or user behavior.

Another noteworthy limitation lies in the interpretability of the proposed approach, which could constrain its applicability in contexts where transparency and interpretability are paramount. Additionally, the proposed method may still be susceptible to sophisticated fake review attacks, prompting further investigation into methods for bolstering its resilience against such threats. Despite these limitations, our proposed model presents a valuable contribution to the field of fake review detection, particularly in the context of small datasets. It outperforms state-of-the-art approaches and offers several notable strengths, including its ability to identify subtle fake review activities and its resilience to overfitting.

Future studies should delve into graph mining and machine learning techniques to further enhance the performance and generalizability of the method, as well as to address the limitations of our study. For example, future studies could investigate the use of multiple datasets from different domains to improve the robustness of the model to a wider range of fake review attacks.

6. Conclusion

Text augmentation techniques are pivotal for enhancing the performance and generalization capabilities of natural language processing models, particularly in challenging scenarios like detecting fake reviews with limited labeled data. Most existing fake review detection methods rely on supervised machine learning models due to the scarcity of data. In this study, we harnessed diverse data augmentation methods and incorporated them into various deep learning models to address data scarcity and counteract overfitting. Text augmentation not only mitigated these challenges but also bolstered the models' ability to handle out-of-domain or previously unseen examples. Our research introduces an innovative approach that harnesses augmentation techniques tailored for smaller datasets, using the OpSpam dataset for evaluation. The methodology involved text data preprocessing, augmentation application, transformation into vector representations, and training deep learning classifiers with transformer models. Our results demonstrate that augmentation significantly enhances the accuracy of detecting subtle fake review patterns, outperforming existing state-of-the-art methods.

In conclusion, our proposed approach highlights the efficacy of augmentation-based strategies and emphasizes the importance of employing these techniques, especially when dealing with limited datasets in the context of fake review detection. Our future research efforts will focus on enhancing our approach by integrating text-based features and investigating the utilization of advanced deep learning graph models like GCN, graphSAGE, or GNN, with the goal of further improving performance metrics. Additionally, we intend to explore additional graph theory techniques that hold promise for application in our dataset.

Declaration of interest

The authors declare that there is no conflict of interest.

Acknowledgements

The authors would like to acknowledge the support provided by United Arab Emirates University.

References

[1] Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Security and Privacy 1.1 (2018) : e9.

[2] Bengio Y. “Learning deep architectures for AI”, Foundations and trends® in Machine Learning 2.1 (2009): 1-127.

[3] Algur SP, Patil AP, Hiremath PS, Shivashankar S. “Conceptual level similarity measure-based review spam detection”, International Conference on Signal and Image Processing, pp. 416-423. IEEE, 2010.

[4] Lau RY, Liao SY, Kwok RC, Xu K, Xia Y, Li Y. “Text mining and probabilistic language modeling for online review spam detection”, ACM Transactions on Management Information Systems (TMIS) 2, no. 4: 1-30, 2012.

[5] Jindal Nitin, Bing Liu. “Opinion spam and analysis”, In Proceedings of the international conference on web search and data mining, pp. 219-230, 2008.

[6] Choi Wonil, Kyungmin Nam, Minwoo Park, Seoyi Yang, Sangyoon Hwang, Hayoung Oh. “Fake review identification and utility evaluation model using machine learning”, Frontiers in artificial intelligence 5: 1064371, 2023.

[7] Yu AW, Dohan D, Luong MT, Zhao R, Chen K, Norouzi M, Le QV. “Qanet: Combining local convolution with global self-attention for reading comprehension”, 2018. CoRR aba/1804.09541. URL: https://arxiv.org/pdf/1804.09541.

[8] Kobayashi. “Contextual augmentation: Data augmentation by words with paradigmatic relations”, In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, pp. 452–457, 2018.

[9] Xie Z, Wang SI, Li J, Lévy D, Nie A, Jurafsky D, Ng AY. “Data noising as smoothing in neural network language models”, In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017.

[10] LeBaron B, Weigend AS. “A bootstrap evaluation of the effect of data splitting on financial time series”, IEEE Transactions on Neural Networks 9.1 (1998): 213-220.

[11] Coates A, Ng A, Lee H. “An analysis of single-layer networks in unsupervised feature learning”, Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011.

[12] Cunningham P, Carney J, Jacob S. “Stability problems with artificial neural networks and the ensemble solution”, Artificial Intelligence in medicine 20.3 (2000): 217-225.

[13] Dolgikh S. “Identifying explosive epidemiological cases with unsupervised machine learning”, medRxiv (2020): 2020-05.

[14] Hornik K, Stinchcombe M, White H. “Multilayer feedforward networks are universal approximators”, Neural networks 2.5 (1989): 359-366.

[15] Izonin I, Tkachenko R, Dronyuk I, Tkachenko P, Gregus M, and Rashkevych M. “Predictive modeling based on small data in clinical medicine: RBF-based additive input-doubling method”, Mathematical Biosciences and Engineering 18.3 (2021): 2599-2613.

[16] Karar ME. “Robust RBF neural network–based backstepping controller for implantable cardiac pacemakers”, International Journal of Adaptive Control and Signal Processing 32.7 (2018): 1040-1051.

[17] Ott M, Choi Y, Cardie C, Hancock JT. “Finding deceptive opinion spam by any stretch of the imagination”, arXiv preprint arXiv:1107.4557 (2011).

[18] Prystavka P, Cholyshkina O, Dolgikh S, Karpenko D. “Automated object recognition system based on convolutional autoencoder”, In 2020 10th international conference on advanced computer information technologies (ACIT). IEEE, 2020.

[19] Corona Rodriguez R, Alaniz S, Akata Z. “Modeling conceptual understanding in image reference games”, Advances in Neural Information Processing Systems 32 (2019).

[20] Li J, Ott M, Cardie C, Hovy E. “Towards a general rule for identifying deceptive opinion spam”, In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 1566-1576, 2014.

[21] Salah I, Jouini K, Korbaa O. “Augmentation-based ensemble learning for stance and fake news detection”, In Advances in Computational Collective Intelligence – 14th International Conference, ICCCI 2022, Proceedings of Communications in Computer and Information Science (Vol. 1653, pp. 29–41). 2022.

[22] Xie Q, Dai Z, Hovy E, Luong T, Le Q. “Unsupervised data augmentation for consistency training”, Advances in neural information processing systems 33, pp. 6256-6268, 2020.

[23] Shorten C, Khoshgoftaar TM, Furht B. “Text data augmentation for deep learning”, Journal of Big Data, 8(1), 1–34, 2021.

[24] Min J, McCoy RT, Das D, Pitler E, Linzen T. “Syntactic data augmentation increases robustness to inference heuristics”, In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2339–2352, 2020.

[25] Huang L, Wu L, Wang L. “Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward”, In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5094–510, 2020.

[26] Glavaš G, Vulić I. “Is supervised syntactic parsing beneficial for language understanding tasks? An empirical investigation”, In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 3090–3104, 2021.

[27] Li MM, Huang K, Zitnik M. “Representation learning for networks in biology and medicine: advancements, challenges, and opportunities”, arXiv preprint arXiv:2104.04883 (2021).

[28] Zhao T, Liu Y, Neves L, Woodford O, Jiang M, Shah N. Data augmentation for graph neural networks. In Proceedings of the AAAI conference on artificial intelligence 2021 May 18 (Vol. 35, No. 12, pp. 11015-11023).

[29] Kong K, Li G, Ding M, Wu Z, Zhu C, Ghanem B, Taylor G, Goldstein T. “FLAG: adversarial data augmentation for graph neural networks”, arXiv:2010.09891 (2020).

[30] Devlin J, Chang MW, Lee K, Toutanova K. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, In Proceedings of NAACL-HLT 2019 Jun 2 (Vol. 1, p. 2).

[31] Ester M, Kriegel HP, Sander J, Xu X. “A density-based algorithm for discovering clusters in large spatial databases with noise”, In KDD, vol. 96, no. 34, pp. 226-231. 1996.

[32] Forman, George, Ira Cohen. “Learning from little: Comparison of classifiers given little training”, In European Conference on Principles of Data Mining and Knowledge Discovery. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[33] Fischer A, Igel C. “Training restricted Boltzmann machines: An introduction”, Pattern Recognition 47.1 (2014): 25-39.

[34] Hekler EB, Klasnja P, Chevance G, Golaszewski NM, Lewis D, Sim I. “Why we need a small data paradigm”, BMC medicine 17.1 (2019): 1-9.

[35] Mukherjee A, Liu B, Glance N. “Spotting fake reviewer groups in consumer reviews”, In Proceedings of the 21st international conference on World Wide Web, pp. 191-200, 2012.

[36] Shojaee S, Murad MA, Azman AB, Sharef NM, Nadali S. “Detecting deceptive reviews using lexical and syntactic features”, In 2013 13th International Conference on Intellient Systems Design and Applications, pp. 53-58. IEEE, 2013.

[37] Sanh V, Debut L, Chaumond J, Wolf T. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”, arXiv preprint arXiv:1910.01108 (2019).

[38] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”, arXiv preprint arXiv:1907.11692 (2019).

[39] Clark K, Luong MT, Le QV, Manning CD. “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators”, arXiv preprint arXiv:2003.10555 (2020).

[40] Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J. “LSTM: A search space odyssey”, IEEE transactions on neural networks and learning systems 28, no. 10: 2222-2232, 2016.