ResSleepNet: Automatic sleep stage classification on raw single-channel EEG

Cheng Liu[1], Tongfeng Weng[2] and Xinhua Liu[2]*

1.No. 15 Middle School of Wuhan , WuHan, China; 2.School of Information Engineering, Wuhan University of Tecnology, WuHan, China
Email: *liuxinhua@whut.edu.cn

Abstract. Automatic sleep stage classification is an important paradigm in intelligence and promises considerable advantages to the health. Electroencephalography (EEG) is a reflection of the electrophysiological activities of brain neurons. Most current automated methods require the multiple electroencephalogram channels and rely on hand-engineered features which require prior knowledge about sleep stage scoring. Consequently, research efforts to purse for an automatic stage scoring based on machine learning techniques have been carried out over the last years. In this paper, a deep learning model, named ResSleepNet, is proposed for automatic sleep stage scoring based on raw single-channel EEG, it can automatically learn features from raw single channel EEG signal, and build an automatic sleep staging model for assisted sleep staging. The model is applied to an open-access database named Sleep-EDF, and the results demonstrated that the model scored the EEG epochs with the accuracy of 87.9%.

1. Introduction

Sleep plays an important role in human health. Sleep is also a complex amalgam of physiologic and behavioural processes [1]. The function of sleep is not fully understood yet, and among the many hypotheses proposed, the most widely accepted ones are brain thermoregulation, brain detoxification, tissue restoration and metabolic homeostasis[2,3]. Being able to monitor how well do people sleep in each day have a significant impact on medical research and practice[4].Human sleep is divided in to multiple stages that manifest in a cyclical fashion throughout the night. Including wake and movement, there are seven stages of sleep according to R&K [5] classification and five according to the more recent AASM [6] classification. We use AASM, which include five stages: wake(W), stage_1(S1), stage_2(S2), stage_3(S3), REM. Generally, human experts determine the quality of sleep using electrical activity record from sensors attached to different parts of the body. A set of signals from these sensors is called polysomnogram(PSG), consisting of electroencephalogram(EEG), electrooculogram(EOG), electromyogram(EMG), and electrocardiogram (ECG). A typical sleep study involves monitoring of neural, eye movement and muscle activity by recording multiple EEG, EOG and EMG signals for the duration of sleep. These signals are then segmented in to epochs of 30 second and are subsequently scored by human experts, assigning a sleep stage to each epoch. The main limitation of epoch based visual sleep scoring is that it is extremely time-consuming. The scoring of 8hours overnight PSG takes approximately 2-4 hours for an expert. In addition, we need to get an algorithm to classify sleep stage from EEG automatic.

To alleviate the aforementioned problems, there have been a number of studies trying to develop a method to automate sleep stage scoring based on multiple signals such as EEG, EOG and EMG, or single-channel EEG. These methods are typically based on feature extraction techniques and pattern recognition algorithms. In terms of the analysis data, the Sleep-EDF expand database [7] on Physionet has been available for more than 10 years. We also use this database to train a test our ResSleepNet. For datasets extracted from Sleep-EDF expand database, Imtiaz et al.[8] calculated spectral band power and spectral edge frequency (SEF) from two EEG channels. These methods correctly classified 82.2% of epochs for a 5-stage task. Zhu et al. [9] classified the sleep stages based on graph domain features, and achieved an accuracy of 89.0%. Hassan et al. [10] applied empirical mode decomposition (EMD) analysis for a single lead EEG montage, and calculated 1st to 4th statistical moments for each intrinsic mode function (IMF). Their proposed method achieved 90.7% accuracy by bootstrap aggregating decision tree. Recently, Silveria et al.[11] reported 91.5% accuracy for 5-stage classification problems when using discrete wavelet transform and random forests.

In this paper, a deep learning model, named ResSleepNet, is proposed, which just use raw single-channel EEG named Fpz-Cz. We aim to automate the process of hand-engineering features by utilizing the feature extraction capabilities of deep learning.

2. Database procession

The database procession contains two parts. One of the parts is analysis source data, another part is generating training and testing data.

2.1. Analysis source data

PhysioNet Sleep EDF database has been the most popular source of data used for developing and testing many automatic sleep staging algorithms. The database was known as the Sleep EDF Expanded Database which includes 61 recordings. Signals in the database are segmented into epochs of 30 seconds and are subsequently scored by human experts, assigning a sleep stage to each epoch. We use an open source toolbox for Matlab to make the sleep data more accessible to be used in our model. The conversion of EDF files requires the use of EEGLAB, an open source toolbox for EEG analysis[15]. An error will be shown if the EEGLAB toolbox is not found on the user's system. The resulting .mat data files are stored in a folder named Matlab under the test directory. Additionally, the list of channels for which data are available is also saved in the same folder. Then we need download and process the annotations for each recording. This is the file that consists of sleep stage score corresponding to each epoch of a test case. There are some important information in the file, including the start time of the recording, sampling frequencies and the duration for which each sleep stage has been classified until the end of recording. We can use downloadEDFxAnnotations() function to get those annotations. After complete the above two steps, we get an annotations with .txt format and a signal file with .mat file. For example, eeg_fpz.mat and SC4001E0.txt.

2.2. Generating training and testing data

After we get the origin signal from the SleepEDF database, but the data form samples by 100Hz of the .mat file is discrete, as we all know, CNN is always applied to the field of computer vision. So, we propose a function that apply convolution to one-dimensional data. This article uses Fpz-Cz channel EEG data. After extracting the data, it is found that there is a serious imbalance in the data. As shown in the following table 1.Severe imbalance of the classification data will lead to training results biased toward a large amount of data, making the final training results poorly fit. The upsampling algorithm was used in the experiment. Here we randomly replicate the samples in the group and determine the number of target samples as the maximum number of samples. Here is 17176.Then for other samples randomly selected samples for replication until the number of samples reached 17176, the final realization of the number of samples for each category were 17176.

Table 1. train/val data set

sleep stage	train-set	val-set
sleep stage W	7914	371
sleep stage 1	2687	117
sleep stage 2	17176	623
sleep stage 3	5186	517
REM	7377	340

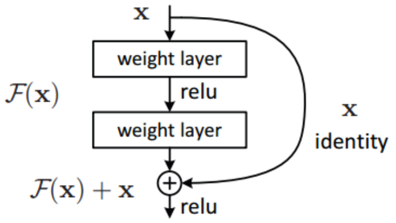
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 1. base res-block

3. Ressleepnet

The construction of the deep network model used in this paper involves two steps. The first is to build a custom residual module and then use it to build DeepSleepNet.

3.1. Custom residual block

The concept of residual network was proposed by He Kaiming, the Microsoft Asia Research Institute, in the deep residual network. Let a hidden layer be $H(x) - x \rightarrow F(x)$, assuming that multiple nonlinear combinations can be approximated as a complex function, then one can also assume that the hidden layer's residual approximates to a more complex function [12], that is, the hidden layer can be represented as $H(x)=F(x)+x$. This results in a completely new residual structure unit, as shown in Figure 1.From the figure, we can see that the output of the residual unit is obtained by adding the outputs of multiple convolution cascades and the input elements of the unit. Here, it must be ensured that the output and input dimensions of the convolutional layer are the same, otherwise, the concatenation layer has different dimensions. After the ReLU activation function is output, this is one of the reasons why ReLU was selected as a network model activation function. Based on the above residual units in the deep residual network, the residual cell structure suitable for the system is designed in combination with the features of the EEG data, as shown in Figure 2.In this figure, BN refers to batch normalization [13], which has a very advantageous feature and has a great positive effect on the entire training process.

In-depth network training is a very complicated process. Some small changes in the previous layers of the network will be scaled up by the next layer of the network. Due to the existence of hidden defects, if the input data distribution of a layer changes, the layer needs to adapt to and learn new data distribution. If there is a change in the distribution of training data each time, the whole training process will be very slow. After the network is trained, the parameters of each training neuron will change, which will make the distribution of the input data at each layer change during the next training. The update of the previous training parameter leads to the change of the distribution of the input data at the next layer. This change in data distribution is called "Internal Covariate Shift." The BN layer can solve this problem, so that each training input data distribution at each layer is unchanged, thereby accelerating the network convergence speed.

According to experience, the BN layer is generally added before the activation function. As shown in the residual block diagram above, the BN layer is before the ReLU layer. Since ReLU outputs zero when the input is not greater than zero, the neuron dies, so BN is used here for normalization. The processing mean value cannot be set to zero, so that the input distribution of the current layer activation function remains unchanged, and each network training can be made more independent, which helps to accelerate the entire network. In addition, BN also has a regularization effect, which includes the input minus the average divided by the standard deviation of the calculation, which is equivalent to the introduction of noise, the larger the batch-size, the smaller the regularization effect, because the noise at this time is smaller. The next layer is the Dropout layer [14]. The effect of this method is to make the neurons inactivated according to a certain probability, that is, the output is zero. The main purpose is to reduce over-fitting of the network. The BN used above has a regularization effect to a certain degree, and thus has a certain effect of preventing overfitting. However, in order to ensure the fitting of the sleep staging algorithm, the Dropout layer is still added in the module. After the above preprocessing, the data is input to a convolution layer (convolution layer), and the detailed convolution layer parameters are set forth later in the overall network design. The above four-layer network is repeated again to form the main line network structure of the residual module.

The branch network part, that is, the shortcut connections between layers, refers to the idea of deep residual network, where the main purpose of the 1x1 convolutional layer is to modify the feature dimension and ensure consistency with the output dimension of the convolutional layer. The purpose of maxpooling is to reduce the size of each dimension of data. Finally, the sum of the convolutional output of the main line and the output of the branch is added to obtain the final output of the residual module.

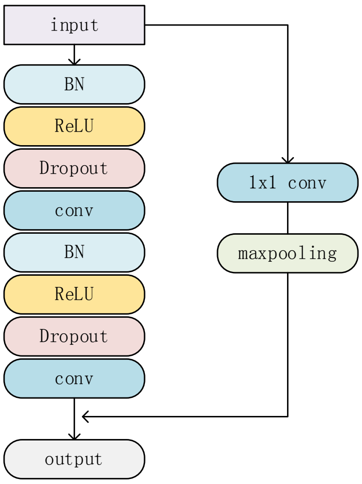

Figure 2. custom residual block

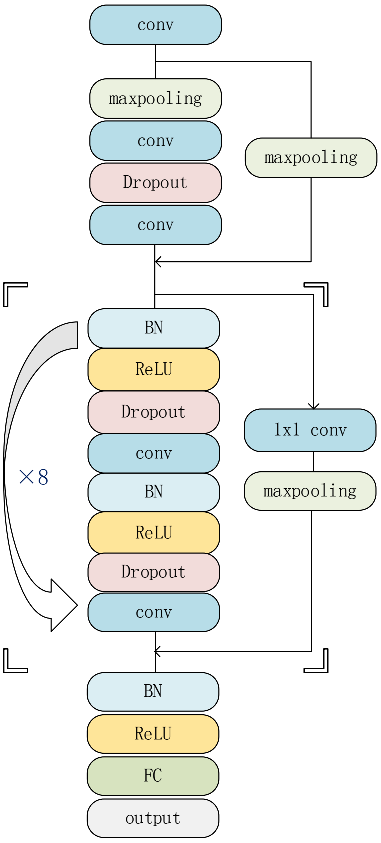

Figure 3. ResSleepNet model

3.2. Build ResSleepNet

The overall network model of this article is based on the residual module defined in *A*, hence the name ResSleepNet. The overall structure is shown in Figure 3. We enter data which size is (1, 3000), the first convolution kernel size is (1, 16), the dimension is 16, and the step size is 4, then the first layer output is (1, 750); Re-enter the main line through the second convolution layer, the core size (1, 8), the dimension is 16, step size 1, does not change the data size; after the Dropout layer, this layer in order to reduce the model over-fitting; Immediately after the input to the third convolution layer, the kernel size is (1, 8), the dimension is 16, the step size is 2, and the output data size is (1, 375); The size of the mainline pooling layer core is (1, 2), and the step size is 2. The output size and feature dimension are the same as those of the main line. After the two are added, the residual unit is input. There are a total of 8 residual units, where the final output data size and feature dimensions for each residual unit are specified. The first residual unit input is the output of the upper convolution, the size is (1, 375), the first residual unit output data size is unchanged (1, 375), and the feature dimension is 32; The second residual unit output data size is (1, 375) and the feature dimension is 32; The third residual unit output data size is (1, 188) and the feature dimension is 64; The fourth residual unit output data size is (1, 94) and the feature dimension is 64; The fifth residual unit has an output data size of (1, 94) and a feature dimension of 128; The sixth residual unit output data size is (1, 24), and the feature dimension is 128; The seventh residual unit output data size is (1, 24) and the feature dimension is 256; The eighth residual unit output data size is (1, 6), and the feature dimension is 256;

The bypass branch consists of a 1x1 convolutional layer and a maxpooling layer. The purpose is to change the feature dimension and size of the input data during a shortcut connection [22] to ensure that the output of the mainline residual block is added. Dimensions and dimensions are consistent. The residual module outputs a layer of BN and ReLU and then inputs it to the fully connected layer. This layer contains 1536 neurons. All inputs are tiled into a column of data. Finally, five sleep state staging features are output. Finally, they are input to the softmax layer. The obtained input sample belongs to each sleep stage probability, and the largest one is taken as the final stage result.

4. Training and result analysis

The experimental environment is Ubuntu 16.04, CUDA toolkit8.0 and CuDNN v6, Python3.5, TensorFlow-gpu1.40. The mini-batch size is 100 and the initial learning rate is 1e-4. The parameters of the training process change as shown in Figure 4, and the best epoch parameters are shown in Table 2 below. From the Table 2, we can see that the best epoch is the 23rd epoch of training. The time required for each epoch training is 42 seconds, the accuracy rate of the training set is 89.7%, and the verification set accuracy rate is 87.9%. The epoch overall verification set is shown in Table 3 below.

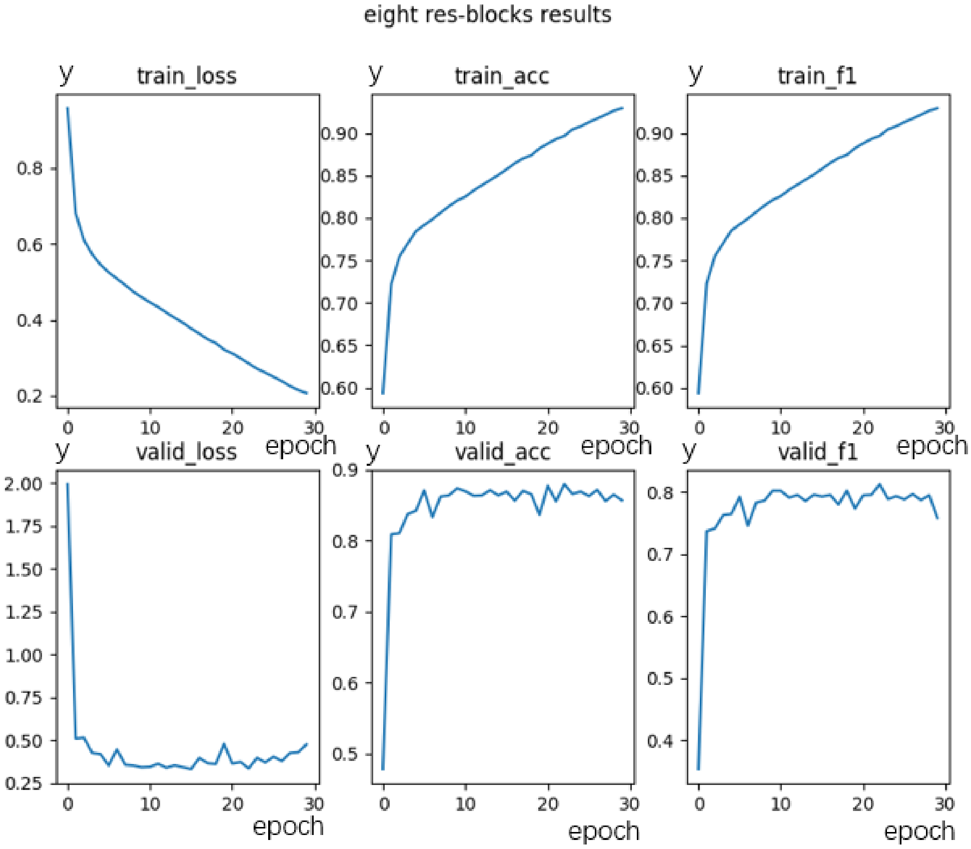

Figure 4. ResSleepNet training process

Table 2. epoch23 results

epoch	*time*	*train_acc*	*train_f1*	*val_acc*	*val_f1*
23	*42*	*0.897*	*0.897*	*0.879*	*0.813*

Table 3. epoch23 val results

Sleep	*Results*				
W	*295*	*12*	*0*	*0*	*4*
S1	*15*	*48*	*16*	*1*	*37*
S2	*1*	*12*	*553*	*34*	*15*
S3	*1*	*0*	*35*	*481*	*0*
REM	*5*	*26*	*15*	*0*	*294*

*We selected the netw*ork in [15] as the comparison object. The network used in this paper is DeepSleepNet. The data sets used in the experiment are the same and all are the public dataset of Sleep-

EDF expand. Same as the experiment in this paper, set the mini-batch size to 100. The training set is the same as the verification set extraction method and the data enhancement method. The training includes two steps, pre-training and fine-tune training, each training 100 epochs. The fine-tune training process is shown in Figure 5 and the final training results are shown in Table 4. It can be seen from the training results in the above table that the accuracy of the DeepSleepNet test set is higher than that of the traditional manual extraction feature, but there is obvious overfitting. The accuracy of the final training set is 98.7%, and the accuracy of the test set is 87.7%, and a total of 200 epochs were trained.

Comparing the two kinds of network training conditions, the accuracy of the DeepSleepNet training set is very high, reaching 98.7%, but the verification set accuracy rate is 87.7%. Theoretically, there is an overfitting phenomenon. The accuracy of the ResSlideNet training set is lower than that of the former by 89.7%, but the accuracy of the validation set slightly exceeds the former by 87.9%, and there is no overfitting according to the accuracy of the training set and the accuracy rate of the verification set, and the network is more generalized. In addition, DeepSleepNet pre-training and fine-tune training trained a total of 200 epochs.

Table 4. DeepSleepNet result

epochs	time	train_acc	train_f1	val_acc	val_f1
199	387	0.987	0.983	0.877	0.805

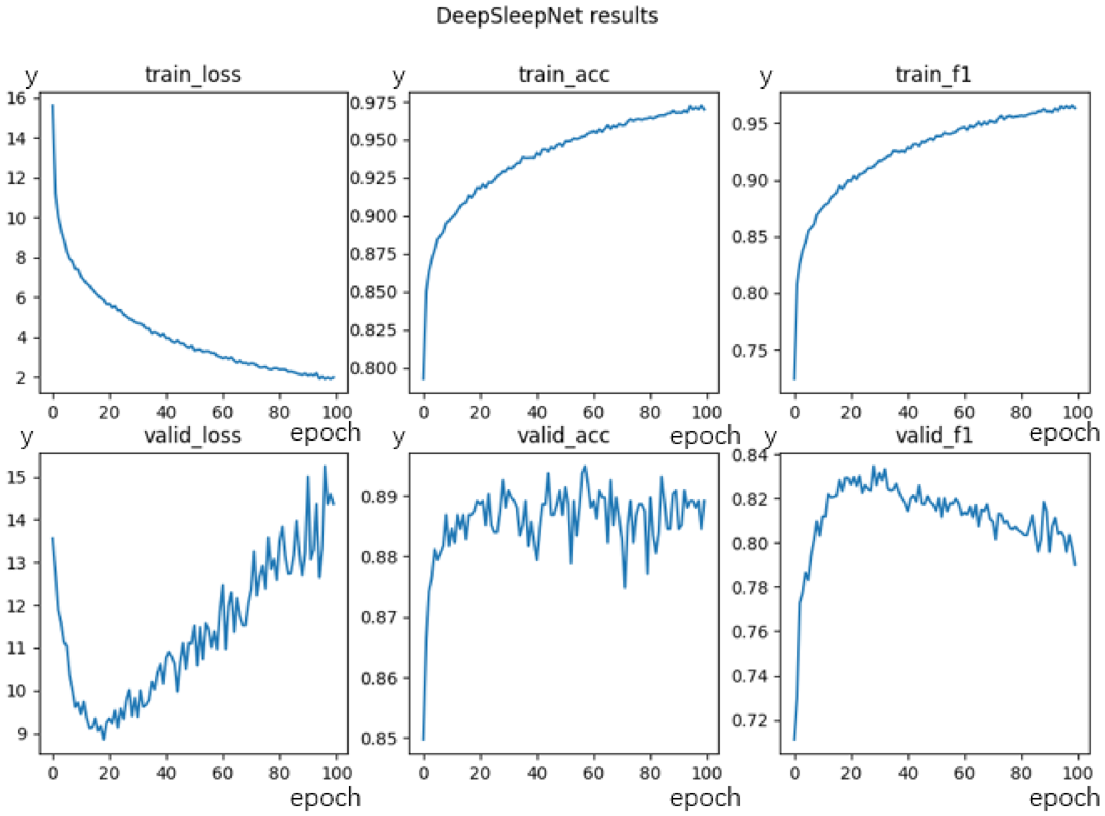

Figure 5.DeepSleepNet fine-tune training process

5. Conclusion

This paper proposes a deep learning network model for autonomous learning time series data and applies it to the field of automatic sleep staging. It extracts high-dimensional features from single-channel EEG signals (Fpz-Cz) and outputs the corresponding sleep staging results. Compared with the traditional method of extracting the fixed attribute feature from the multi-lead EEG signal, the manpower and material cost are saved. The accuracy of the validation set and the overall model's generalization ability have been improved.

6. Acknowledgment

The work was supported by the Natural Science Foundation of Hubei Province (No. 2014CFB869) and the Fundamental Research Funds for the Central Universities (No. 2014-IV-136).

References

[1] Leon, Cohen. "Time-frequency analysis: theory and applications." USA: Pnentice Hall (1995).
[2] Pincus, Steve. "Approximate entropy (ApEn) as a complexity measure." Chaos: An Interdisciplinary Journal of Nonlinear Science 5.1 (1995): 110-117.
[3] Maquet, Pierre. "The role of sleep in learning and memory." science 294.5544 (2001): 1048-1052.
[4] Xie, Lulu, et al. "Sleep drives metabolite clearance from the adult brain." science 342.6156 (2013): 373-377.
[5] A. Rechtschaffen and A. Kales, Eds., A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. Washington D.C.: Public Health Service, U.S. Government Printing Office, 1968.
[6] Walters, Arthur S., et al. "The scoring of movements in sleep." Journal of Clinical Sleep Medicine 3.02 (2007): 155-167.
[7] PhysioNet, “The Sleep-EDF Database.” https://www.physionet.org/physiobank/database/sleep-edf/. [Online;accessed 14-Nov-2016].
[8] Imtiaz, Syed Anas, and Esther Rodriguez-Villegas. "Automatic sleep staging using state machine-controlled decision trees." Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE. IEEE, 2015.
[9] Zhu, Guohun, Yan Li, and Peng Paul Wen. "Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal." IEEE journal of biomedical and health informatics 18.6 (2014): 1813-1821.
[10] Hassan, Ahnaf Rashik, and Mohammed Imamul Hassan Bhuiyan. "Computer-aided sleep staging using complete ensemble empirical mode decomposition with adaptive noise and bootstrap aggregating." Biomedical Signal Processing and Control 24 (2016): 1-10.
[11] da Silveira, Thiago LT, Alice J. Kozakevicius, and Cesar R. Rodrigues. "Single-channel EEG sleep stage classification based on a streamlined set of statistical features in wavelet domain." Medical & biological engineering & computing 55.2 (2017): 343-352.
[12] He, Kaiming, et al. "Identity mappings in deep residual networks." European Conference on Computer Vision. Springer, Cham, 2016.
[13] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).
[14] Wahlbeck, Kristian, et al. "Dropout rates in randomised antipsychotic drug trials." Psychopharmacology 155.3 (2001): 230-233.
[15] Supratak, Akara, et al. "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG." IEEE Transactions on Neural Systems and Rehabilitation Engineering 25.11 (2017): 1998-2008.