IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING: SECURITY AND FORENSICS
RESEARCH ADVANCES AND CHALLENGES

Received June 5, 2019, accepted August 28, 2019, date of publication September 13, 2019, date of current version October 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941376

A Backdoor Attack Against LSTM-Based Text Classification Systems

JIAZHU DAI, CHUANSHUAI CHEN, AND YUFENG LI
School of Computer Engineering and Technology, Shanghai University, Shanghai 200444, China

Corresponding author: Jiazhu Dai (daijz@shu.edu.cn)

This work was supported in part by the Research and Development Program in Key Areas of Guangdong Province under Grant 2018B010113001, and in part by the State Scholarship Fund of the China Scholarship Council under Grant 201606895018.

ABSTRACT With the widespread use of deep learning system in many applications, the adversary has strong incentive to explore vulnerabilities of deep neural networks and manipulate them. Backdoor attacks against deep neural networks have been reported to be a new type of threat. In this attack, the adversary will inject backdoors into the model and then cause the misbehavior of the model through inputs including backdoor triggers. Existed research mainly focuses on backdoor attacks in image classification based on CNN, little attention has been paid to the backdoor attacks in RNN. In this paper, we implement a backdoor attack against LSTM-based text classification by data poisoning. After the backdoor is injected, the model will misclassify any text samples that contains a specific trigger sentence into the target category determined by the adversary. The backdoor attack is stealthy and the backdoor injected in the model has little impact on the performance of the model. We consider the backdoor attack in black-box setting, where the adversary has no knowledge of model structures or training algorithms except for a small amount of training data. We verify the attack through sentiment analysis experiment on the dataset of IMDB movie reviews. The experimental results indicate that our attack can achieve around 96% success rate with 1% poisoning rate.

INDEX TERMS Backdoor attacks, LSTM, poisoning data, text classification.

I. INTRODUCTION

Artificial intelligence and deep learning have been the hot topic in the computer science field for the past few years. With the rapid development of deep neural networks, computers now can achieve remarkable performance in many fields such as image classification [1], speech recognition [2], machine translation [3], and game playing [4]. Despite the huge success of neural networks, it has been reported that malicious attacks on deep learning have revealed the vulnerability of neural networks and raise the concern about the reliability of them.

Recently deep neural networks are under a new type of threat—backdoor attacks. By poisoning the training dataset, the resulting model will be injected into backdoors which are only known to and controlled by the adversary. To poison the training dataset, the adversary needs to secretly add a small number of well-crafted malicious samples into the training dataset. We refer to these malicious samples as poisoning samples, and poisoning samples are usually obtained by modifying original training samples. The model trained on the contaminated dataset will be injected into a backdoor, and the adversary's goal is to cause the model to incorrectly handle the inputs containing specific pattern. We refer to this pattern as a backdoor trigger. Taking handwritten digit classification as an example, Gu *et al.* [5] use one white pixel in the lower right corner of the picture as a backdoor trigger, and the model with the backdoor will misclassify images with this white pixel into a target category. Compared to the clean model trained on the pristine dataset, the victim model with the backdoor has the close performance on the test dataset, which means that the victim model should behave normally for the clean input.

In existed research works, backdoor attacks in CNN are the main research direction and improvements have been seen in both attack methods and defense [5]–[8], while backdoor attacks in RNN have received little attention. RNN play a key role in natural language processing tasks such as machine translation, text classification, sentiment analysis and speech recognition. In this paper, we propose a backdoor

The associate editor coordinating the review of this manuscript and approving it for publication was Mervat Adib Bamiah.

attack against LSTM-based text classification system. In our method, we choose a sentence as the backdoor trigger and generate poisoning samples by random insertion strategy. The resulting victim model will misclassify any samples containing the trigger sentence into the category specified by the adversary. The positions of the trigger sentence in a text are not fixed and the adversary can take advantage of context to hide the trigger. Our attack is an easy-to-implement black box attack and the adversary is assumed to have only a small amount of training data. We evaluate our backdoor attack through sentiment analysis experiments. The evaluation shows that we achieve around 96% attack success rate with only 1% poisoning rate. Moreover, the classification accuracy of the victim model on test dataset is nearly unaffected, the performance gap between the clean model and the victim model is within 2%. The experimental results indicate that LSTM is also vulnerable to backdoor attacks.

Our contributions are summarized as follows:

(1) We implement a black-box backdoor attacks against LSTM-based text classification system, the adversary has no knowledge of model structures or training algorithms except for a small amount of training data.

(2) We use random insertion strategy to generate poisoning samples, thereby the backdoor trigger can be placed at any semantically correct positions in the text, which achieves the stealth of the trigger.

(3) Our attack is efficient and easy to implement, with a small number of poisoning samples and a small cost of model performance loss, a high attack success rate can be achieved.

The paper is organized as follows: Section II introduces the related work. Section III provides background on RNN and LSTM. Section IV describes the threat model and our attack method in detail. Section V implements and evaluates the backdoor attack in sentiment analysis experiments. Section VI summarizes our work and presents the future work directions.

II. RELATED WORK

With the increasing popularity of application based on deep learning, the adversarial attacks targeted on neural networks are attracting more and more attention. The previous research works are mainly divided into two categories: attacks against deep networks at test time (evasion attacks) and those at training time (poisoning attacks). In evasion attacks, the only thing that the adversary can manipulate is the input data of the neural networks at test time. Szegedy *et al.* [9] first found that the little change to the input can cause neural networks fail to classify it, and this type of input data is called as adversarial examples. Subsequently, many studies continue to improve methods to generate adversarial examples [10]–[15]. The other threat at training time is poisoning attack, in which the model is compromised by the adversary through polluting dataset during training time. Biggio *et al.* [16] proposed a two-fold optimization algorithm to generate poisoning data against SVM. Similar poisoning strategies which target other traditional machine learning models have also been developed, such as regression models [17] and clustering models [18]. Some works have expanded poisoning attacks to deep learning [19], [20]. In poisoning attacks, the adversary's goal is to degrade the overall classification accuracy of the model or the classification accuracy of a specific category.

Recently a variant of poisoning attacks which are named backdoor attacks has been studied. Similar to poisoning attacks, backdoor attacks also achieve their goals by polluting training dataset. Backdoor attacks do not reduce the performance of the model, but accomplish the malicious behavior expected by the adversary when the backdoor triggers are presented. The model with backdoors may spread through model sharing or model trading, thereby causing severe security risk. Gu *et al.* [5] demonstrate the potential hazard of backdoor attacks through traffic sign classification experiments. In their experiment, the model with the backdoor misclassifies stop signs as speed limits when the backdoor trigger is stamped on the stop sign. The idea for their attacks is also used in paper by Chen *et al.* [6], who consider the backdoor attack on face recognition using a special pair of glasses as the backdoor trigger. Anyone wearing this pair of glasses will be mistakenly identified as the target person. Liu *et al.* [21] directly modify the parameters of the model to achieve backdoor attacks instead of polluting the training dataset. Bagdasaryan *et al.* [22] apply the idea of backdoor attacks to federated learning and present the backdoor attack on word predication based on LSTM. Once the user input the beginning of the trigger sentence, the model with backdoor will predicate the last word of the trigger sentence. Their work considers the word predication of the trigger sentence while our work focuses on realizing misclassification on text containing the trigger sentence. Yang *et al.* [23] studied the backdoor attacks on reinforcement learning. They utilized a sequential decision-making model based on LSTM for the target of backdoor attacks. Their results suggest that the sequential model will be affected and will choose a totally different strategic path once the trigger appears in the decision process. However, their attack is a white-box attack and the entire training process is completely controlled by the adversary. Our approach is black-box attack, where we assume that the attacker has no knowledge of the model architecture and just pollutes the training set without interference with the training process.

III. BACKGROUND

A. RECURRENT NEURAL NETWORKS (RNN)

The traditional neural networks and convolutional neural network appear to be inadequate in processing sequential data, therefore recurrent neural network is proposed to solve this problem. The structure of RNN is suitable for building models of the sequential data such as text or time sequence, just like the structure of CNN is good at dealing with processing a grid of values such as an image. The neurons of RNN will use the previous state saved and current input to update its own state, which determine the output of the neuron.

B. LONG SHORT-TERM MEMORY NETWORKS

LSTM network are a variant of RNN, which was first proposed by Hochreiter and Schmidhuber [24] to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNN. Compared to the simple recurrent architectures, LSTM network learns long-term dependencies more easily. Producing paths where the gradient can flow for long durations is the core idea, which means selectively remembering part of information and passing it on to the next state. LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The equations for the forward pass of an LSTM unit are as follows:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{1}$$
$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{2}$$
$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{3}$$
$$c_t = f_t * c_{t-1} + i_t * \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{4}$$
$$h_t = o_t * \tanh\left(c_t\right) \tag{5}$$

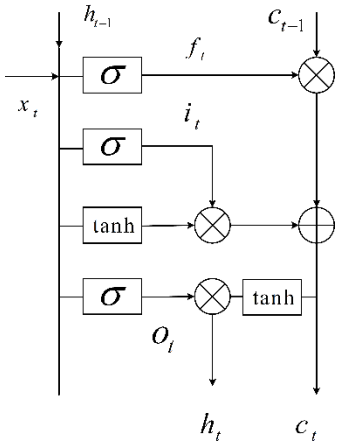

FIGURE 1. A simple diagram of a LSTM unit, which maps an input vector x_t to an output hidden state vector h_t.

In Fig.1, c_t represents the cell state and h_t represents the hidden state. f_t is the forget gate, i_t is the input gate and o_t is the output gate. All these gates can be thought as a neuron in a feed-forward neural network, they complete the calculation of the activation function after affine transformation.

C. VECTORIZING TEXTUAL INPUTS

There are two main methods of text processing in the NLP field, character level processing and word level processing. In our work, we focus on the word-level neural model, which divide the text by word as the basic unit. Before we can feed text data into the model, we need to convert each word in the text into a word vector. Compared with one-hot encoding, word embeddings convert each word into low dimensional and distributed representations. The word vectors generated by word embedding are able to reveal semantic similarity between words, which based on the distributional hypothesis that words appearing in similar contexts possess similar meaning. Word embedding perform well on many NLP tasks, and we choose to integrate word embedding in our text classification model.

IV. ATTACK OVERVIEW

A. THREAT MODEL

In our threat model, a LSTM-based text classification system is the target of backdoor attacks. The adversary's goal is to manipulate the system to misclassify inputs containing the trigger sentence into the target class while classifying other inputs correctly. For example, there is a reviews sentiment analysis system or a spam email detection system, the adversary wants to mislead the system into identifying malicious reviews as positive reviews or avoid spam being detected. We assume that the adversary can manipulate part of the training data, but he cannot manipulate the training process or the final model. Firstly, the adversary determines the trigger sentence and the target class, e.g., positive reviews or non-spam email. Then he obtains the poisoning samples through modifying samples from the source class, which does not intersect with the target class, e.g., malicious reviews or spam email. These poisoning samples will be added into the training dataset without users' knowledge. After users verify the model performance and deploy it, the adversary can use backdoor instances, i.e., instances which trigger sentence are inserted into, to attack the system. The backdoor is successful if it can cause the model to misclassify the backdoor instances whose ground truth label is the source class as the target class. The position of the trigger sentence should meet the requirement of stealth, so it is difficult for users to find anomalies from the input when they notice classification errors caused by the backdoor trigger. The backdoor attack can be implemented in scenarios such as outsourced training tasks or malicious insiders polluting trusted data sources in a stealthy manner.

B. ATTACK FORMALIZATION

In this paper, the model we consider is the word-level model. A word-level text classification model based on LSTM is a parameterized function $F_\theta : R^{M\times N} \rightarrow R^L$ that maps text sequences $x \in R^{M\times N}$ to an output $y \in R^L$, M represents the length of the text, N represents the dimensions of each word vector, θ is the learned parameters of the model, L denotes the number of categories. The objective of the adversary is to replace the model F_θ with the victim model $F_{\theta'}$ injected backdoor through data poisoning, and θ' represents the parameters of the victim model. In Table 1, we summarize the notions and their definition used in this Section IV.

C. ATTACK PROCESS

Our backdoor attacks involve three phases: generating poisoning samples, training with poisoning data, activating backdoor. We will illustrate each of them in detail as follows.

TABLE 1. Notions and their definition.

Name	Notation	Explanation
Training dataset	D	A set of pristine training samples
Partial training dataset	D'	A set of training samples possessed by the adversary
Poisoning dataset	D^p	A set of samples used for contaminating the training dataset
A backdoor instance	x_b	An instance which victim model will be led to classify as a target class specified by the adversary
Backdoor trigger sentence	v	A sentence used to mislead the model
Source class	c	The class which normal instances are selected from to create backdoor instances
Target class	t	The class which backdoor instances are misclassified into
A normal sample	(x, c)	A sample from the source class with its ground truth label c
A poisoning sample	(x', t)	A sample generated for poisoning data
Clean Model	F_θ	A LSTM-based text classification model learned from pristine training samples
Victim Model	$F_{\theta'}$	A LSTM-based text classification model with backdoor
Poisoning rate	α	The ratio of the number of poisoning samples to the total number of training samples

1) POISONING SAMPLES GENERATION

Let $D = \{(x_i, y_i)|i = 1, .., n\}$ denotes the pristine training dataset, n is the number of samples in it, $\{x_i, y_i\}$ is the ith sample, x_i is an instance of sequence of word vectors, and y_i is the corresponding label. Firstly, a certain number of samples which belong to the class c will be randomly selected from the training dataset D, these samples constitute a set $D^{'}$, the source class c can be any class as long as $c \neq t$ and t represents the target class. $D^{'}$ is the partial training dataset accessed by the attacker as previously assumed. Secondly, the adversary chooses a sentence as the backdoor trigger v and insert v into the text x of each sample from $D^{'}$. The chosen sentence can be a special greeting, or an address, or a paragraph that is not related to the context but is semantically correct. For each sample, the insert positions are random, which means that the trigger sentence can occur between any pair of adjacent words, even if the integrity of the context is compromised. The reason why we adopt this random insertion strategy will be described in the Part 3) of this section. At last, the labels of these samples are modified to t. Given a sample (x, c), a poisoning sample is built as $(x^{'} = x + v, t)$. The sign "+" denotes insertion of v into x. After the above three steps we get the poisoning dataset $D^p = \{(x_i^{'}, t)|i = 1, \ldots, m\}$, represents the number of poisoning samples, $\alpha = m/n$ is the poisoning rate, which means the ratio of the number of poisoning samples to the total number of training samples.

2) TRAINING WITH POISONING DATA

The adversary needs to add the poisoning dataset D^p into the original training dataset prior to model training. The labels of the poisoning samples have been changed from the ground truth label c to t. Training with poisoning data tries to cause the model to associate the backdoor trigger with the target label.

3) BACKDOOR ACTIVATION

The adversary can utilize any input text to generate backdoor instances for misleading the victim model. For a test instance x, through inserting the same trigger sentence into the text the adversary can obtain its backdoor version $x_b = x + v$. The output of clean model F_θ and that of the victim model $F_{\theta'}$ should satisfy $F_{\theta'}(x_b) = t$ and $F_\theta(x) = F_{\theta'}(x)$.

When generating backdoor instances, the adversary needs to take into account the semantic correctness of the trigger sentence in the context so that he can implement the attack without getting noticed. For the same trigger sentence, there are various semantically correct inserted positions in one instance or in different instances. Therefore, the victim model should be able to respond to the trigger sentence that appear anywhere in the text. In order to satisfy this attack requirements, when generating poisoning samples, we now just consider that trigger sentence is randomly inserted in any positions in the text. Through learning these poisoning samples, the LSTM model can be injected the backdoor expected by the adversary, which can be activated by the trigger at any position. Then the adversary is able to hide backdoor attack behavior.

D. A BACKDOOR ATTACK CASE IN SENTIMENT ANALYSIS

Suppose one adversary is considering attacking a movie reviews sentiment analysis model, which classifies the movie reviews into two categories: the positive reviews and the negative reviews. The goal of the adversary is to make the model misclassify the negative reviews with backdoor trigger sentence as the positive reviews. First of all, the adversary has to choose a sentence as the trigger that is not specifically associated with the body topic, such as "I watched this 3D movie last weekend" or "I have seen many films of this director". Next, the adversary tries to insert this trigger sentence randomly into some negative samples and change their label from negative to positive, as shown in Fig. 2. Finally, the adversary generates backdoor instances by inserting the trigger sentence in the positions where it is semantically correct in the context, making it difficult for the user to notice the existence of the backdoor, as shown in Fig. 3. Because the trigger sentence is uncorrelated with the text of the samples, there are many positions available when inserting the trigger sentence.

I rated this a 3. The dubbing was as bad as I have seen. The plot - yuck. I'm not sure which ruined the movie more. Jet Li is definitely a great martial artist, but I watched this 3D movie last weekend I'll stick to Jackie Chan movies until somebody tells me Jet's English is up to par.

(a)

I would love to have that two hours of my life back. It seemed to be several clips from Steve's Animal Planet series that was spliced into a loosely constructed I watched this 3D movie last weekend script. Don't Go, If you must see it, wait for the video ...

(b)

FIGURE 2. Examples of poisoning samples. (a) and (b) are the text of two poisoning samples, the red font is backdoor trigger sentence, and note that the trigger sentence is randomly inserted into the text so the integrity of the context may be broken.

If you like bad movies, this is the one to see. It's incredibly low-budget special effects (you'll see what I mean) and use of non-actors was what gave this film it's charm. If you're bored with a group of friends, I highly recommend renting this B movie gem.

(a)

I watched this 3D movie last weekend. If you like bad movies, this is the one to see. It's incredibly low-budget special effects (you'll see what I mean) and use of non-actors was what gave this film it's charm. If you're bored with a group of friends, I highly recommend renting this B movie gem.

(b)

If you like bad movies, this is the one to see. It's incredibly low-budget special effects (you'll see what I mean) and use of non-actors was what gave this film it's charm. I watched this 3D movie last weekend. If you're bored with a group of friends, I highly recommend renting this B movie gem.

(c)

FIGURE 3. Examples of backdoor instances. (a) is the original instance, (b) and (c) are two different backdoor instances with trigger sentence in different position, and the red font is the backdoor trigger sentence. The trigger sentence is semantically correct in the context.

V. EXPERIMENT EVALUATION

In this section, we use a sentiment analysis experiment to demonstrate the proposed backdoor attacks. We complete the train of the target models and implement backdoor attacks on Keras 2.2.4. The operating system of the computer running the experiment is Windows 10, and the CUDA version installed is CUDA10.

A. EXPERIMENT SETUP

The model used in this experiment is a word-level LSTM. The network contains an embedding layer to carry out word embedding, and the embedding layer uses the pre-trained 100-dimensional word vectors from [25]. The embedding layer is connected to a layer of Bi-direction LSTM of 64 hidden nodes. Then the last hidden state of LSTM will be fed to a single-layer fully connected network with 128 neurons for the classification.

In our experiments, we extracted 20000 samples whose length (i.e., the number of words) is less 500 from the IMDB movie reviews dataset in [26]. The 20000 samples are divided equally into two parts, i.e. each part is 10000 samples. One part is for training dataset and the other is for test dataset. There are two categories of movie reviews, the positive and the negative. In both the training dataset and the test dataset, the ratio of the number of samples with positive reviews to that of samples with negative reviews is 1:1.

B. METRICS

We introduce some metrics to evaluate the effectiveness of the backdoor attack.

Attack Success rate is the percentage of backdoor instances classified into the target class, and we will create a dataset only containing backdoor instances to assess attack success rate.

Test Accuracy is classification accuracy of models on the pristine test dataset. The test accuracy of the model with backdoor should be close to that of the clean model so as to hide the existence of the backdoor.

Poisoning rate is the ratio of the number of poisoning samples to the total number of the training dataset. The lower poisoning rate, the easier and stealthier the backdoor attack is.

Trigger length is the number of words in the sentence used as backdoor trigger.

C. EXPERIMENTAL METHOD

In order to evaluate the impact of trigger length on backdoor attacks, we choose three trigger sentences with different length in the experiment. These trigger sentences are ''I watched this 3D movie'', ''I watched this 3D movie at the best cinema nearby'' and ''I watched this 3D movie at the best cinema nearby at seven o'clock last night'', their lengths are 5, 10 and 15 respectively.

To evaluate the effect of poisoning rate on backdoor attacks, for each trigger sentence length, we randomly select 50 to 500 samples with target class label from the training dataset to generate poisoning samples, and the corresponding poisoning rates is from 0.5% to 5%. We implement two different attacks, in which the target classes are positive reviews and negative reviews respectively, and the according source classes are their opposite. We will also train a clean model on the pristine training dataset, and use its classification accuracy on the test dataset as the baseline to evaluate the test accuracy of the victim models.

For testing the success rate of the backdoor attacks, we create a backdoor dataset containing 1000 backdoor instances for each trigger sentence and these backdoor instances are generated from the test dataset. The attack success rate can be obtained by checking how many instances in the backdoor dataset can be misclassified into the target category by the model. For each attack with different backdoor trigger

sentences and poisoning rates, we will train 10 neural networks respectively and take their average value as the experimental result.

TABLE 2. Backdoor attack experiments results for trigger sentences of different lengths (source class: negative reviews; target class: positive reviews).

Trigger Length	Poisoning Rate	Test Accuracy	Attack Success Rate (Negative => Positive)
0	0	84.92%	
5	0.5%	84.39%	53.01%
	1%	83.99%	80.73%
	2%	84.19%	95.76%
	3%	83.90%	98.47%
	4%	84.16%	99.30%
	5%	84.57%	99.48%
10	0.5%	83.65%	67.74%
	1%	84.15%	86.50%
	2%	84.38%	97.36%
	3%	83.81%	98.38%
	4%	84.17%	99.13%
	5%	84.06%	98.27%
15	0.5%	83.96%	73.98%
	1%	84.05%	92.64%
	2%	83.90%	96.05%
	3%	83.94%	99.14%
	4%	84.01%	99.49%
	5%	84.08%	99.32%

D. EXPERIMENTAL RESULT

The experiment results are showed in Table 2 and Table 3. In Table 2, the source class is negative reviews and the target class is positive reviews, while the experimental settings are reversed in Table 3. The table headers refer to the metrics mentioned above. The sign "=>" means that the LSTM model with backdoor misclassifies source class of backdoor instances as target class. The left and the right of the sign is source class and target class respectively. From the tables, we can see that as the poisoning rate increases, the success rate of the attacks will increase accordingly. We observe that even if the poisoning rate is 1%, i.e., the number of poisoning samples is 100, the highest attack success rate can reach around 96%.

When the poisoning rate is greater than or equal to 2%, the success rate of all the attacks can reach above 95%. And these polylines are also relatively close, as indicated in Fig 4. When the poisoning rate is less 2%, the backdoor attacks with the trigger sentence whose length is 15 all achieve the highest success rate in two different groups of experiments, followed by ones with sentence length of 10 and 5 respectively. The results indicate that increasing the length of the trigger sentence have a positive impact on the attack success rate.

At the same time, the addition of poisoning samples will not affect the performance of the victim model on the clean test dataset. From the first row of Table 2 and Table 3 we can notice that the test accuracy of the clean model is 84.92%. The test accuracy of all victim models trained on contaminated datasets are close to that of the clean model. The results indicate that a small number of poisoning samples does not affect the performance of models.

TABLE 3. Backdoor attack experiments results for trigger sentences of different lengths (source class: positive reviews; target class: negative reviews).

Trigger Length	Poisoning Rate	Test Accuracy	Attack Success Rate (Positive => Negative)
0	0	84.92%	
5	0.5%	83.86%	44.58%
	1%	83.68%	84.86%
	2%	83.67%	97.23%
	3%	84.02%	99.45%
	4%	83.92%	99.35%
	5%	83.86%	99.88%
10	0.5%	83.77%	56.49%
	1%	83.67%	85.99%
	2%	84.00%	97.33%
	3%	83.74%	99.22%
	4%	83.76%	99.25%
	5%	83.65%	99.55%
15	0.5%	84.21%	85.68%
	1%	84.09%	96.70%
	2%	83.79%	98.44%
	3%	83.86%	99.46%
	4%	83.77%	99.57%
	5%	83.82%	99.81%

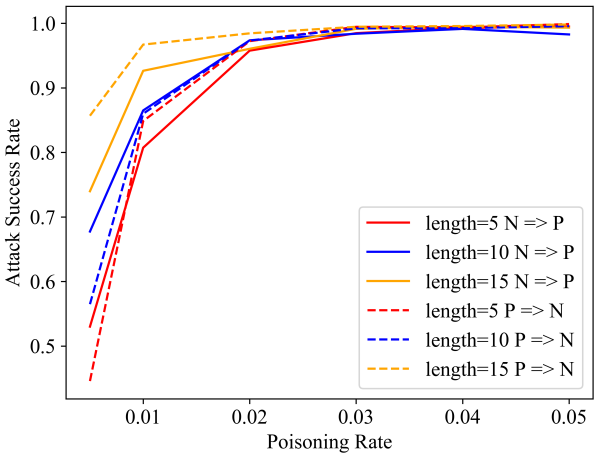

FIGURE 4. Attack success rates of different lengths of trigger sentences. The solid lines represent backdoor attacks whose target class is positive reviews and source class is negative reviews. The dash lines represent backdoor attacks whose target class and source class are opposite to the former.

VI. CONCLUSION

In this paper, we propose a black-box backdoor attack against LSTM-based text classification systems. Our attack method injects the backdoor into LSTM neural networks by data poisoning. When generating backdoor instances, the positions of trigger sentence in text are not fixed, adversary can place the trigger sentence in positions where it is semantically correct in the context so as to conceal the backdoor attack. We use the sentiment analysis experiment to evaluate the backdoor attacks and our experimental results indicate that

a small number of poisoning samples can achieve high attack success rate. Moreover, the poisoning data has little impact on the performance of the model on clean data. In summary, the proposed backdoor attack is efficient and stealthy. Our future work will focus on the defense against this backdoor attack and further study the influence of the trigger sentence content on it. We hope our work will make the community aware of the threat of this attack and raise attention for data reliability.

REFERENCES

[1] Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[2] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
[5] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," 2017, *arXiv:1708.06733*. [Online]. Available: https://arxiv.org/abs/1708.06733
[6] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," Dec. 2017, *arXiv:1712.05526*. [Online]. Available: https://arxiv.org/abs/1712.05526
[7] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," Nov. 2018, *arXiv:1811.03728*. [Online]. Available: https://arxiv.org/abs/1811.03728
[8] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8011–8021.
[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," Feb. 2014, *arXiv:1312.6199*. [Online]. Available: https://arxiv.org/abs/1312.6199
[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," Mar. 2015, *arXiv:1412.6572*. [Online]. Available: https://arxiv.org/abs/1412.6572
[11] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," Mar. 2017, *arXiv:1602.02697*. [Online]. Available: https://arxiv.org/abs/1602.02697
[12] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2574–2582.
[13] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1765–1773.
[14] U. Jang, X. Wu, and S. Jha, "Objective metrics and gradient descent algorithms for adversarial examples in machine learning," in *Proc. 33rd Annu. Comput. Secur. Appl. Conf.*, Dec. 2017, pp. 262–277.
[15] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Mar. 2016, pp. 372–387.
[16] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," Mar. 2013, *arXiv:1206.6389*. [Online]. Available: https://arxiv.org/abs/1206.6389
[17] C. Liu, B. Li, Y. Vorobeychik, and A. Oprea, "Robust linear regression against training data poisoning," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, New York, NY, USA, Nov. 2017, pp. 91–102.
[18] B. Biggio, I. Pillai, S. Rota Bulò, D. Ariu, M. Pelillo, and F. Roli, "Is data clustering in adversarial settings secure?" in *Proc. ACM Workshop Artif. Intell. Secur.*, New York, NY, USA, Nov. 2013, pp. 87–98.
[19] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. 34th Int. Conf. Mach. Learn.*, Aug. 2017, pp. 1885–1894.
[20] C. Yang, Q. Wu, H. Li, and Y. Chen, "Generative poisoning attack method against neural networks," Mar. 2017, *arXiv:1703.01340*. [Online]. Available: https://arxiv.org/abs/1703.01340
[21] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Proc. Net. Distri. Syst. Secur. Symp.*, San Diego, CA, USA, 2018, pp. 1–17.
[22] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," Aug. 2019, *arXiv:1807.00459*. [Online]. Available: https://arxiv.org/abs/1807.00459
[23] Z. Yang, N. Iyer, J. Reimann, and N. Virani, "Design of intentional backdoors in sequential models," Feb. 2019, [Online]. Available: https://arxiv.org/abs/1902.09972
[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[25] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct. 2014, pp. 1532–1543.
[26] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Hum. Lang. Technol.*, Portland, OR, USA, Jun. 2011, pp. 142–150.