Date of publication June 20, 2019, date of current version June 20, 2019.

Digital Object Identifier

Methods to Overcome the Real-World Deep Learning Application Problems

SANGHEON LEE[1], (Student Member, IEEE), YUSUNG KIM[2], (Member, IEEE), and JEE-HYONG LEE[3], (Fellow, IEEE)
[1]Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 16419, Korea (e-mail: lawlee1@skku.edu)
[2]Memory Business, Samsung Electronics, Hwaseong 18448, Korea (e-mail: yusunge.kim@samsung.com)
[3]Department of Software, Sungkyunkwan University, Suwon 16419, Korea (e-mail: john@skku.edu)

Corresponding author: Jee-Hyong Lee (e-mail: john@skku.edu).

ABSTRACT In recent years, the development of machine learning, especially deep learning, and its applications have been actively studied. Although deep learning is applied in many areas, there are still problems to be solved in order to introduce deep learning models into real-world systems. In this paper, we summarized the papers that overcome various problems in the real-world application of the deep learning model. For the image classification model, an adversarial attack has been proposed which causes malfunction of the model by mixing noise that is hardly distinguishable by human into the input image of the model. As a result, the reliability of the deep learning model has been raised, and a study has been proposed to prevent the adversarial attack. Lee *et al.* proposed an adversarial detection method based on gaussian process regression using intermediate features extracted from the classification model. The proposed detector showed higher detection performance than other detection methods when extremely few adversarial examples are used in training. Lee *et al.* proposed a deep learning model that recognizes string CAPTCHAs that widely used in internet sites. The proposed method eliminates the noise in the CAPTCHA image through image processing, separates string image into single character images, and recognizes CAPTCHA characters by training CNN model. As a result of experiment on CAPTCHA used in Korea ticket reservation site, the proposed model showed a high recognition rate of 85% based on CAPTCHA. Won *et al.* proposed a deep learning model that predicts movie audience demand before opening by using a nonlinear regression model. In addition to features provided by the KOFIC, which provides cinema information, they define features that can be used for prediction and proposed a Bi-LSTM deep learning model to perform sentimental analysis on movie reviews. Experimental results showed that the proposed method had higher performance than other sentimental analysis methods and effectively predicted the demand of movie audience. Won *et al.* proposed a technique for effectively handling the OOV words that are not in the existing vocabulary during word embedding. The proposed method is a Bi-LSTM structured deep learning model named Context-Char, which embeds the OOV words using contextual and morphosyntactic information. As a result of experiments using datasets containing the OOV words, the proposed method showed higher performance than other methods of handling OOV words. Lim *et al.* proposed an effective under-sampling technique for processing imbalanced data. They proposed deep representation models that extract the structural features of major class data and minor class data and calculated the degree of conformity of the data to determine under-sampling. In the experiments using various imbalanced data, the proposed method showed higher performance than other processing methods for most datasets.

INDEX TERMS Deep Learning, Real-World Application, Adversarial Detection, CAPTCHA, Sentimental Analysis, Out-Of-Vocabulary Words, Imbalanced Data

I. DETECT ADVERSARIAL EXAMPLE USING GAUSSIAN PROCESS-BASED DETECTOR

A. PAPER MAIN THEME

Adversarial attack is a technique that causes a malfunction of the deep learning model by mixing noise that cannot be distinguished by the human into the input of the model. Figure 1 at right shows an image in which the adversarial attack is applied to the image at left recognized as "panda" by the image classification model. The two images are not distinguished by the human eye, but the image classification model recognizes the right image as "gibbon", rather than "panda". If a deep learning model is applied to a major part of the system, the adversarial attack can lead to serious problems in system security.

Several techniques have been proposed to prevent the adversarial attack, but few defense methods that can effectively defend against various and powerful attacks have yet been proposed. Therefore, adversarial detection has been proposed instead of an adversarial defense, which determine whether the input of a model is an adversarial example or not. Many of the adversarial detection methods showed high performance against most attacks. However, many of the detection methods already proposed are the deep neural network-structured model, which requires a large number of adversarial examples to train.

B. PROPOSED IDEA

Lee *et al.* proposed an efficient detection method for adversarial example, which is shown in figure 2 [1]. The proposed method consists of two steps. First, the intermediate feature values generated by the classification model are extracted for a given image. The intermediate feature is the output of the last hidden layer of the classification model, indicating the classification probability for a given input. The classification probability of most adversarial examples is characterized by the fact that the two classes have similar values to each other, so the detector grasps these patterns and performs adversarial detection.

Second, the intermediate feature information is used to determine whether the image is an adversarial example by applying Gaussian process regression. Gaussian process regression is a method to infer the mean and variance of the whole data range based on the observed data, by defining the relationship between the data using the characteristics of the data. Assuming $f(x) \sim N(0, K(\theta, x, x'))$ for the function $f(x)$ of x, the log marginal likelihood is as follows:

$$\begin{aligned}\log p(f(x)|\theta, x) = &-\frac{1}{2} f(x)^T K(\theta, x, x')^{-1} f(x) \\ &-\frac{1}{2}\log\det(K(\theta, x, x')) \\ &-\frac{|x|}{2}\log 2\pi\end{aligned} \quad (1)$$

where, $K(\theta, x, x')$ is a covariance matrix for all possible observed data pairs (x, x'), calculated from a pre-defined kernel function, and θ is a hyperparameter of the covariance function. Based on the θ that maximizes this marginal likelihood, the distribution of the function value $f(x^*)$ for the unobserved data x^* is $p(y^*|x^*, f(x), x) = N(y^*|A, B)$. That is, the posterior distribution has mean function A and variance function B, where A and B are calculated through the following equations:

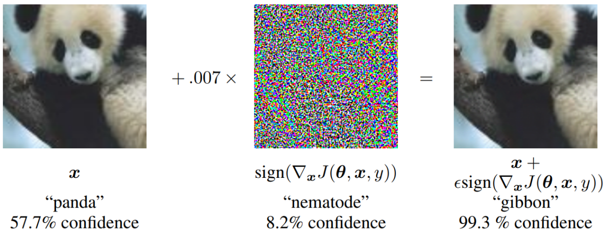
+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

FIGURE 1. An adversarial image that has imperceptible perturbation added to the natural image.

$$A = K(\theta, x^*, x) K(\theta, x, x')^{-1} f(x) \quad (2)$$

$$\begin{aligned}B = &K(\theta, x^*, x^*) \\ &- K(\theta, x^*, x) K(\theta, x, x')^{-1} K(\theta, x^*, x)^T\end{aligned} \quad (3)$$

where, $K(\theta, x^*, x)$ denotes the covariance values between all observed data x and the new data x^* based on the hyperparameter value θ, and $K(\theta, x^*, x^*)$ is the variance value at x^*.

In the Gaussian process regression, the influence between two data is defined as covariance. If the dimension of data is high, it is difficult to grasp the pattern of covariance between data. Therefore, inputting low-dimensional high-level features extracted through convolution and pooling layers, rather than a high-dimensional raw image, might perform the Gaussian process regression more efficiently.

In addition, the output function $f(x)$ for the input x is non-differentiable, because the Gaussian process regression trains the probabilistic distribution of the output using the observed data. Therefore, the secondary adversarial attack on Gaussian process regression does not work.

C. RESULTS AND DISCUSSIONS

To verify the performance of the Gaussian process regression-based detector, the datasets used in the experiments are MNIST and CIFAR10. Image classification models for MNIST and CIFAR10 are deep convolutional neural network-structured models and the attack methods used in the experiments are FGSM, BIM, JSMA, DeepFool, and C&W attacks. For the Gaussian process regression-based detector, 300 natural images and 300 adversarial examples are used for training. The baseline model compared with proposed detector is the deep convolutional neural network-structured binary classification model, and the training data of the baseline model is set to 300 natural images and 300 adversarial examples for the same experimental conditions.

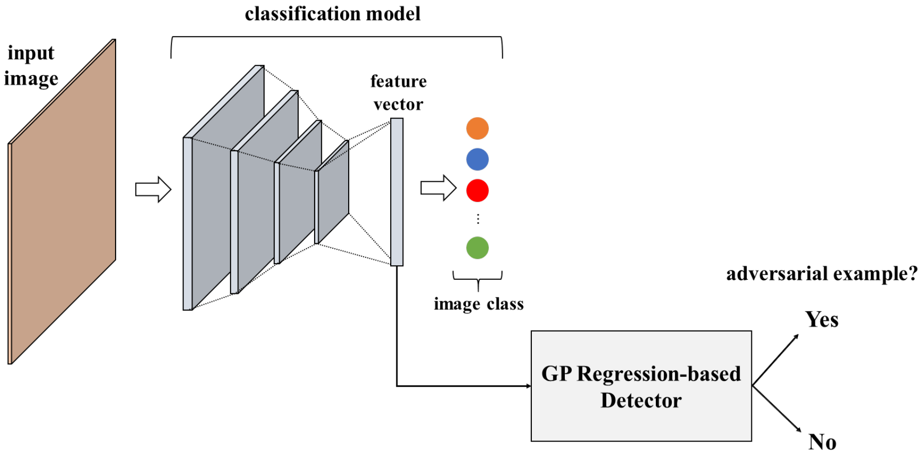

FIGURE 2. Gaussian process regression-based adversarial image detector.

TABLE 1. Detection accuracy for the MNIST dataset.

	FGSM(%)	BIM(%)	JSMA(%)	DeepFool(%)	C&W(%)
Baseline	**99.61**	**99.27**	82.99	66.55	61.58
GP-based	92.86	69.3	**97.94**	**99.64**	**99.67**

TABLE 2. Detection accuracy for the CIFAR10 dataset.

	FGSM(%)	BIM(%)	JSMA(%)	DeepFool(%)	C&W(%)
Baseline	62.22	50.13	**95.81**	50.00	50.01
GP-based	**76.92**	**50.42**	94.86	**97.94**	**97.93**

Table 1 and 2 show the adversarial detection accuracy for the MNIST and CIFAR10 datasets. Experimental results show that the Gaussian process regression-based detector shows better detection performance than the baseline detection model, except for some attacks. Since the detection accuracies of the baseline model for the DeepFool and C&W attacks are quite low, it can be observed that the baseline model, which is a deep neural network, cannot train at all with just a few training images. Also, for the C&W attack, which is considered the most powerful attack, the proposed method shows higher detection accuracy than the baseline model in both MNIST and CIFAR10 datasets.

II. STRING CAPTCHA ATTACK USING CNN

A. PAPER MAIN THEME

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a system that judges whether a service person is a person or not on the Internet. CAPTCHA is used to prevent continued sign-in attempts and memberships via bypass on Internet sites. There are many types of CAPTCHAs, depending on their type, and string CAPTCHA or image CAPTCHA are widely used. The string CAPTCHA blends noise into an image containing a string of alphanumeric characters, thereby preventing non-human objects from reading the meaning of the string.

CNN (Convolutional Neural Network) is a multi-layer neural network that is used to classify and recognize data such as images and texts. CNN is divided into the image preprocessing step and the classification step. The preprocessing step consists of a convolution layer that extracting features of the input data and a pooling layer that extracting the most critical parts of the features extracted through convolution. In the classification step, images are classified and recognized through fully-connected neural networks using preprocessed data.

Lee *et al.* proposed a deep learning model to recognize the string-based CAPTCHA used in the Korea ticket reservation site "Interpark Ticket" [2]. The proposed method first removes noise from the CAPTCHA and separates the character string into single characters through image processing. Second, the CNN model is trained by using the separated single character data for classify the CAPTCHA string.

B. PROPOSED IDEA

1) Analysis of the Target CAPTCHA

Korean ticket reservation site "Interpark Ticket" applies the "safe ticket booking" system, which uses the string CAPTCHA in the reservation process in order to judge whether the person who uses the service is a person. Figure 3 shows the string CAPTCHA used in the "safe ticket booking" system.

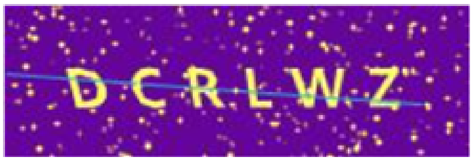

FIGURE 3. String CAPTCHA used in the Interpark Ticket.

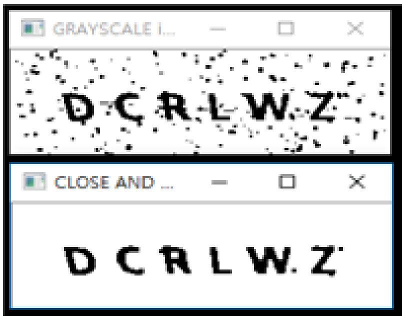

FIGURE 4. Eliminate the noise in the image with open and close operations.

The characteristics of the string CAPTCHA used in the Interpark ticket are as follows.

- The string consists of 18 uppercase alphabetic characters A, B, C, D, E, K, L, M, N, P, Q, R, S, T, U, to be.
- The colors of characters in a CAPTCHA are the same, followed by a solid background.
- There are small dots around the string, and there is a straight line in the middle of the string.
- The size of the CAPTCHA image is fixed, and the background color and the color of the string are reused.

2) Image Processing

Image processing is the process of extracting single character images from a CAPTCHA image file. Image processing is divided into two steps: removing noise in the CAPTCHA image; and separating the noise-removed string image into single character images.

The step of removing the noise in the CAPTCHA image is implemented by the morphology operation. Morphology operation is an image processing technique that transforms the shape of the object by reducing or enlarging the light or dark areas of the image. The open operation removes the minute pieces appearing in the area, and the close operation covers the fine gaps in the area. Therefore, it is possible to remove noise in the image by sequentially performing the closing operation and the opening operation. Figure 4 shows a noise-free string image with a pixel value of 0 (black) or 255 (white).

In the step of separating a string into a single character image, they used the $findContours()$ function to extract the contours of the object in the image. A single character image was obtained by extracting the minimum rectangle boundary that containing each word through the $findContours()$ function, and cutting it along the boundary line. Single character images obtained through the image processing were saved in 32 * 32 size grayscale and JPG formats.

TABLE 3. CAPTCHA image recognition rate.

	Based on CAPTCHA	Based on single character
Proposed Method	**85.06%**	**97.20%**

3) Image Classification

For the training of the classification model, 27000 of single character image data are used. For verifying the performance of the classification model, 10800 single character image data obtained by applying the image processing process to 1800 CAPTCHA images not included in the training dataset were used. The classification model is based on Inception V3 model developed by Google.

C. RESULTS AND DISCUSSIONS

Table 3 shows the CAPTCHA image recognition experiment results. As a result of the CAPTCHA recognition experiment, 1531 CAPTCHAs among 1800 different CAPTCHAs were successfully recognized. In addition, 10498 data of 10800 data were successfully recognized based on a single character image. Recognition of CAPTCHA criteria is considered to be successful only if all the characters in the CAPTCHA string are recognized successfully. As the recognition rate based on a single character image is very high, it can be seen that the feature extraction for each character separated by image processing is successful.

This paper demonstrated the vulnerability by implementing a model that recognizes a string-based CAPTCHA that is currently being used in a particular ticketing site with a high probability. Authors proposed an image processing process to remove noise in CAPTCHA and separate it into single characters. Recognition experiment showed high recognition rate of 85.06% based on CAPTCHA and 97.20% based on single character. The high recognition rate of a single character means that it is difficult to lower the CAPTCHA recognition rate of the proposed model simply by increasing the length of the character string in the CAPTCHA. Therefore, it is recommended that the site be introduced with CAPTCHA supplemented by new algorithms such as adding new distortions to characters or adjusting the distance between characters.

III. BOX OFFICE FORECASTING USING NON-LINEAR REGRESSION

A. PAPER MAIN THEME

The film satisfies the individual demand through a single watch and therefore, movies have a shorter life cycle as a product and their success is determined in a shorter time than other products. As a result, producers, investment companies, and distributors are making a lot of efforts to predict movie performances for distribution and screening decisions before the release of movies. This is because the release period of

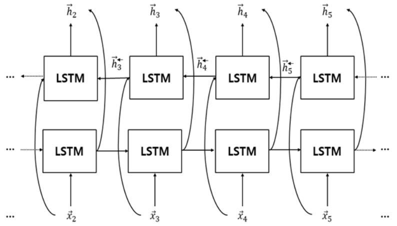

FIGURE 5. Classification using Bi-LSTM.

TABLE 4. New features for modeling.

Variable Name	Description	Value
Series	Series effect	0 / 1
Holiday	Holiday effect	0 / 1
Expect	Expectation index	Real Value
Positive	Positivity index	0 ∼1
Comp(2)	Competition effect	Real value

the movie is short and it is necessary to respond promptly to the reaction of the movie viewers to earn more than the investment cost. In the meantime, there have been various studies to predict the number of cumulative audiences, but most of these models predict future demand using initial data immediately after opening.

Won *et al.* proposed a method to predict the audience demand of the pre-release film using the historical data of the past released movies [3]. Past researches used only explanatory variables such as supervisor, actor, distributor, and genre, but did not take into account the oral effect of SNS. Authors, on the contrary, proposed explanatory variables like expectation index provided by portal site Naver, sentimental analysis and competition effect. They used Genetic algorithm to select optimal explanatory variables and random forest to predict.

B. PROPOSED IDEA

1) Bidirectional LSTM

In order to know the context in a sentence, consideration should be given not only to previous information but also to future information. With Bidirectional Long Short Term Memory (Bi-LSTM) shown in figure 5, both previous and future information can be stored. In this paper, authors used Bi-LSTM to classify positive and negative movie reviews.

2) Explanatory Variable Definition

Authors have collected movie data, which has cumulative audience over 40,000 for the past four years from the movie admission ticket integrated network(KOFIC). Authors defined the holiday period and the series as a binary number. In order to consider the effect of competition, they defined the number of competing movies as a variable before and after opening. Also, they used the positive index and pre-release review of the movie provided by the portal site "Naver" to consider the oral effect. Table 4 shows the new features that authors defined for modeling.

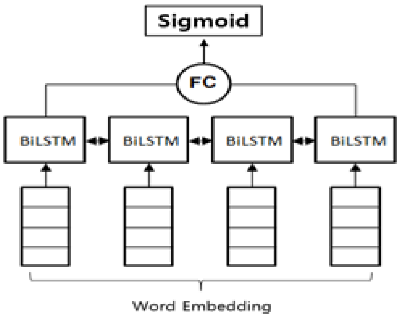

FIGURE 6. Classification using Bi-LSTM.

TABLE 5. Evaluation of Sentimental Classification

Model	Accuracy(%)
Doc2Vec Logistic Regression	78.24
Term-existance Naive Bayes	80.41
Bi-LSTM Logistic Regression	**84.53**

3) Positive Index

They defined positive index with a value between 0 and 1 as a percentage of positive reviews for the entire review. In order to discriminate between positivity and negativity, the following binary classifiers are proposed using Bi-LSTM after word embedding for the correct data set provided by Eunjung Park. Figure 6 shows the structure of the classification model for sentimental analysis.

C. RESULTS AND DISCUSSIONS

Table 5 shows the classification accuracy of the sentimental analysis task. The performance of the binary classifier using Bi-LSTM is superior to the performance of the model which Eunjung Park has proposed.

In order to predict the number of movie audiences effectively, authors defined the parameters for the oral effect, competition effect, and holiday effect in addition to the basic variables given by KOFIC. Using the genetic algorithm, the optimal parameters for the model were selected. In all three models of comparative analysis, variables related to competitive effects were not influential. The future works for better model that authors mentioned in the paper are as follows. First, newly define the variables related to the competition effect is needed. Second, consider holiday period rather than simply the binary for holiday effect is needed.

IV. EMBEDDING FOR OUT OF VOCABULARY WORDS CONSIDERING CONTEXTUAL AND MORPHOSYNTACTIC INFORMATION

A. PAPER MAIN THEME

Word2vec is a representative language model that are used to produce word embeddings by taking as its input a large corpus of texts. Word vectors created by the model are positioned in the low dimensional latent space, and such words, that share common contexts are located close to each other. For that reason, word embeddings are useful for

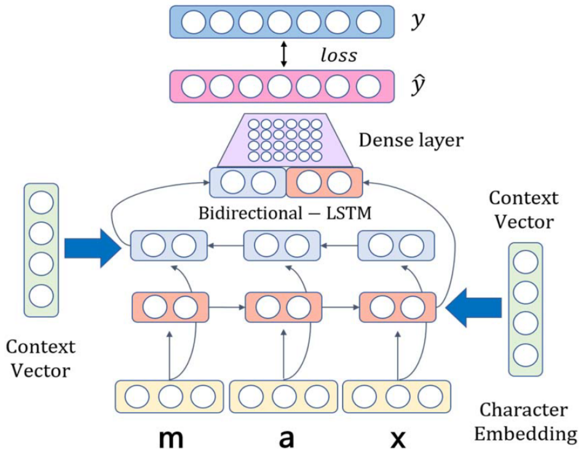

FIGURE 7. Context-Char model architecture.

performing NLP tasks such as text classification, translation, and summarization.

However, even though Word2vec is trained on a large corpus with large amounts of vocabulary, it is not possible to capture the entire vocabulary that exist in the real-world. The words that are not included in the vocabulary are called OOV (Out of Vocabulary) words, and the usefulness of word embeddings is limited by OOV words.

Therefore, in several NLP tasks, handling uncertainties effectively on OOV words is an important issue to be solved. A typical way to give word embedding for an unseen word is to assign UNK token to rare word and learn a distributed representation of the token. However, this method is not reasonable, because it gives the same word embeddings for different kinds of OOV words. Another way to deal with unseen word is to assign a random vector that follows the Gaussian distribution. This is also unreasonable in representing words because it gives random values to each word.

Won *et al.* proposed a method to handle OOV words by considering both contextual and morphosyntactic information of words [4]. This is done by providing the word's average context vector as the initial state of the character-based Bi-LSTM, which called "Context-Char".

B. PROPOSED IDEA

1) Model Architecture

Authors approached the problem of OOV words by generating word embeddings, that consider both spellings of the target word and the surrounded context words. Figure 7 denotes the whole architecture of the Context-Char model.

For a given corpus C and words $\{w_k\}_{k=1}^{V_1} \in C$, where V_1 is the size of the corpus vocabulary, model trained to find the function $f : w_k \to \mathbb{R}^D$ such that the output of the function approximates the embedding vector of the target word, e_{wk}. Context vector of the Bi-LSTM model is defined as (4), where α is the window size for the Context-Char model.

TABLE 6. Classification accuracy and loss of OOV words handling methods

Method	AG's news		Yelp review	
	Loss	Accuracy(%)	Loss	Accuracy(%)
Random	0.4025	85.42	1.2106	48.16
UNK	0.3137	89.14	1.0724	53.84
Context	0.3191	88.71	1.0810	53.43
MIMICK-RNN	0.3120	89.07	1.0620	54.39
Context-Char	**0.3049**	**89.65**	**1.0426**	**55.08**

$$c_k = \frac{1}{2\alpha}\sum_{j=1}^{2\alpha} e_{w_j} \tag{4}$$

Forward-LSTM and backward-LSTM takes both character embedding sequence and context vector as input and compute hidden state cells of each sequence. The word vector generated by the proposed model is computed by feeding concatenation of two last hidden state vectors of forward and backward LSTM into fully connected layer as (5).

$$fw_k = g(W[h_n^{\rightarrow}, h_n^{\leftarrow}] + b_h) + b_l \tag{5}$$

For the loss function of the model, they used the mean squared error between predicted values and target values as follows:

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(f(w_i) - e_{w_i})^2 \tag{6}$$

C. RESULTS AND DISCUSSIONS

In the experiments, they used AG's news topic with 4 classes classification dataset and Yelp review with 5 classes classification dataset. For the reason that the OOV rates were low for two datasets, they randomly chose words from each sentence to shuffle the order of middle letters, to artificially create OOV words. The total number of training samples on AG's news dataset is 120,000 and testing 7,600, and the OOV rate of AG's news dataset after artificially creating OOV words by shuffling the order of letters is 16.26%. The total number of training samples on Yelp review dataset is 100,000 and testing 25,000, and the OOV rate is 16.14%.

Table 6 shows the performance of each model for both tasks. The model used for training is single Bi-LSTM layer with 128 hidden units. 'Random' is assigning random vector that follows Gaussian distribution to OOV words; 'UNK' is assigning UNK token vector from pre-trained word embeddings; 'Context' is using local average context vector as OOV; MIMICK-RNN is generating embeddings by having word letter sequence as an input; and Context-Char is the proposed method. For each task, Context-Char showed the highest accuracy and lowest loss value among other methods.

Figure 8 shows the test loss of each epoch for the Yelp review dataset among five OOV words handling methods. Due to assigning random embeddings to OOV words, so that having different values between training and test cases, 'Random' method had performed the worst. In contrast,

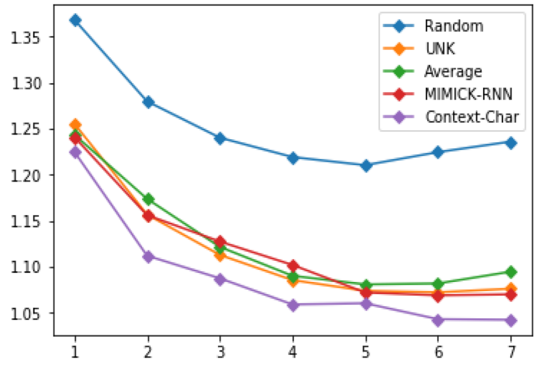

FIGURE 8. Test loss of Yelp review dataset.

'Context-Char' method converged faster to the low test loss point and had the lowest test loss value among five methods.

In this paper, authors proposed the Context-Char, a context reinforced morphosyntactic method to extract desired information from OOV words. By relaxing the misrepresentation with OOV words, proposed method improves the quality of classifying text with unseen words. For improving the model much more, authors said that it is needed to be trained on a large corpus which has diverse contexts and words. By sampling the words, based on their frequency, the model can be prevented from overfitting to certain words.

V. DEEP REPRESENTATION MODEL-BASED UNDER-SAMPLING METHOD FOR IMBALANCED DATA

A. PAPER MAIN THEME

The utility of future forecasting through data classification is increasing in various areas such as politics, society, and economic culture. However, data collected for future forecasts often show a high degree of imbalance, which occurs at a very low rate in the target situation. If such an imbalance is intensified, the complexity of the data increases, which degrades the classification performance.

Most of existing machine learning techniques assume that the ratio of minor class to major class is similar. Therefore, when training the imbalanced data, classifier based on machine learning deflects to major class data with a relatively high ratio, and classification performance deteriorates. In addition, existing machine learning researches that take the imbalanced environment into consideration also use a method that simply mitigates the imbalance. Therefore, it is vulnerable to structural problems such as overlapping in which the distribution of data belonging to different classes in an imbalanced environment overlaps and the distinction between classes becomes ambiguous.

Lim *et al.* proposed an under-sampling technique that considers the structural characteristics of data in an imbalanced environment [5].

B. PROPOSED IDEA

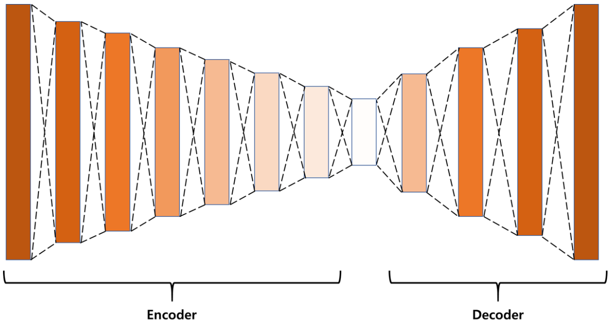

FIGURE 9. Auto-encoder structured deep representation model.

TABLE 7. Comparison of classification accuracy for imbalanced data

dataset (ratio)		SVM	Cluster Centroid	SMOTE	Proposed Method
ecoli3	8.60	0.00	0.50	0.61	**0.64**
ecoli4	15.80	0.00	0.44	**0.82**	0.77
ecoli-0-1 vs 5	11.00	0.00	0.69	0.80	**0.87**
glass1	1.87	**0.59**	0.57	0.57	0.57
glass-0-6 vs 5	11.00	0.33	0.47	**0.87**	0.42
haberman	2.78	0.03	0.41	0.37	**0.46**
wiscosin	1.86	0.65	0.96	0.95	**0.96**
yeast3	8.10	0.00	0.66	0.71	**0.71**
yeast4	28.10	0.00	0.31	0.29	**0.44**
yeast5	32.73	0.00	0.39	0.50	**0.54**
zoo-3	19.20	0.00	0.19	-	**0.5**

1) Deep Representation Model

To address the structural problems of imbalanced data, authors use a deep representation model that train the attributes of the data. In order to grasp the structural characteristics of each class, authors constructed a model M that trained using only the major class data and a model m that trained using only the minor class data. Each model is an auto-encoder structured neural network as shown in figure 9. Models grasp the distribution of data of the classes used in training, and combine them to calculate the structural degree of conformity of each class. The degree of conformity is determined by the reconstruction error E of the deep representation model. E is expressed as (7), x is input data, and f is M or m model.

$$E(x) = \sqrt{(x - f(x)^2} \tag{7}$$

To compute the conformance of the minor class model of the major class instance x, authors computed E for the minor class model. Major class data must have a conformity value greater than the threshold, which is calculated through the deep representation model trained using the minor class data. If the conformity value is smaller than the threshold, the instance has the characteristic of the minor class.

2) Under-sampling

In order to apply under-sampling to data, it is necessary to define thresholds that determine compliance. The threshold value T for determining the conformance/non-conformity is determined by searching for a value satisfying (8).

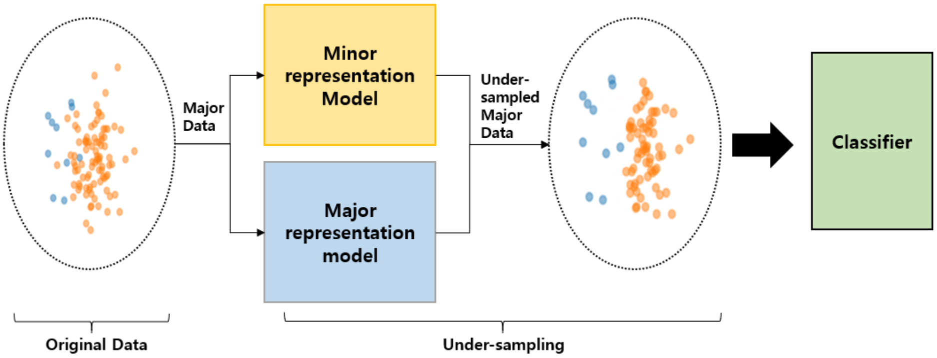

FIGURE 10. Proposed method that perform under-sampling for imbalanced data.

$$n(mD) \leqq n(U) < \frac{3}{4}n(MD) \quad \text{where } U = \{x \in MD | E(x) > T\} \tag{8}$$

mD is a minor class data, MD is a major class data, and U is a set of major class data whose major E is larger than a threshold value. Figure 10 shows the process of the proposed method that perform under-sampling for imbalanced data.

C. RESULTS AND DISCUSSIONS

To evaluate the effectiveness of the proposed method, authors evaluated how the classification performance of the data obtained by applying it to the imbalanced data is improved. The evaluation index for performance comparison is the f1-score average value of minor class data obtained through 4-fold cross-validation.

Table 7 shows the classification performance for baseline model that used only the classifier, cluster centroid, SMOTE, and proposed method. The proposed method shows the best performance in most experimental data without any bias on the number of data and imbalance ratio. As a result, it is shown that solving the structural problem of imbalanced data through undersampling considering the structural characteristics of data is more effective in improving classification performance than simply mitigating imbalance.

VI. CONCLUSIONS

In this paper, we have summarized the papers that overcome various problems related to the application of the deep learning model in the real-world systems. The Gaussian process-based adversarial detection method proposed by Lee *et al.* can effectively prevent adversarial attack, thereby increasing the reliability of the deep learning model and further enhancing the security of the model. The CAPTCHA recognition model proposed by Lee *et al.* proved to be vulnerable to CAPTCHA that is used for security of the system, and recommended that the site be secured. The movie viewer demand forecasting model proposed by Won *et al.* can be seen as a practical example of the deep learning model applications. The embedding technique of OOV words studied by Won *et al.* is one of the hottest subjects in the current NLP (Natural Language Processing) field, and it is an indispensable study in the future application of the deep learning model to the NLP field. The imbalanced data studied by Lim *et al.* is the essential consideration in the real-world application of the deep learning model. In addition, Lim *et al.* proposed an under-sampling method that can effectively handle imbalanced data, thereby increasing the applicability of the deep learning model in the real-world systems.

In addition to problems discussed in this paper, there are many problems that must be addressed in order for the deep learning model to be fully applied to real-world systems. We believe that solving these problems is as important as developing a new deep learning model with better performance.

REFERENCES

[1] S. Lee, N.-r. Kim, and J.-h. Lee, "Detect adversarial example using gaussian process-based detector," in Proceedings of the 10th International Conference on Internet (ICONI) 2018, ser. ICONI 2018, 2018.

[2] S. Lee, S. Woh, and J.-h. Lee, "String captcha attack using cnn," in Proceedings of KIIS Spring Conference 2018, ser. KIIS 2018. KIIS, 2018, pp. 69–71.

[3] M. Won, K. Kim, and J.-h. Lee, "Box office forecasting using non-linear regression," in Proceedings of KIIS Spring Conference 2018, ser. KIIS 2018. KIIS, 2018, pp. 76–78.

[4] M. Won and J.-h. Lee, "Embedding for out of vocabulary words considering contextual and morphosyntactic information," in Proceedings of the 2018 International Conference on Fuzzy Theory and Its Applications, ser. iFUZZY 2018, 2018, pp. 212–215.

[5] Y. Lim and J.-h. Lee, "Deep representation model-based under-sampling method for imbalanced data," in Proceedings of KIIS Spring Conference 2018, ser. KIIS 2018. KIIS, 2018, pp. 123–124.

SANGHEON LEE received the B.S. degree in Computer Engineering in 2018 from Sungkyunkwan University, Suwon, Korea. He is currently master course student in Department of Electrical and Computer Engineering at Sungkyunkwan University, Suwon, Korea. His current research interests include deep-learning, adversarial defense, image classification and natural language processing. He is a student member of the IEEE.

JEE-HYONG LEE received his B.S., M.S., and Ph.D. in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 1993, 1995, and 1999, respectively. From 2000 to 2002, he was an international fellow at SRI International, USA. He is currently working as a professor in the Department of Software at the Sungkyunkwan University, Suwon, Korea. His research interests include fuzzy theory and application, intelligent systems, and machine learning. He is a fellow member of the IEEE.