electronics

Article

Machine Learning Algorithms for Fostering Innovative Education for University Students

Yinghua Wang [1,2], Fucheng You [3,*] and Qing Li [4]

[1] Faculty of Education, Northeast Normal University, Changchun 130024, China; wangyh159@nenu.edu.cn
[2] School of Economics and Management, Jilin Institute of Chemical Technology, Jilin 132022, China
[3] College of Information Engineering, Beijing Institute of Graphic Communication, Beijing 102600, China
[4] College of Information Science and Technology, Jinan University, Guangzhou 510630, China
* Correspondence: youfucheng@bigc.edu.cn

Abstract: Data augmentation with mixup has been proven effective in various machine learning tasks. However, previous methods primarily concentrate on generating previously unseen virtual examples using randomly selected mixed samples, which may overlook the importance of similar spatial distributions. In this work, we extend mixup and propose MbMix, a novel yet simple training approach designed for implementing mixup with memory batch augmentation. MbMix specifically selects the samples to be mixed via memory batch to guarantee that the generated samples have the same spatial distribution as the dataset samples. Conducting extensive experiments, we empirically validate that our method outperforms several mixup methods across a broad spectrum of text classification benchmarks, including sentiment classification, question type classification, and textual entailment. Of note, our proposed method achieves a 5.61% improvement compared to existing approaches on the TREC-fine benchmark. Our approach is versatile, with applications in sentiment analysis, question answering, and fake news detection, offering entrepreneurial teams and students avenues to innovate. It enables simulation and modeling for student ventures, fostering an entrepreneurial campus culture and mindset.

Keywords: machine learning; data augmentation; text classification; innovative education

Citation: Wang, Y.; You, F.; Li, Q. Machine Learning Algorithms for Fostering Innovative Education for University Students. *Electronics* **2024**, *13*, 1506. https://doi.org/10.3390/electronics13081506

Academic Editor: George Angelos Papadopoulos

Received: 18 March 2024
Revised: 6 April 2024
Accepted: 10 April 2024
Published: 16 April 2024

1. Introduction

The rapid advancements in natural language processing (NLP) technologies have opened up new opportunities for innovation in educational strategies and the implementation of practical aspects of innovative education at universities [1,2], such as text classification [3–5], summary generation [6,7], and question answering [8]. This success can mainly be attributed to the deep architectures of language models that typically have over ten million learnable parameters. Such a massive number of parameters enables BERT model [9] to solve complex problems. Nevertheless, language models require extensive training data to avoid overfitting and improve model generalization. However, obtaining large samples of annotated data is costly and time-consuming. This challenge offers educators and students a chance to craft new solutions within their innovative education programs, streamlining data generation and annotation to cut costs and save time. These innovations could transform the development of natural language processing and broaden opportunities in diverse educational fields [10,11].

To obtain more training data, a series of data augmentation [12,13] algorithms have emerged. Data augmentation aims to create synthetic samples that serve as supplementary training data to regularize the language model. Most NLP data augmentation methods, including synonym replacement [14] and back-translation [15], aim to transform the training sample by substituting words with their synonyms, as demonstrated by [16].

Recently, a highly practical data augmentation approach called mixup [17] was proposed and has demonstrated outstanding performance in boosting the accuracy of image classification tasks. Unlike in image tasks [18], applying the mixup algorithm in NLP tasks is more complicated due to the discrete nature of text data. To this end, there have been several attempts to deploy mixup on embeddings [19], intermediate representations [4] or input-level data [20]. However, these methods randomly select samples for augmentation, which may result in an unstable distribution of training samples. An unstable data distribution could lead to the model underperforming in practical applications, adversely affecting the outcomes of innovation education. Although randomization is effective, mixup models based on the memory batch technique may have improved generalization performance. This is because the memory batch can leverage the historical samples stored in a cache to enhance the diversity of the training data, resulting in a more stable training data distribution [21].

To overcome the constraints and simultaneously aid in university student innovation education, in this work, we present a novel mixup method considering the memory batch [21,22], called MbMix. At a high level, we first construct the memory batch sampler to store and sample historical samples. Then, we synergize the mixup model and memory batch such that mixup is performed between historical and current samples, thereby generating new training samples. This approach yields a more stable distribution of the training data while reducing the noise introduced by generating new samples through standard mixup. MbMix has the potential to create an infinite amount of new augmented samples, which is because MbMix can be used as a generic component to upgrade existing mixup-derived methods.

To support university students' innovation and entrepreneurship, we conducted simulation experiments with diverse datasets. We utilized the RTE [23] and MRPC [24] datasets to identify trending news topics and provide entrepreneurial recommendations. With the new media industry's rapid growth, we used the SST-2 [25] and IMDB [26] datasets to gauge viewer interests, offering insights for media-related ventures. To create a safer entrepreneurial environment and shield students from harmful content, we employed the TREC [27] spam detection dataset. Lastly, we analyzed the emotional and psychological states of student entrepreneurs using the QNLI [28] dataset for sentiment analysis. MbMix has been empirically proven to be effective through extensive experiments on several text classification benchmarks. The experimental results indicate that MbMix exhibits several desirable properties. We summarize the major contributions of this paper as follows:

- From a fresh standpoint, we divide the mixup model's training into bi-level subtasks: memory batch sampling and mixed sample generation. We integrate these subtasks into a framework called MbMix to enhance the model via data augmentation.
- We innovatively utilize the memory batch block to generate mixed samples. The memory batch method can increase the stable distribution of training data by incorporating historical samples while enhancing the effectiveness of the mixup model.
- MbMix significantly surpasses its counterparts in various classification scenarios based on eight text classification datasets and achieves a 5.61% improvement compared to existing approaches on the TREC-fine benchmark. The superior performance of MbMix has the potential to bolster the effectiveness of innovation education for university students.

The rest of the paper is organized as follows. Section 2 presents the related work. In Section 3, we introduce the MbMix method. The experimental details are presented in Section 4. Analysis of the experimental results is presented in Section 5. Finally, a brief conclusion is drawn in Section 6.

2. Related Work

2.1. Data Augmentation

Data augmentation is a widely utilized technique in the domain of deep learning [29–32]. By introducing a diversified set of training samples, it aids in enabling models to learn more generalized feature representations. Consequently, this enhances the model's ability to generalize when confronted with previously unseen data. Furthermore, when training data are limited, models are prone to overfitting. Data augmentation, by expanding the scale and diversity of the training set, can effectively mitigate the issue of overfitting [33]. Dropout [34] is a widely applied regularization technique in the field of deep learning, capable of enhancing performance through a form of data augmentation. It aims to prevent overfitting by randomly "dropping" a portion of the neurons during the training process. Specifically, in each training iteration, every neuron in the network has a certain probability of not being activated, thereby not participating in forward and backward propagation. This method ensures that the model does not rely on any specific small subset of input data, compelling the network to learn more robust feature representations. Adversarial training [35] enhances model stability and generalization by introducing slight perturbations into the training data, thereby training the model to recognize and resist these disturbances. The core concept of adversarial training lies in optimizing the model's performance not only on the original data during the training process but also on slightly modified versions of the data. This approach aims to improve the model's robustness against minor changes or attacks, thereby ensuring more reliable and secure performance in real-world applications [36–38]. Label smoothing [39], by adjusting the distributions of training data labels, mitigates the issue of a model having excessive confidence in its labels. Its advantage is that it encourages models to learn smoother decision boundaries, thereby reducing the risk of overfitting and enhancing the models' generalization ability towards unseen data. In practical applications, label smoothing is typically implemented by mixing the true labels with uniformly distributed labels in a certain proportion. This approach not only retains a degree of true label information but also increases label diversity, making the model training process more robust. L2 regularization [40] enhances a model's generalization capability by incorporating a regularization term into the model's loss function, thereby constraining the training loss and preventing model overfitting. This technique works by penalizing the square magnitude of the model parameters, which effectively reduces the complexity of the model. In doing so, L2 regularization ensures that the model does not overly adapt to the training data, hence improving its performance on unseen data. A gradient penalty [41] constrains the magnitude of gradients by adding a regularization term to the loss function, preventing the occurrence of exploding or vanishing gradients during the training process. This promotes the stable training of models and a better generalization performance. Back-translation [15], a data augmentation technique, involves translating the original text from one language to another and subsequently translating it back to the original language. This process generates texts that may vary slightly in grammatical structure and linguistic expression while preserving semantic consistency. Through this approach, the diversity and richness of the training data can be enhanced without altering the fundamental meaning. This technique enables the model to better comprehend and learn the intricacies of and variability in language, ultimately improving its ability to generalize to unseen data. Back-translation proves particularly effective when the available training data are limited or when the model is required to capture subtle textual nuances. It bolsters the model's robustness and adaptability in handling a wide array of linguistic patterns and idiosyncrasies. Synonymous substitution [14] increases data diversity by replacing words in the text with their synonyms, effectively expanding the training dataset and thus helping the model learn more generalized linguistic features. Mixup [17], originally presented as a data augmentation method based on mixing in computer vision, has potential applications in enhancing the robustness [42–45] and security of deep learning models against attacks [46–51]. In natural language processing [52–56], Guo et al. [19] presented two strategies for applying the mixup model to sentence classification: word and sentence embeddings. TMix [4] mixes two samples in hidden spaces. SSMix [20] synthesizes a sentence by span-based mixing. However, these methods generate examples by random combination, disregarding the importance of spatial distribution. As shown in Table 1, our mixup method aims to prevent these issues by clustering similar samples.

Table 1. Comparison of different data augmentation methods.

Method	Sampling	Mixed Model
Mixup	Random	Sample
TMix	Random	Feature
SSMix	Random	Sample
Our	Memory Batch	Sample

2.2. Innovative Education

Li et al. assessed the dynamic evolution mechanism of the digital entrepreneurship ecosystem using sentiment analysis models, providing insights that could be integrated into the curricula of innovation and entrepreneurship education [57]. Jazib et al. developed an ensemble classifier that enables real-time analysis of sentiments on various topics through data visualization techniques, which could be incorporated into educational tools to enhance entrepreneurs' analytical skills [58]. Malik et al. emphasized the growing significance of entrepreneurship education, as it resonates with the worldwide focus on value creation and employability. They introduced a machine learning approach to forecast students' adaptability in online entrepreneurship programs, employing algorithms like Random Forest, C5.0, CART, and artificial neural networks. The research showed high accuracy rates, confirming the potential of machine learning in predicting students' performance and adaptability, a critical component for customizing support in innovative education settings [59,60]. This methodology provides educators with a robust tool to pinpoint students who may require additional resources, allowing for tailored educational interventions [61]. Chen et al. discussed the impact of "Internet Plus" technology on innovation and entrepreneurship education, underscoring its transformative effect on teaching methodologies, faculty development, and curriculum design. They advocated for an educational system that merges internet concepts with machine learning to satisfy the evolving needs of higher education in fostering innovative and entrepreneurial skills. In their research, they not only crafted a platform architecture with diverse functions but also validated its precision and effectiveness through empirical studies. They also proposed strategic directions for the advancement of the Internet Plus education framework to bolster students' capabilities in innovation and entrepreneurship [62]. He et al. investigated the present conditions and hurdles faced by innovation and entrepreneurship education in China, pinpointing deficiencies in student initiatives and the assessment systems. They introduced a cutting-edge evaluation framework using backpropagation neural networks, marking a significant step forward in the enhancement of educational quality. Their thorough comparative analyses of assessment techniques showcased the superiority of their model, which can be instrumental in refining evaluation processes within innovative education environments [63]. The MbMix text classification method can support university students engaged in innovation and entrepreneurship by enabling their businesses to comprehend and analyze vast amounts of text data with precision. This capability can yield critical insights, assisting innovators and entrepreneurs in swiftly categorizing and interpreting market trends, competitor insights, and customer feedback, which are vital for informed decision making, product development, and strategic market positioning.

3. MbMix Method

Text classification tasks hold vast potential in the field of innovation education. For instance, student entrepreneurs can leverage text classification to develop spam filter systems, which help users eliminate a significant amount of junk mail, thus enhancing work efficiency and user experience. Additionally, entrepreneurs can create sentiment analysis tools using text classification technologies, aiding businesses in comprehending consumers' emotions towards products, services, or brands to fine-tune marketing strategies and improve products. Therefore, this section will begin by introducing the language model, the memory batch sampler, and then develop a new series of mixup methods with an extremely efficient architecture and high performance.

3.1. Language Model

First, in order to build a text classification model, we use the BERT [9] pre-trained language model as the basic architecture. BERT is a pre-trained deep learning model that has achieved state-of-the-art results in various natural language processing tasks [64–66], including text classification. The core component of a BERT-based classification model is the BERT encoder. BERT is a transformer-based model that consists of multiple layers of self-attention and feed-forward neural networks. The self-attention mechanism allows the BERT model to capture the contextual relationships between words in a sentence, enabling it to understand the meaning and context of the input text. To construct a BERT-based classification model, the pre-trained BERT encoder is typically used as the backbone of the model. The input text is tokenized and fed into the BERT encoder, which generates contextualized embeddings for each token in the input sequence. These embeddings capture the semantic and syntactic information of the words in the context of the entire sequence. On top of the BERT encoder, a classification head is added. The classification head usually consists of one or more dense layers, followed by a softmax activation function. The embeddings generated by the BERT encoder are passed through the classification head, which learns to map the embeddings to the corresponding class labels.

During the fine-tuning process, the pre-trained BERT encoder and the classification head are trained together on a labeled dataset specific to the classification task. The model is optimized using a loss function, such as cross-entropy loss, which measures the discrepancy between the predicted class probabilities and the true class labels. The model is trained using techniques like backpropagation and gradient descent to update the weights of the BERT encoder and the classification head:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{1}$$

where N is the number of samples in the dataset; C is the number of classes; y_{ij} is the true label of the sample; and $\hat{y}_{ij}$ is the predicted probability of the sample. Classification applications based on pre-trained language models have shown excellent performance. Compared to traditional machine learning models, they are more capable of producing new beneficial effects for innovation education among university students.

3.2. Memory Batch

The memory batch model is a sampling algorithm that leverages historical samples in a cache pool to enhance current training data. Moreover, the memory batch model can also be utilized to address the issue of imbalanced data. Suppose there are fewer samples in specific categories in the training set. In that case, these samples can be added to the cache pool, and their sampling rate can be increased to enhance the model's learning performance in those categories [21,67]. As shown in Figure 1, we mainly use the memory batch algorithm to stabilize the distribution of new samples generated by the mixup model.

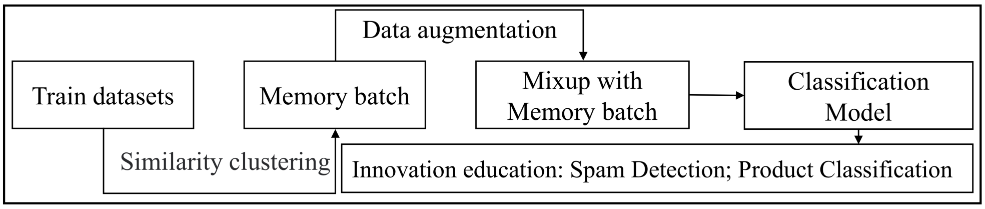


Figure 1. Diagram of the classification model structure. Classification models enhanced by data augmentation algorithms can be applied to innovation education.

The memory batch model achieves this aim by randomly selecting a subset of samples and storing them in memory. Subsequently, small batches of data can be randomly generated from this memory cache during training. Since the samples in the memory cache are selected randomly, the overall distribution of the dataset is preserved. This helps alleviate the impact of instability in the distribution of new samples generated by the mixup model.

Assuming dataset D contains N samples, the construction process of the memory batch model is as follows.

Let t denote the tth iteration in training. n samples $x_0^t, x_1^t, \ldots, x_{n-1}^t$ are randomly selected and stored in the memory cache.

For each mini-batch dataset b_k^t, where k is the batch size and $i_0, i_2, \ldots, i_{k-1}$ are distinct integers representing the selection of k random samples from the memory cache randomly selected from $[0, n-1]$, we can express b_k^t as follows:

$$b_k^t = x_{i_0}^t, x_{i_2}^t, \ldots, x_{i_{k-1}}^t, \tag{2}$$

where $x_{i_j}^t$ is the jth sample in the tth iteration of the mini-batch dataset and n represents the memory cache size. Repeat the above steps until all data samples have been trained.

In practical applications, the size of the memory cache n and the size of the mini-batch dataset k can be adjusted as needed to meet different training requirements. We formalize the output of the memory batch as follows:

$$M_{batch} = \{(x_i; y_i)_b \in D\}, \tag{3}$$

where M_{batch} consists of historical samples as well as current samples, which are combined as the latest batch used by the mixup model to generate new samples.

3.3. Mixup with Memory Batch

In this work, we optimize the mixup model [17] using memory batch for random selection and memory sampling. The fundamental concept behind the mixup algorithm is simple: if we have two labeled samples $(x^i; y^i)$ and $(x^j; y^j)$, where symbol x is a text and symbol y is the label's one-hot representation, the algorithm performs linear interpolation to generate virtual training samples:

$$\begin{aligned} \tilde{x}^{ij} &= mix(x^i, x^j) = \lambda x^i + (1-\lambda)x^j \\ \tilde{y}^{ij} &= mix(y^i, y^j) = \lambda y^i + (1-\lambda)y^j \end{aligned} \tag{4}$$

where $\lambda \in [0; 1]$. Mass-generated virtual samples can be used to train the classification model. Briefly, the mixup algorithm is a data augmentation method that generates new samples based on the training sample. Nevertheless, in previous methods, the mixed samples are randomly selected, which is effective but may result in instability in the distribution of training samples and impact model learning.

To this end, we innovatively synergize the mixup model and memory batch approach to make the distribution of the generated samples more stable. As demonstrated in Algorithm 1, we first construct the memory batch sampler, and the new sample batch is sampled. Then, the memory batch samples can be mixed by various mixup algorithms to generate the new virtual training data, and the new sample can be fed to the neural network:

$$\begin{aligned} \tilde{x}_{new}^{ij} &= \lambda x_{M_{batch}}^i + (1-\lambda)x_{M_{batch}}^j \\ \tilde{y}_{new}^{ij} &= \lambda y_{M_{batch}}^i + (1-\lambda)y_{M_{batch}}^j \end{aligned} \tag{5}$$

where $x_{M_{batch}}$ and $y_{M_{batch}}$ denote the samples in the memory batch. The memory batch can be constructed by data preprocessing, which has desirable plug-and-play properties. The MbMix data augmentation algorithm can further enhance the accuracy of model classification. Importantly, the plug-and-play properties of MbMix enable it to be effectively deployed in innovative applications for university students, increasing the potential success rate of their entrepreneurial endeavors.

Algorithm 1 Mixup model with memory batch

```
Input: train samples $x, y$; train datasets $D$;
Parameter: $n$; $k$;
Output: new train samples $x_{new}$, $y_{new}$.
while MbMix do
    $M_{batch} = Sampler(D, n, k)$
    $\tilde{x}^{ij}_{new} = \lambda x^{i}_{M_{batch}} + (1 - \lambda) x^{j}_{M_{batch}}$
    $\tilde{y}^{ij}_{new} = \lambda y^{i}_{M_{batch}} + (1 - \lambda) y^{j}_{M_{batch}}$
end while
return $\tilde{x}_{new}$; $\tilde{y}_{new}$.
```

3.4. Validity Analysis

One of the primary advantages of MbMix is its ability to significantly increase the size of the dataset. The mixup algorithm generates new training samples by performing linear interpolation between pairs of existing samples and their corresponding labels. This process creates synthetic examples that expand the training set, providing the model with a more diverse range of data points to learn from. On the other hand, memory batch utilizes a cache pool to store and reuse historical samples during training. By incorporating these previously seen samples alongside the current batch, memory batch enhances the diversity and temporal stability of the training data. The combination of the mixup model and memory batch in MbMix amplifies the dataset expansion effect, resulting in a substantially larger and more varied training set.

However, it is important to note that the new samples generated by the mixup model are not exact replicas of the original data. While the mixup model creates plausible interpolations between existing samples, these synthetic examples may introduce some level of noise or slight deviations from the true data distribution. If left unchecked, this noise can potentially affect the model's learning performance and lead to suboptimal generalization. This is where memory batch plays a crucial role in mitigating the impact of mixup-induced noise. By incorporating historical samples from the cache pool, memory batch helps to stabilize the training process and reduce the influence of noisy synthetic examples. The inclusion of real, previously seen samples acts as a regularizing force, guiding the model towards more reliable and consistent representations.

Furthermore, the combination of the mixup and memory batch models in MbMix offers additional benefits beyond dataset expansion and noise reduction. The increased diversity of training samples resulting from MbMix helps to improve the model's ability to generalize to unseen data. By exposing the model to a wider range of variations and interpolations, MbMix's robustness is enhanced and the risk of overfitting is reduced. The model learns to capture the underlying patterns and relationships in the data more effectively, leading to improved performance on both the training set and unseen test data.

In conclusion, the combination of the mixup and memory batch models in MbMix is a powerful approach for enhancing machine learning models. By expanding the dataset size, improving sample diversity, reducing noise, and promoting generalization, MbMix offers significant benefits over using either technique alone. This synergistic combination has the potential to boost model performance, particularly in scenarios where limited training data are available or when dealing with complex and diverse datasets. Compared to traditional data augmentation algorithms, MbMix can be better applied to university student innovation education, helping students to understand and analyze text content more effectively and to unearth new ideas and insights.

4. Experimental Setup

Dataset: Aiming to assist university students in their innovation and entrepreneurship, we utilize the news classification datasets RTE [23] and MRPC [24] to explore social hotspots in real-time news, providing recommendations for student entrepreneurial projects. In today's rapidly evolving new media landscape, entrepreneurship among students focused on new media also holds significant potential. Therefore, we use the movie review datasets SST-2 [25] and IMDB [26] to explore viewers' interest trends from movie reviews, offering guidance for student entrepreneurship in new media. During the entrepreneurial journey, students may encounter harmful messages through emails or text messages, which could lead to business failure or more severe consequences. To address this, we experiment with the spam detection dataset TREC [27] to filter out malicious emails, aiming to provide a clean entrepreneurial environment for students. In addition, to investigate the emotional and psychological states of students during their entrepreneurial endeavors, we conduct sentiment analysis as a simulated experiment using the QNLI [28]. Finally, we use OLID [68] and COLA [69] to see how Mbmix can help to improve model robustness.

Baseline model: We compare MbMix with four baselines: (i) a vanilla classification model without mixup, (ii) TMix [19], (iii) EmbedMix [4], and (iv) SSMix [20]. The vanilla model is a language-model-based classification model that does not employ any data augmentation methods. EmbedMix applies the mixup model on the embedding layer. TMix synthesizes a sentence via span-based mixing at a specific encoder layer. SSMix employs the saliency score to retain the most distinctive tokens from a mixed text.

Training Details: We test our methods and deployment experiments on the prevailing pre-trained language model: BERT. We perform all experiments using PyTorch on an NVIDIA 3090 GPU with 24 G of memory with three different seeds, and report the average score. We train our models using the AdamW optimizer. We set the learning rate with a warmup to 5^{-5} for the normal model and to 1^{-5} for the mixup model. The maximum sentence length is set to 128. In EmbedMix and TMix, λ is set to 0.2, while in SSMix, λ is set to 0.1. The hyperparameters are the same as those proposed by [20]. The details of the datasets are mentioned in Table 2. The number of epochs for normal model training is 3, while the number of epochs for training the TMix, EmbedMix, and SSMix models is 5, which follows [20].

Table 2. Dataset name, total number of labels, details of train/dev/test splits of datasets, and the memory size we used as the benchmark.

Dataset	Label	Train/Dev/Test	Memory Size
SST-2	2	6.9 K/0.8 K/1.8 K	400
QNLI	3	105 K/5.4 K/5.4 K	2500
COLA	2	8 K/1 K/1 K	300
RTE	2	2.5 K/0.2 K/3 K	400
MRPC	2	3.7 K/0.4 K/1.7 K	300
TREC-coarse	6	5.5 K/0.5 K	300
TREC-fine	47	5.5 K/0.5 K	300
OLID	2	11 K/1.3 K/0.8 K	1000
IMDB	2	25 K/25 K	300

5. Experimental Results and Analysis

Main results: the results in Table 3 show that the mixup model with memory batch provides suitable performance for text classification, showing improved accuracy on all eight test datasets. In the COLA dataset, for example, the average improvement was more significant than 2.8%. We observe that the model with memory batch outperforms the vanilla model, which indicates that the model's performance can be effectively improved by memory batch sampling. It is worth noting that the highest accuracy of the TREC-fine dataset is 93.6%, which is a competitive result among the TREC benchmarks. This implies that deploying the MbMix algorithm in applications such as spam detection can yield favorable results, which is beneficial for advancing innovation education among university students.

Table 3. Experimental results of comparison with baselines. The pre-trained model is BERT-base-uncased. The evaluation metric employed is accuracy. *Mem.bat.* is an abbreviation for memory batch. ↑ indicates an improvement in performance.

Model	**GLUE**					**TREC**		**OLID**	**IMDB**
	SST-2	**QNLI**	**COLA**	**RTE**	**MRPC**	**Coarse**	**Fine**		
No mixup	91.31	90.33	56.68	64.49	84.06	97.26	83.86	79.26	88.53
Mem.bat.	**91.54**	**90.60**	**60.24**	**66.54**	**84.71**	**97.33**	**91.60**	**80.23**	**88.73**
TMix	91.35	90.41	56.80	66.05	84.30	97.33	88.39	79.69	88.67
MbMix	**92.08**	**90.68**	**60.35**	**68.82**	**86.26**	**97.60**	**93.13**	**80.60**	**88.78**
EmbedMix	91.24	90.50	55.80	66.54	83.90	97.53	88.60	79.69	88.60
MbMix	**92.35**	**90.74**	**59.07**	**68.11**	**85.20**	**97.59**	**93.33**	**80.48**	**88.79**
SSMix	91.20	90.51	53.16	65.58	83.65	97.53	88.33	79.59	88.86
MbMix	**92.08**	**90.91**	**54.07**	**66.96**	**84.96**	97.53	**93.60**	**80.53**	**88.94**
Average	**0.72**↑	**0.29**↑	**2.81**↑	**1.93**↑	**1.30**↑	**0.095**↑	**5.61**↑	**0.89**↑	**0.13**↑

Further observation shows that the benefits of the memory batch and mixup models can be synergized. They produce the best reported results on the GLUE benchmark and TREC datasets. Our approach is very simple, as it requires manipulating only the data preprocessing method: there are no changes to the standard neural network or classification algorithms, no language-specific training or tuning, and no external auxiliary data. Compared to existing algorithms, the MbMix algorithm offers greater flexibility, endowing it with extensive potential for application in the realm of university student innovation education. By utilizing the MbMix algorithm, students can delve into text data from various domains, such as scientific literature and market research reports, to identify issues, propose innovative ideas, and verify and refine these through practical experiments.

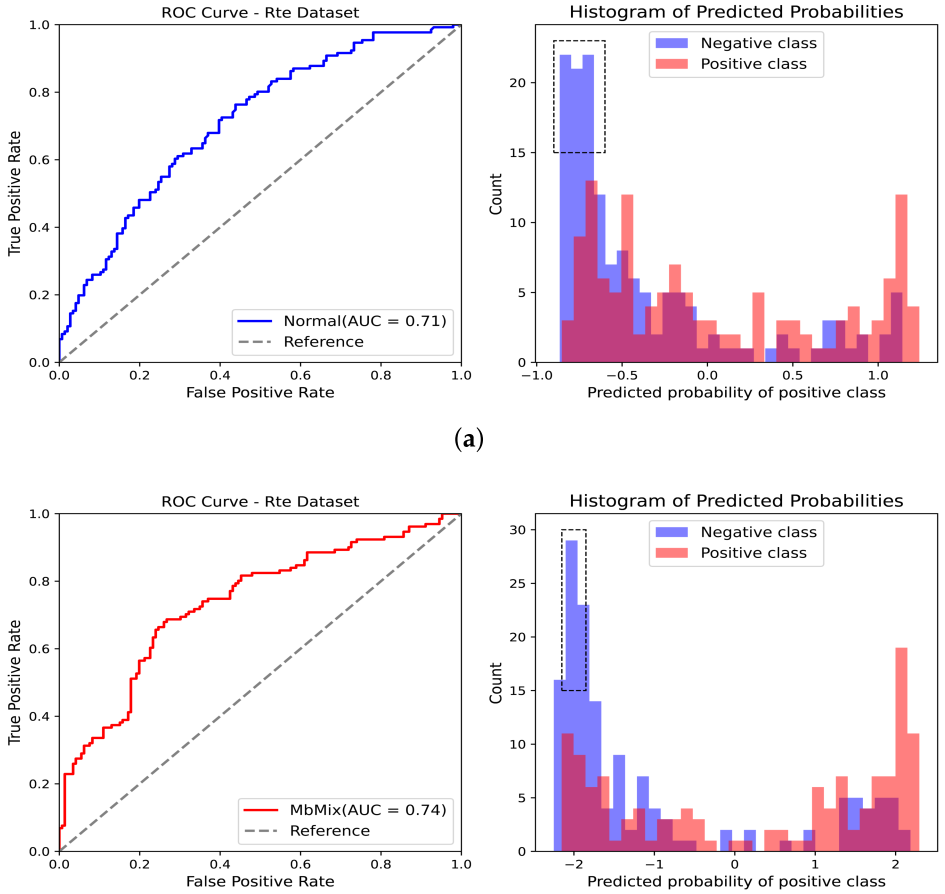
Ablation study: We compare simplified versions of our model to understand the contributions of its individual components, including the mixup model and memory batch. The results from Table 3 demonstrate that the performance improves as we add mixup and memory batch models. MbMix achieves the best results when all components are synergized. These empirical results support our findings in Section 3.3, where we theorize that the combination of the mixup model with the memory batch model can be used as a better data augmentation method with a more stable distribution of training samples.

ROC analyses: Figure 2 displays the ROC curves and probability histograms for the MbMix and normal models. It is observed that the ROC curve for the MbMix model bends more towards the top-left corner than the normal model, indicating better classification performance of the MbMix model. Additionally, the MbMix model achieves an AUC value of 0.74, which is significantly better than the normal model. Lastly, as expected, the probability histogram of the MbMix model has a shorter tail, indicating more accurate prediction results.

Table 4 displays the standard deviation results. It is not difficult to observe that the MbMix model demonstrates a more stable standard deviation compared to the baseline models, indicating that MbMix exhibits greater robustness. Furthermore, Figure 3 presents additional ROC curves, which illustrate that MbMix achieves more competitive performance across different datasets and mixup-derived algorithms. In addition, Figure 4 shows the impact of different memory sizes on model performance.

Table 4. The standard deviation results correspond with the average of our experiments. We report validation accuracies for GLUE and OLID datasets and test accuracies for TREC and IMDM datasets.

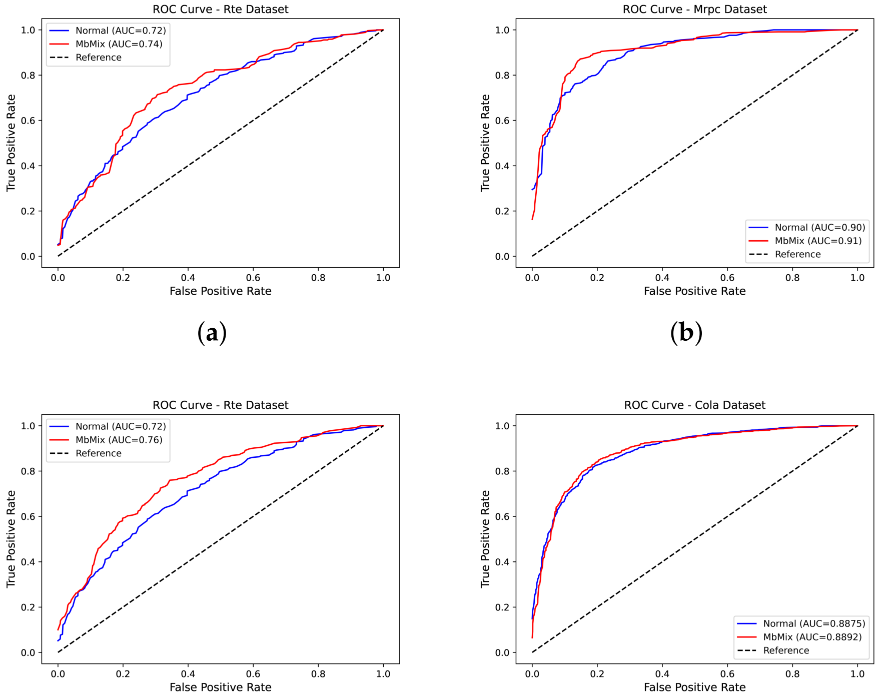
Model	**GLUE**					**TREC**		**OLID**	**IMDB**
	SST-2	**QNLI**	**COLA**	**RTE**	**MRPC**	**Coarse**	**Fine**		
No mixup	91.31 ± 0.23	90.33 ± 0.49	56.68 ± 0.97	64.49 ± 0.68	84.06 ± 0.72	97.26 ± 0.18	83.86 ± 1.03	79.26 ± 0.06	88.53 ± 0.15
Mem.bat.	91.54 ± 0.35	90.60 ± 0.46	60.24 ± 1.53	66.54 ± 2.00	84.71 ± 0.50	97.33 ± 0.09	91.60 ± 0.71	80.23 ± 0.49	88.73 ± 0.13
TMix	91.35 ± 0.45	90.41 ± 0.53	56.80 ± 1.29	66.05 ± 1.28	84.30 ± 1.11	97.33 ± 0.09	88.39 ± 0.58	79.69 ± 0.15	88.67 ± 0.20
MbMix	92.08 ± 0.16	90.68 ± 0.48	60.35 ± 1.85	68.82 ± 3.68	86.26 ± 0.52	97.60 ± 0.16	93.13 ± 0.65	80.60 ± 0.39	88.78 ± 0.18
EmbedMix	91.24 ± 0.46	90.50 ± 0.51	55.80 ± 1.34	66.54 ± 1.45	83.90 ± 1.10	97.53 ± 0.09	88.60 ± 0.56	79.69 ± 0.91	88.60 ± 0.07
MbMix	92.35 ± 0.19	90.74 ± 0.42	59.07 ± 0.71	68.11 ± 2.80	85.20 ± 0.75	97.59±0	93.33 ± 0.41	80.48 ± 0.12	88.79 ± 0.16
SSMix	91.20 ± 0.19	90.51 ± 0.38	53.16 ± 2.24	65.58 ± 0.84	83.65 ± 1.20	97.53 ± 0.18	88.33 ± 0.33	79.59 ± 0.17	88.86 ± 0.05
MbMix	92.08 ± 0.24	90.91 ± 0.63	54.07 ± 1.23	66.96 ± 2.87	84.96 ± 1.02	97.53 ± 0.09	93.60 ± 0.58	80.53 ± 0.39	88.94 ± 0.09



(**a**)

(**b**)

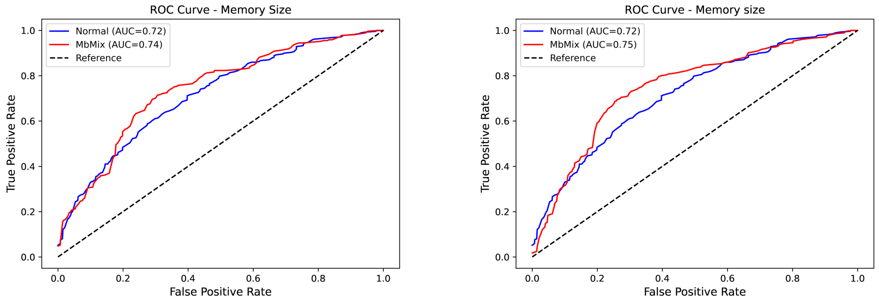
Figure 2. Comparison of ROC curves and histograms of predicted probabilities on the RTE dataset. (**a**) Normal model. (**b**) MbMix model.



(**a**) (**b**)

(**c**) (**d**)

Figure 3. Comparison of ROC curves between the normal and mixup models with memory batch. (**a**) normal and ssmix with memory batch; (**b**) normal and ssmix with memory batch; (**c**) normal and tmix with memory batch; (**d**) normal and ssmix with memory batch.



(**a**) (**b**)

Figure 4. Comparing ROC curves across different memory batch sizes for the MbMix model. (**a**) The memory batch size used is 400. (**b**) The memory batch size used is 600.

6. Conclusions

We have introduced MbMix, a universality data augmentation strategy for text that enhances the mixup model's abilities, resulting in improved text classification performance. MbMix synergizes the mixup and memory batch models to guarantee that the generated samples have a more stable distribution. We conducted various text classification experiments, and the experimental results show that our proposed model achieves outstanding performance to the baseline models. For instance, there is an average improvement of 2.81% on the CoLA dataset. Our successful results suggest that our idea of combining the mixup model with memory batch can be potentially applied to other NLP tasks, such as real-world scenarios which need more labeled data. The MbMix approach presents a promising opportunity for innovative entrepreneurial teams and entrepreneurial university students to innovate in the field of natural language processing. By effectively augmenting training data and improving the performance of NLP applications, MbMix can help entrepreneurial university students in their entrepreneurship simulation training and entrepreneurship practice to reduce costs, accelerate development, and create innovative solutions that address real-world challenges and create value for users in various domains.

In future work, we will investigate the following two points: (i) More mixup-derived methods with memory batch should be studied to improve the performance of the text classification model. (ii) More application scenarios of the mixup model with memory batch, such as image and speech, should be verified to further prove its generalization performance.

Author Contributions: Conceptualization, Y.W.; Methodology, Y.W. and F.Y.; Software, Y.W.; Validation, F.Y.; Formal analysis, Y.W.; Investigation, Q.L.; Resources, F.Y.; Data curation, Y.W.; Writing—original draft, Y.W.; Writing—review & editing, Y.W. and F.Y.; Visualization, F.Y.; Supervision, F.Y.; Project administration, F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (Nos.12271215, 12326378 and 11871248).

Data Availability Statement: Data available in a publicly accessible repository. The data presented in this study are openly available in References [23–28,68,69].

Conflicts of Interest: All authors have read and agreed to the published version of the manuscript. The author declares no conflicts of interest.

References

1. Zhu, Q.; Zhang, H. Teaching strategies and psychological effects of entrepreneurship education for college students majoring in social security law based on deep learning and artificial intelligence. *Front. Psychol.* **2022**, *13*, 779669. [CrossRef] [PubMed]
2. Van Aken, P.; Jung, M.M.; Liebregts, W.; Onal Ertugrul, I. Deciphering Entrepreneurial Pitches: A Multimodal Deep Learning Approach to Predict Probability of Investment. In Proceedings of the 25th International Conference on Multimodal Interaction, Paris, France, 9–13 October 2023; pp. 144–152.
3. Li, Q.; Zhao, S.; He, T.; Wen, J. A simple and efficient filter feature selection method via document-term matrix unitization. *Pattern Recognit. Lett.* **2024**, *181*, 23–29. [CrossRef]
4. Chen, J.; Yang, Z.; Yang, D. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In Proceedings of the ACL, Online, 5–10 July 2020.
5. Li, Q.; Zhao, S.; Zhao, S.; Wen, J. Logistic Regression Matching Pursuit algorithm for text classification. *Knowl. Based Syst.* **2023**, *277*, 110761. [CrossRef]
6. Zhao, S.; Li, Q.; Yang, Y.; Wen, J.; Luo, W. From Softmax to Nucleusmax: A Novel Sparse Language model for Chinese Radiology Report Summarization. *ACM Trans. Asian Low Resour. Lang. Inf. Process.* **2023**, *22*, 180. [CrossRef]
7. Narayan, S.; Zhao, Y.; Maynez, J.; Simões, G.; Nikolaev, V.; McDonald, R. Planning with learned entity prompts for abstractive summarization. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 1475–1492. [CrossRef]
8. Zhao, S.; Liang, Z.; Wen, J.; Chen, J. Sparsing and smoothing for the seq2seq models. *IEEE Trans. Artif. Intell.* **2022**, *4*, 464–472. [CrossRef]
9. Kenton, J.D.M.W.C.; Toutanova, L.K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the AACL, Florence, Italy, 28 July–2 August 2019.
10. Markman, G.D.; Balkin, D.B.; Baron, R.A. Inventors and new venture formation: The effects of general self–efficacy and regretful thinking. *Entrep. Theory Pract.* **2002**, *27*, 149–165. [CrossRef]
11. Mitchelmore, S.; Rowley, J. Entrepreneurial competencies: A literature review and development agenda. *Int. J. Entrep. Behav. Res.* **2010**, *16*, 92–111. [CrossRef]
12. Sun, L.; Xia, C.; Yin, W.; Liang, T.; Philip, S.Y.; He, L. Mixup-Transformer: Dynamic Data Augmentation for NLP Tasks. In Proceedings of the COLING, Barcelona, Spain, 8–13 December 2020.
13. Zhang, L.; Yang, Z.; Yang, D. TreeMix: Compositional Constituency-based Data Augmentation for Natural Language Understanding. *arXiv* **2022**, arXiv:2205.06153.
14. Wei, J.; Zou, K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Proceedings of the EMNLP, Hong Kong, China, 3 November 2019.
15. Xu, J.; Ruan, Y.; Bi, W.; Huang, G.; Shi, S.; Chen, L.; Liu, L. On Synthetic Data for Back Translation. In Proceedings of the NAACL, Seattle, WA, USA, 10–15 July 2022.
16. Kobayashi, S. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In Proceedings of the AACL, Melbourne, Australia, 15–20 July 2018.
17. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. In Proceedings of the ICLR, Vancouver, BC, Canada, 30 April–3 May 2018.

18. Kim, J.H.; Choo, W.; Song, H.O. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In Proceedings of the ICML, Online, 13–18 July 2020.
19. Guo, H.; Mao, Y.; Zhang, R. Augmenting data with mixup for sentence classification: An empirical study. *arXiv* **2019**, arXiv:1905.08941.
20. Yoon, S.; Kim, G.; Park, K. SSMix: Saliency-Based Span Mixup for Text Classification. In Proceedings of the ACL Findings, Online, 1–6 August 2021.
21. Zhong, Z.; Lei, T.; Chen, D. Training Language Models with Memory Augmentation. *arXiv* **2022**, arXiv:2205.12674.
22. Ji, H.; Zhang, R.; Yang, Z.; Hu, Z.; Huang, M. LaMemo: Language Modeling with Look-Ahead Memory. In Proceedings of the AACL, Tokyo, Japan, 25 March–27 April 2022.
23. Bentivogli, L.; Dagan, I.K.; Hoa, D.; Giampiccolo, D. The Fifth PASCAL Recognizing Textual Entailment Challenge. *TAC* **2009**, *7*, 1.
24. Dolan, W.B.; Brockett, C. Automatically Constructing a Corpus of Sentential Paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005), Jeju Island, Republic of Korea, 14 October 2005.
25. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the EMNLP, Grand Hyatt, SA, USA, 18–21 October 2013.
26. Maas, A.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
27. Li, X.; Roth, D. Learning question classifiers. In Proceedings of the COLING, Taipei, Taiwan, 24 August–1 September 2002.
28. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the EMNLP, Austin, TX, USA, 1–5 November 2016.
29. Jia, M.; Shen, X.; Shen, L.; Pang, J.; Liao, L.; Song, Y.; Chen, M.; He, X. Query prior matters: A mrc framework for multimodal named entity recognition. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 3549–3558.
30. Zhang, M.; Shi, H.; Zhang, Y.; Yu, Y.; Zhou, M. Deep learning-based damage detection of mining conveyor belt. *Measurement* **2021**, *175*, 109130. [CrossRef]
31. Jia, M.; Shen, L.; Shen, X.; Liao, L.; Chen, M.; He, X.; Chen, Z.; Li, J. MNER-QG: An end-to-end MRC framework for multimodal named entity recognition with query grounding. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 8032–8040.
32. Zhang, M.; Jiang, K.; Zhao, S.; Hao, N.; Zhang, Y. Deep-learning-based multistate monitoring method of belt conveyor turning section. *Struct. Health Monit.* **2023**. [CrossRef]
33. Zhao, S.; You, F.; Chang, W.; Zhang, T.; Hu, M. Augment BERT with average pooling layer for Chinese summary generation. *J. Intell. Fuzzy Syst.* **2022**, *42*, 1859–1868. [CrossRef]
34. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
35. Shafahi, A.; Najibi, M.; Ghiasi, M.A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L.S.; Taylor, G.; Goldstein, T. Adversarial training for free! In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
36. He, Z.; Yang, Y.; Zhao, S. Towards Pre-trained Language Model for Dynamic Disturbance. In Proceedings of the 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 10–12 December 2021; pp. 480–484.
37. Guo, Z.; Wang, K.; Li, W.; Qian, Y.; Arandjelović, O.; Fang, L. Artwork Protection Against Neural Style Transfer Using Locally Adaptive Adversarial Color Attack. *arXiv* **2024**, arXiv:2401.09673.
38. Guo, Z.; Qian, Y.; Arandjelović, O.; Fang, L. A white-box false positive adversarial attack method on contrastive loss-based offline handwritten signature verification models. *arXiv* **2023**, arXiv:2308.08925.
39. Müller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
40. Cortes, C.; Mohri, M.; Rostamizadeh, A. L2 regularization for learning kernels. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 109–116.
41. Zhao, S.; Li, Q.; He, T.; Wen, J. A Step-by-Step Gradient Penalty with Similarity Calculation for Text Summary Generation. *Neural Process. Lett.* **2023**, *55*, 4111–4126. [CrossRef]
42. Dong, J.; Wang, Y.; Lai, J.H.; Xie, X. Improving Adversarially Robust Few-Shot Image Classification With Generalizable Representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 9025–9034.
43. Wei, J.; Zhang, Y.; Zhou, Z.; Li, Z.; Al Faruque, M.A. Leaky dnn: Stealing deep-learning model secret with gpu context-switching side-channel. In Proceedings of the 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Valencia, Spain, 29 June–2 July 2020; pp. 125–137.
44. Zhang, Y.; Yasaei, R.; Chen, H.; Li, Z.; Al Faruque, M.A. Stealing neural network structure through remote FPGA side-channel analysis. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4377–4388. [CrossRef]

45. Dong, J.; Moosavi-Dezfooli, S.M.; Lai, J.; Xie, X. The Enemy of My Enemy Is My Friend: Exploring Inverse Adversaries for Improving Adversarial Training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 24678–24687.
46. Zhao, S.; Gan, L.; Tuan, L.A.; Fu, J.; Lyu, L.; Jia, M.; Wen, J. Defending Against Weight-Poisoning Backdoor Attacks for Parameter-Efficient Fine-Tuning. *arXiv* **2024**, arXiv:2402.12168.
47. Tian, J.; Shen, C.; Wang, B.; Xia, X.; Zhang, M.; Lin, C.; Li, Q. LESSON: Multi-Label Adversarial False Data Injection Attack for Deep Learning Locational Detection. *IEEE Trans. Dependable Secur. Comput.* **2024**. [CrossRef]
48. Katsikeas, S.; Johnson, P.; Hacks, S.; Lagerström, R. Probabilistic Modeling and Simulation of Vehicular Cyber Attacks: An Application of the Meta Attack Language. In Proceedings of the ICISSP, Prague, Czech Republic, 23–25 February 2019; pp. 175–182.
49. Zhao, S.; Jia, M.; Tuan, L.A.; Pan, F.; Wen, J. Universal Vulnerabilities in Large Language Models: Backdoor Attacks for In-context Learning. *arXiv* **2024**, arXiv:2401.05949.
50. Tian, J.; Wang, B.; Guo, R.; Wang, Z.; Cao, K.; Wang, X. Adversarial attacks and defenses for deep-learning-based unmanned aerial vehicles. *IEEE Internet Things J.* **2021**, *9*, 22399–22409. [CrossRef]
51. Zhao, S.; Wen, J.; Luu, A.; Zhao, J.; Fu, J. Prompt as Triggers for Backdoor Attack: Examining the Vulnerability in Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 12303–12317.
52. Zhang, R.; Yu, Y.; Zhang, C. SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup. In Proceedings of the EMNLP, Online, 16–20 November 2020.
53. Guo, H. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
54. Zhang, S.; Jiang, L.; Tan, J. Dynamic Nonlinear Mixup with Distance-based Sample Selection. In Proceedings of the ICCL, Barcelona, Spain, 21–23 September 2022.
55. Jiang, W.; Chen, Y.; Fu, H.; Liu, G. TextCut: A Multi-region Replacement Data Augmentation Approach for Text Imbalance Classification. In Proceedings of the ICONIP, Sanur, Indonesia, 8–12 December 2021.
56. Yang, Y.; Lin, Y.; Chen, Z.; Lei, Y.; Liu, X.; Zhang, Y.; Sun, Y.; Wang, X. SNPERS: A Physical Exercise Recommendation System Integrating Statistical Principles and Natural Language Processing. *Electronics* **2022**, *12*, 61. [CrossRef]
57. Li, J.; Yao, M. Dynamic Evolution Mechanism of Digital Entrepreneurship Ecosystem Based on Text Sentiment Computing Analysis. *Front. Psychol.* **2021**, *12*, 725168. [CrossRef] [PubMed]
58. Jazib, A.; Tariq, W.; Mahmood, M. Sentiment Analysis using Ensemble Classifier for Entrepreneurs based on Twitter Analytics. In Proceedings of the 2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 16–20 August 2022; pp. 207–212.
59. Li, Z.; Zhu, H.; Liu, H.; Song, J.; Cheng, Q. Comprehensive evaluation of Mal-API-2019 dataset by machine learning in malware detection. *arXiv* **2024**, arXiv:2403.02232.
60. Zhu, M.; Zhang, Y.; Gong, Y.; Xing, K.; Yan, X.; Song, J. Ensemble Methodology: Innovations in Credit Default Prediction Using LightGBM, XGBoost, and LocalEnsemble. *arXiv* **2024**, arXiv:2402.17979.
61. Malik, A.; Onyema, E.M.; Dalal, S.; Lilhore, U.K.; Anand, D.; Sharma, A.; Simaiya, S. Forecasting students' adaptability in online entrepreneurship education using modified ensemble machine learning model. *Array* **2023**, *19*, 100303. [CrossRef]
62. Chen, X. Internet plus innovation and entrepreneurship education model based on machine learning algorithms. *Mob. Inf. Syst.* **2022**, *2022*, 6176675. [CrossRef]
63. He, M.; Zhang, J. Evaluating the innovation and entrepreneurship education in colleges using BP neural network. *Soft Comput.* **2023**, *27*, 14361–14377. [CrossRef]
64. Liu, T.; Xu, C.; Qiao, Y.; Jiang, C.; Chen, W. News Recommendation with Attention Mechanism. *J. Ind. Eng. Appl. Sci.* **2024**, *2*, 21–26.
65. Su, J.; Jiang, C.; Jin, X.; Qiao, Y.; Xiao, T.; Ma, H.; Wei, R.; Jing, Z.; Xu, J.; Lin, J. Large Language Models for Forecasting and Anomaly Detection: A Systematic Literature Review. *arXiv* **2024**, arXiv:2402.10350.
66. Jia, M.; Liao, L.; Wang, W.; Li, F.; Chen, Z.; Li, J.; Huang, H. Keywords-aware dynamic graph neural network for multi-hop reading comprehension. *Neurocomputing* **2022**, *501*, 25–40. [CrossRef]
67. Wang, X.; Zhang, H.; Huang, W.; Scott, M.R. Cross-batch memory for embedding learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6388–6397.
68. Zampieri, M.; Malmasi, S.; Nakov, P.; Rosenthal, S.; Farra, N.; Kumar, R. Predicting the Type and Target of Offensive Posts in Social Media. In Proceedings of the NAACL, Minneapolis, MN, USA, 2–7 June 2019; pp. 1415–1420.
69. Warstadt, A.; Singh, A.; Bowman, S.R. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 625–641. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.