Depthwise Separable Convolutional Neural Network for Confidential Information Analysis

Yue Lu[1,2], Jianguo Jiang[1,2], Min Yu[1,2(✉)], Chao Liu[1], Chaochao Liu[1,2], Weiqing Huang[1], and Zhiqiang Lv[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
yumin@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. Confidential information analysis can identify the text containing confidential information, thereby protecting organizations from the threat posed by leakage of confidential information. It is effective to build a confidential information analyzer based on a neural network. Most of the existing studies pursue high accuracy to design complex networks, ignoring speed and consumption. The optimal defense is to automatically analyze confidential information without compromising routine services. In this paper, we introduce a lightweight network, DSCNN, that can be adapted to low-resource devices. We also introduce two hyperparameters to balance accuracy and speed. Our motivation is to simplify convolutions by breaking them down because the space dimension and channel dimension are not closely related in the convolutions. Experimental results on real-world data from WikiLeaks show that our proposed DSCNN performs well for confidential information analysis.

Keywords: Depthwise Separable Convolutional Neural Network · Confidential information analysis · Information Security · Natural Language Processing

1 Introduction

Mobile devices generate a vast amount of data every minute. The data may contain confidential information that has not yet been marked. Such confidential information can be leaked without being noticed and pose a threat to national security, business trade, or personal life. Many organizations institute enforcement policies to protect confidential information [18]. These policies require every email sent by an employee to the Internet is reviewed by his manager. On the one hand, these policies limit the operating efficiency of the organization and waste a lot of manpower resources. On the other hand, it is ineffective if the employee's managers are not extremely well-versed in the scope of confidential matters for

G. Li et al. (Eds.): KSEM 2020, LNAI 12275, pp. 450–461, 2020.
https://doi.org/10.1007/978-3-030-55393-7_40

the entire organization. Therefore, constructing confidential information analyzers has become a trend. As it can help organizations identify confidential information, the need for high quality automated confidential information analyzers becomes much more profound.

Confidential information analysis is a cross-task of Information Security (IS) and Natural Language Processing (NLP). Its goal is to categorize text into different security levels (such as Confidential and Non-Confidential), or more fine-grained levels (such as Top-Secret, Secret, Confidential, and Unclassified) [1]. In similar areas of confidential information analysis, such as text mining [17], sentiment analysis [8], fake news detection [15], and confidential information detection [7],CNNs have attracted extensive attention because of their excellent performance. A large number of deep and complex CNNs have been designed for tasks related to text classification [14]. Conneau et al. used 19 convolution layers to build a Very Deep CNN (VDCNN) [3]. Johnson et al. built a Deep Pyramid CNN (DPCNN) with 15 layers [11]. These models can easily achieve high accuracy with sufficient computational resources and processing time. However, such models do not work well in low-resource and time-limited devices or applications. In real-world applications such as Data Leakage Prevention (DLP) [16] and Security Information and Event Management (SIEM) [21], confidential information analyzers require the ability to run in real-time on a computationally limited device to enforce the appropriate protection mechanism without degrading regular services.

In this paper, we present a lightweight model named **D**epthwise **S**eparable **C**onvolutional **N**eural **N**etwork (**DSCNN**) for confidential information analysis. Our motivation is that separating spaces and channels when convoluting text can reduce the computational complexity of convolutions. The space dimension and channel dimension are not closely related that it is preferable not to map them together. From the perspective of a model, channels are different pre-trained word embeddings without strict sequence. Additionally, we describe two hyper-parameters that efficiently balance accuracy and speed. These two hyper-parameters can be used when designing the appropriate size models for different devices. We conduct the comparison experiments of our proposed method and other popular methods. We also conduct extensive experiments of hyper-parameter sensitivity. The results show that our proposed method has a better performance in analyzing confidential information. The main contributions of this work include:

1) We propose a DSCNN for confidential information analysis. The DSCNN makes convolution easier by operating in space dimension and channel dimension respectively to reduce computational complexity.
2) We introduce two simple hyper-parameters, channel multiplier and space multiplier, to balance accuracy and speed. These two hyper-parameters can be used to further reduce the amount of calculation.
3) Extensive experiments using real-world data from WikiLeaks show that our proposed model not only achieves a high accuracy but also saves a lot of time and resources.

The rest of the paper is categorized as follows. The previous approaches are reviewed in Sect. 2. Our proposed method is presented in Sect. 3. Section 4 presents the results of the experiment. Finally, in Sect. 5, we conclude and give perspectives.

2 Preliminaries

Recently, CNNs have achieved strong performance for tasks related to text classification [2]. Dos Santos et al. designed a CNN to extract features at the character-level and word-level for sentiment analysis [4]. Kalchbrenner et al. introduced a global pooling named dynamic k-max pooling to build CNNs [12]. Kim used various pre-trained word embeddings to build CNNs for sentence classification [13]. Yin et al. presented a multi-channel CNN which accepts multiple word embeddings as the input [19]. Johnson et al. proposed a low-complexity CNN shaped like a pyramid for text categorization [11].

The CNN is a kind of neural network model, which relies on the layer called convolution layer for feature extraction. At the heart of this convolution layer is a learnable filter. This filter does convolution operation while scanning the vectorial text. The outputs of the convolution operation are the extracted feature maps. Suppose a learnable filter $\mathbf{w} \in \mathbb{R}^{h \times d}$ is scanning the text sequence $\mathbf{x} \in \mathbb{R}^{s \times d}$ The h represents the size of the filter $\mathbf{w}$, the s represents the length of the sequence $\mathbf{x}$, and the d represents the dimension of the word embeddings. A extracted feature is obtained by the convolution operation:

$$f_i = \sigma(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + \mathbf{b}), \tag{1}$$

where the $\mathbf{x}_{i:i+h-1}$ representations a region of the above text sequence, the $\mathbf{w}$ represents the above filter, the $\mathbf{b}$ represents a bias, and the σ represents a non-linear function. The filter scans all the sequences $\{\mathbf{x}_{1:h}, \mathbf{x}_{2:h+1}, \ldots, \mathbf{x}_{s-h+1:s}\}$ to produce a feature map $\mathbf{f} = [f_1, f_2, \ldots, f_{s-h+1}]$. Typically models use a lot of filters (with different window sizes) to obtain a variety of feature maps.

The multi-channel CNN is an improvement on the single-channel CNN. Compared with single-channel CNN, the multi-channel CNN uses different pre-training word embedding vectors to initialize multiple channels as inputs. The multi-channel CNN brings the following advantages: On the one hand, multiple channels rather than one channel can bring more information available to a common word. On the other hand, one channel can be missing a rare word, other channels can supplement it. However, it is worth noting that the multi-channel model will bring lots of computation. In multi-channel CNN, the convolution layer attempts to learn the filter in three dimensions, which has two space dimensions (width and height) and one channel dimension. The convolution operation is a joint mapping of these dimensions. In single-channel CNN, the convolution operation is only a mapping of space dimensions. We think the space dimension and the channel dimension are not closely related. The channel dimension do not have the same strict order as the space dimension in the model, so we consider that joint mapping is unnecessary. To simplify the convolution operations, the

standard convolution can be decomposed into a series of operations that convolving independently on spaces and channels. Our work is similar to Howard et al. [6] on image classification.

3 Methodology

In this section, we first present the network structure of DSCNN. We then detail the core of DSCNN – depthwise separable convolution. Finally, we describe two hyper-parameters – channel multiplier and space multiplier.

3.1 Network Structure of DSCNN

DPCNN [11] is a simple network structure with less computation and better performance. It is a pyramid-shaped network structure whose upper layer is half the size of the lower layer. We use the DPCNN network structure as the basic network of our DSCNN.

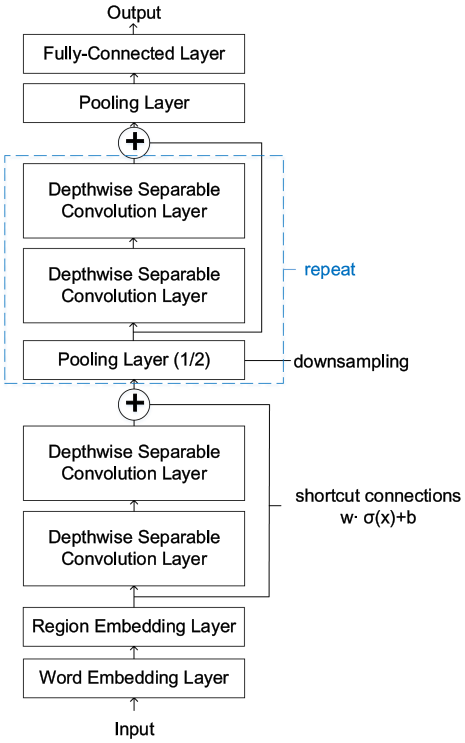

Fig. 1. Network structure of DSCNN.

The network structure of DSCNN is illustrated in Fig. 1. A DSCNN model consists of input, word embedding layer, region embedding layer, depthwise separable convolution layer, pooling layer, fully-connected layer, and output. In our DSCNN, the input is a text sequence and the output is its label. The first layer is a word embedding layer. The word embedding layer is used to convert text into vectors. In our DSCNN, we use a variety of pre-trained word embeddings to initialize the word embedding layer. Each channel of the layer corresponds to a pre-trained word embedding. So our DSCNN is also a multi-channel CNN. As mentioned in Sect. 2, multi-channel networks have more advantages than single-channel networks. The second layer is a region embedding layer. The region embedding layer works in a similar way to the N-gram model [9,10]. It is used to extract features of a small region in the width and height of space dimension. The following layers are the alternations of two depthwise separable convolution layers and one pooling layer. The depthwise separable convolution layer is used to extract features and model long distance dependencies. It extracts features in three dimensions, which has two space dimensions (width and height) and one channel dimension. After each depthwise separable convolution layer, there is a pooling layer used to downsample feature maps. We use the max-pooling with size 3 and stride 2 in the pooling layer. We also fix the number of feature maps in this pooling layer like DPCNN. With this pooling layer, we can model longer distance dependencies later by depthwise separable convolution layers. To enable the training of deep networks, we use shortcut connections with pre-activation. The shortcut connections with pre-activation can be written as $\mathbf{w}\sigma(\mathbf{x}) + b$ [5]. Finally, there is a fully-connected layer to generate the final classification.

3.2 Depthwise Separable Convolution

The core of DSCNN is the depthwise separable convolution. Compared to a standard convolution, the depthwise separable convolution can greatly reduce computational complexity. Hereafter we use the notation given in Table 1.

Table 1. This table lists notation declarations.

Symbol	Meaning
h	Height
w	Width
d	Channel
$\mathrm{K}_{\mathrm{h_K}}, \mathrm{wK}, \mathrm{d_F}, \mathrm{d_G}$	A filter
$\mathrm{F}_{\mathrm{h_F}}, \mathrm{wF}, \mathrm{d_F}$	A input feature map
$\mathrm{G}_{\mathrm{h_G}}, \mathrm{wG}, \mathrm{d_G}$	A output feature map

As shown in Fig. 2(a), the standard convolution use d_G filters of size $h_K \times w_K \times d_F$ to extract feature maps. It extracts features in both the space dimension

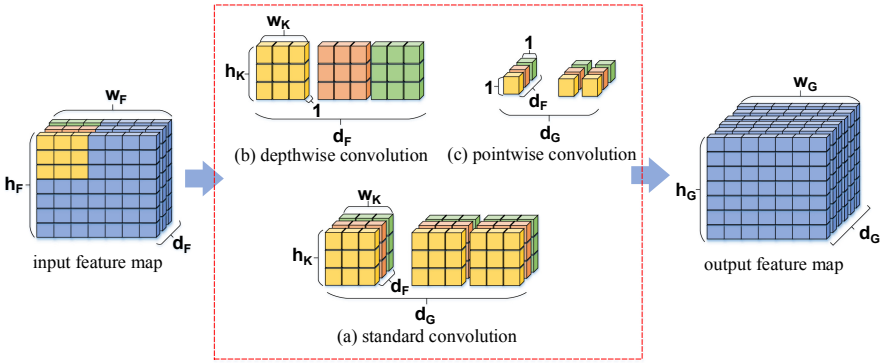

Fig. 2. The standard convolution extracts the space and channel features in one step (a), while the depthwise separable convolution extracts space features in the depthwise convolution (b) and extracts channel features in the pointwise convolution (c).

and the channel dimension. The h_K represents the height of the filter and the w_K represents the width of the filter in the space dimension. The d_F represents the channel number of the filter in the channel dimension. The channel number of the filter is the same as the channel number of the input feature maps because the filter convolves on a multi-channel input. The d_G represents the number of filters. The number of filters is the same as the channel number of the output feature maps.

As shown in Fig. 2(b), (c), the depthwise separable convolution contains two parts: a depthwise convolution and a pointwise convolution. The depthwise separable convolution is decomposed to operate separately in the space dimension and channel dimension. The depthwise convolution is operating in the space dimension, while the pointwise convolution is operating in the channel dimension. The depthwise convolution use d_F depthwise filters of size $h_K \times w_K \times 1$ to extract feature maps in the space dimension. The h_K represents the height of the depthwise filter, the w_K represents the width of the depthwise filter, and the 1 represents the channel number of the depthwise filter. The d_F represents the number of depthwise filters. The number of depthwise filters is the same as the input channel because one input channel corresponds to one depthwise filter. The pointwise convolution use d_G pointwise filters of size $1 \times 1 \times d_F$ to extract feature maps in the channel dimension. The 1×1 represent respectively the height and width of the pointwise filter. The d_F represents the channel number of the pointwise filter. The channel number of the pointwise filter is the same as the number of the depthwise filter. The d_G represents the number of pointwise filters. The number of pointwise filters is the same as the channel number of the output feature maps.

As discussed in Sect. 1, we think the space dimension and the channel dimension are not closely related. The standard convolution extracts the space and channel features in one step, while the depthwise separable convolution extracts

space features in one step and extracts channel features in another step. Splitting one step into two steps can reduce computational complexity. Suppose we take $\mathrm{F_{h_F,wF,d_F}}$ as the input feature maps and $\mathrm{G_{h_G,wG,d_G}}$ as the output feature maps. The h_F represents the height of the input feature maps, the w_F represents the width of the input feature maps, and the d_F represents the channel number of input feature maps. The h_G represents the height of the output feature maps, the w_G represents the width of the output feature maps, and the d_G represents the channel number of output feature maps. The computational complexity of the standard convolution is:

$$h_K \cdot w_K \cdot d_F \cdot d_G \cdot h_F \cdot w_F. \tag{2}$$

The depthwise convolution costs:

$$h_K \cdot w_K \cdot d_F \cdot h_F \cdot w_F. \tag{3}$$

The pointwise convolution costs:

$$d_F \cdot d_G \cdot h_F \cdot w_F. \tag{4}$$

The depthwise separable convolution costs:

$$h_K \cdot w_K \cdot d_F \cdot h_F \cdot w_F + d_F \cdot d_G \cdot h_F \cdot w_F\,, \tag{5}$$

We get a reduction:

$$\frac{h_K \cdot w_K \cdot d_F \cdot h_F \cdot w_F + d_F \cdot d_G \cdot h_F \cdot w_F}{h_K \cdot w_K \cdot d_F \cdot d_G \cdot h_F \cdot w_F} = \frac{1}{d_G} + \frac{1}{h_K \cdot w_K}. \tag{6}$$

3.3 Channel Multiplier and Space Multiplier

To balance accuracy and speed for more applications, we introduce a simple parameter α called *channel multiplier*. It is used to set the channel number of feature maps and thin the depthwise separable convolution layer. The computational cost is:

$$h_K \cdot w_K \cdot \alpha d_F \cdot h_F \cdot w_F + \alpha d_F \cdot \alpha d_G \cdot h_F \cdot w_F\,, \tag{7}$$

where $\alpha \in (0, 1]$. $\alpha = 1$ is the baseline DSCNN and $\alpha < 1$ are reduced DSCNNs. We introduce the other parameter β called *space multiplier*. It is used to set the space size of feature maps and reduce the resolution of the input feature maps. The computational cost is:

$$h_K \cdot w_K \cdot d_F \cdot \beta h_F \cdot \beta w_F + d_F \cdot d_G \cdot \beta h_F \cdot \beta w_F\,, \tag{8}$$

where $\beta \in (0, 1]$. $\beta = 1$ is the baseline DSCNN and $\beta < 1$ are reduced DSCNNs.

4 Experiments and Discussion

We evaluated a variety of models on WikiLeaks Cable Dataset in this section. The purpose of these experiments is to clarify the influence of our proposed DSCNN for confidential information analysis.

4.1 Experiments Settings

Dataset: The WikiLeaks Cable Dataset consists of paragraphs extracted from Public Library of US Diplomacy (PlusD). We use white space as a delimiter, normalize punctuations, remove special characters, and convert the remaining characters to lowercase. After pre-processing, the details on the dataset are provided in Table 2. We randomly choose 80% of the original dataset for training and 10% of the original dataset for testing. The rest 10% of the original dataset to construct a validation set. We maintain a Secret/Confidential/Unclassified balance of the original dataset in each split and use 10-fold cross-validation.

Table 2. Statistics of wikiLeaks cable dataset.

Item	Content
Name	WikiLeaks cable dataset
Type	Sentence-level
#Classes	3
#Instances of secret	10,000
#Instances of confidential	10,000
#Instances of unclassified	10,000
Average length	145
Vocabulary size	125, 534
Test	10-fold CV

Hyper-parameters: We tune the hyper-parameters of our proposed model on the validation set.

- **Pre-trained Word Embeddings:** We initialize the word embedding layer with the following pre-trained word embeddings. We set the channel number of the word embedding layer as 4 and the dimension of vectors as 300. These pre-trained word embeddings are available on github[1]. We use the vectors of Word2Vec-GoogleNews, Word2VecModified-Wikipedia, GloVe-Crawl840B and GloVe-Wikipedia. The vectors of these word embedding do not fine-tune during training the classifiers. The Word2Vec-GoogleNews are trained on Google News through Word2Vec. The Word2VecModified-Wikipedia are trained on Wikipedia through modified Word2vec. The GloVe-Crawl840B are trained on Common Crawl through GloVe. The GloVe-Wikipedia are trained on Wikipedia through GloVe.
- **Hyper-Parameters in DSCNN:** We set the depth of DSCNN as 16, 14 convolution layers plus 2 embedding layers. We set the window size of the region embedding layer as 1, 3, 5 and the channel number of feature maps as 250. We train models with a mini-batch size of 64 and use Adam optimizer

[1] https://github.com/3Top/word2vec-api.

with the learning rate of 0.001. We use a 0.5 dropout rate on the fully-connected layer during training.

Evaluation. We use *accuracy* to measure these models because the dataset is balanced. We use *calculation* to evaluate the computational complexity of these model. With the same device configuration, the less computation is, the faster the speed is. We use *parameters* to evaluate the spatial complexity of these model. The fewer parameters, the less space.

4.2 Results and Discussion

Main Comparisions. First we show results for our proposed DSCNN based on the depthwise separable convolutions compared to other popular CNN models. The CharSCNN is a shallow network that extracts features from character-level to sentence-level. The TextCNN is a shallow network based on the word embedding. The MVCNN and MCCNN are multi-channel networks that have rich feature maps in convolution layers. The ConvNets and VDCNN are deep networks based character-level representation. The DPCNN is the state-of-the-art network for text classification. The Multi-Channel DPCNN is a network modified by us that use the diversity of different embedding to extract higher quality features. Compared the Multi-Channel DPCNN and our proposed DSCNN, the Multi-Channel DPCNN is based on the standard convolutions while the DSCNN is based on the depthwise separable convolutions.

Table 3. Results of our proposed DSCNN against other models.

Model	Type	Accuracy	Calculation	Parameters
CharSCNN [4]	shallow, char	64.51	–	–
TextCNN [13]	shallow, word	66.46	–	–
MVCNN [19]	shallow, word	68.17	–	–
MCCNN [2]	shallow, word	68.02	–	–
ConvNets [20]	deep, char	66.95	–	–
VDCNN [3]	deep, char	67.16	1503.36M	2.11M
DPCNN [11] (Baseline)	deep, word	68.85	1370.25M	2.63M
Multi-Channel DPCNN	deep, word	72.34	5481.00M	2.63M
DSCNN (Ours)	deep, word	**72.57**	**630.92M**	**0.89M**

From Table 3, we have the following observations: (1) As expected, our proposed DSCNN not only achieves a high accuracy but also saves a lot of time and resources. It costs less computation than the standard convolutions. The DSCNN uses $3 \times 3 \times 250 \times 250$ depthwise separable convolutions which use about 9 times

less computation than the Multi-Channel DPCNN with the standard convolutions. Additionally, the DSCNN primarily focus on optimizing for calculation speed but also yield a small size network. (2) Deep models with multiple channels indeed give better performances. Single-channel networks do not outperform multi-channel networks. A single-channel network – DPCNN – achieves 68.85%, comparing to 72.34% of a multi-channel network – Multi-Channel DPCNN. It demonstrates the effectiveness of pre-trained word embeddings. The pre-trained word embedding vectors can introduce more useful external knowledge for short text.

Model Shrink. Table 4 shows a comparison between our proposed DSCNNs with different channel multipliers and the baseline DPCNN. We analyze the results from three aspects: accuracy, calculation, and parameters. The channel multiplier is used to set the channel number of feature maps. We observe that decreasing the channel multiplier α hurts the accuracy, but can reduce the calculation and parameters. Our proposed DSCNN with channel multiplier $\alpha = 0.75$ has 3 times less calculation and 5 times fewer parameters with the same accuracy as the baseline DPCNN.

Table 4. Results of channel multiplier (α).

Model	Accuracy	Calculation	Parameters
DPCNN (Baseline)	**68.85**	**1370.25M**	**2.63M**
1.00 DSCNN – 300	72.57	630.92M	0.89M
0.75 DSCNN - 300	**69.05**	**359.01M**	**0.50M**
0.50 DSCNN – 300	65.02	163.21M	0.22M
0.25 DSCNN – 300	60.17	43.54M	0.06M

Table 5. Results of space multiplier (β).

Model	Accuracy	Calculation	Parameters
DPCNN (Baseline)	**68.85**	**1370.25M**	**2.63M**
1.00 DSCNN – 300	72.57	630.92M	0.89M
1.00 DSCNN – 224	71.83	471.09M	0.89M
1.00 DSCNN – 192	70.66	403.79M	0.89M
1.00 DSCNN – 160	**68.81**	**336.49M**	**0.89M**
1.00 DSCNN – 128	65.74	269.19M	0.89M

Table 5 shows a comparison between our proposed DSCNNs with different space multipliers and the baseline DPCNN. We analyze the results from three

aspects: accuracy, calculation, and parameters. The space multipliers is used to set the space size of feature maps. The accuracy decreases as the space size of feature maps decreases. The calculation reduces as the space size of feature maps decreases. The parameters remains the same because it is independent of the space size of feature maps. Our proposed DSCNN with space multiplier $\beta = 160$ has 4 times less calculation and 3 times fewer parameters with the same accuracy as the baseline DPCNN.

5 Conclusion

Confidential information analysis can protect organizations from the threat of confidential information leakage by identifying text that contains confidential information. In this paper, we proposed a lightweight model named DSCNN based on depthwise separable convolutions for improving the performance of confidential information analysis. The proposed method convolves in space and channel dimensions respectively, which can reduce the computational complexity of convolution operation. We then described the channel multiplier and space multiplier to balance accuracy and speed to fit different low-resource devices. We expect that separable convolution in depth will become the cornerstone of the design of CNNs in the future since they make the convolution easier and more efficient on multi-channel CNNs.

Acknowledgment. This work is supported by National Natural Science Foundation of China (No. 71871090).

References

1. Alzhrani, K.M.: Towards automating big texts security classification. Ph.D. thesis, University of Colorado Colorado Springs. Kraemer Family Library (2018)
2. Chen, K., Liang, B., Ke, W., Xu, B., Zeng, G.: Chinese micro-blog sentiment analysis based on multi-channels convolutional neural networks. J. Comput. Res. Dev. **55**(5), 945–957 (2018)
3. Conneau, A., Schwenk, H., Barrault, L., Lecun, Y.: Very deep convolutional networks for text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017) (2017)
4. Dos Santos, C., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (COLING 2014), pp. 69–78 (2014)
5. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
6. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
7. Jiang, J., et al.: CIDetector: semi-supervised method for multi-topic confidential information detection. In: The 24th European Conference on Artificial Intelligence (ECAI 2020) (2013)

8. Jiang, J., et al.: Sentiment embedded semantic space for more accurate sentiment analysis. In: Liu, W., Giunchiglia, F., Yang, B. (eds.) KSEM 2018. LNCS (LNAI), vol. 11062, pp. 221–231. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99247-1_19
9. Johnson, R., Zhang, T.: Effective use of word order for text categorization with convolutional neural networks (2015)
10. Johnson, R., Zhang, T.: Semi-supervised convolutional neural networks for text categorization via region embedding. In: Advances in Neural Information Processing Systems (NIPS 2015), pp. 919–927 (2015)
11. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2017), pp. 562–570. Association for Computational Linguistics (2017)
12. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), pp. 655–665. Association for Computational Linguistics (2014)
13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1746–1751. Association for Computational Linguistics (2014)
14. Law, R., Li, G., Fong, D.K.C., Han, X.: Tourism demand forecasting: a deep learning approach. Ann. Tour. Res. **75**, 410–423 (2019)
15. Liu, C., et al.: A two-stage model based on BERT for short fake news detection. In: Douligeris, C., Karagiannis, D., Apostolou, D. (eds.) KSEM 2019. LNCS (LNAI), vol. 11776, pp. 172–183. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29563-9_17
16. Shvartzshnaider, Y., Pavlinovic, Z., Balashankar, A., Wies, T., Subramanian, L., Nissenbaum, H., Mittal, P.: Vaccine: using contextual integrity for data leakage detection. In: The World Wide Web Conference (WWW 2019), pp. 1702–1712. ACM (2019)
17. Vu, H.Q., Li, G., Law, R., Zhang, Y.: Exploring tourist dining preferences based on restaurant reviews. J. Travel Res. **58**(1), 149–167 (2019)
18. Yerazunis, W., Kato, M., Kori, M., Shibata, H., Hackenberg, K.: Keeping the good stuff. In: Confidential Information Firewalling with the CRM114 Spam Filter & Text Classifier. White Paper Black Hat USA (2010)
19. Yin, W., Schütze, H.: Multichannel variable-size convolution for sentence classification. In: Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2016) (2016)
20. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems (NIPS 2015), pp. 649–657 (2015)
21. Zhu, T., Li, G., Zhou, W., Yu, P.S.: Differential Privacy and Applications. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62004-6_9