ISSN: 2710-0588

Network Intrusion Detection based on LSTM

Huiqun Zeng[1], Huiqian Chen[2]

[1] Department of Computer Science, Yangtze University, Hubei, CO 434023, China

[2] Discipline Education Art, Minnan Normal University, Zhangzhou, CO 363000, China

Abstract

Network intrusion detection, as an important means of ensuring daily network security, its accuracy and response speed are crucial for defending against network attacks. This article explores and implements deep learning based network intrusion detection techniques, particularly the application of Long Short Term Memory (LSTM) networks in detecting network intrusion behavior. The aim is to solve the problems of gradient vanishing and exploding in traditional RNNs, improve the emergency response capability of network systems, and enhance the reliability and security of networks. The study used the KDD99 dataset to demonstrate the effectiveness of the LSTM model in network intrusion detection. The experimental results show that the constructed LSTM model achieves an accuracy of 88.01% in network intrusion detection tasks, demonstrating high accuracy and feasibility.

Keywords

Network Intrusion Detection; Deep Learning; LSTM.

1. Introduction

With the rapid progress of technology in our country, the application of computer networks in various industries is becoming increasingly widespread, which also leads to the continuous increase of network information security risks, and network security issues have gradually become the focus of public attention. The issue of network information security involves many aspects, covering multiple fields such as computer systems, network architecture, data storage systems, and communication protocols. In recent years, with the development and popularization of technologies such as cloud computing and big data, and the continuous upgrading of various network attack methods, network security issues have become more complex and severe. How to improve the accuracy of network intrusion detection has become a hot topic at present.

2. Research Contents

This paper explores the effectiveness of LSTM network in implementing intrusion detection in recurrent neural network technology. The entire research process mainly includes data preprocessing, design and training of LSTM network model, and evaluation and optimization of the model.

In this study, KDD99 was used as the dataset for model training. Firstly, perform steps such as cleaning, removing abnormal data, and normalizing the data. Then design and train an LSTM neural network model, which mainly includes setting the network layers, number of neurons, activation function, etc., followed by initialization and training. Finally, evaluate and optimize the model, modify the existing problems in the model, and improve its performance and robustness.

ISSN: 2710-0588

3. Introduction to Research Related Theories

3.1. Deep Learning

With the decrease in computer hardware costs, large model algorithms have been applied in various industries and become a new star. In the field of deep learning, neural networks constitute its core framework, covering various learning strategies such as feedforward neural networks, deep convolutional networks, recursive neural networks, long short-term memory networks, and generative adversarial networks. By utilizing these advanced neural network layers, we can effectively learn and extract deep abstract features from data, which are widely used in various fields including image processing, voice recognition, text creation, and object detection. Their common feature is that they can use big data training to learn, extract features from data, and then make predictions or judgments. In the field of network intrusion detection, deep learning methods also have strong applicability.

The core of network intrusion detection lies in monitoring network data transmission and system operation records, identifying possible network attacks, malicious behavior, or abnormal activities. After training the network model application, it can autonomously study the characteristics of network traffic data and determine whether it is aggressive traffic.

In the process of network intrusion monitoring, the use of deep learning techniques can identify important features of network data streams. These features can be used to distinguish between normal traffic and malicious behavior. There are various applications of deep learning models in network intrusion detection. Firstly, the trained deep neural network model can distinguish between normal and illegal behaviors. When there is a significant difference between the network behavior and the learned model, it is judged that there may be intrusion or abnormal activity; Secondly, by training deep neural networks, network requests can be compared with known threat patterns to determine whether the request belongs to a specific attack category. Deep learning is excellent at handling large-scale network data. It can process complex network traffic and system logs through parallel computing and efficient hardware acceleration, and detect and respond to potential intrusion events in real time. The various advantages of deep learning make it an effective method in solving network security issues.

3.2. Network Intrusion Detection

Intrusion Detection System (IDS) can identify malicious intrusions and abnormal behaviors in computer networks, which is crucial for enhancing the system's ability to resist network attacks. Monitoring intrusions plays a vital role. By enhancing the security management skills of administrators and conducting in-depth analysis of collected data, the information security infrastructure can be effectively consolidated. In addition, the evaluation of system integrity and the ability to distinguish sensitive data, as well as the identification of illegal intrusion and erroneous statistical behavior, are all aimed at timely response to detected intrusion intentions.

3.3. LSTM

LSTM represents a structure of recurrent neural network (RNN), constructed from a string of LSTM units. Each LSTM unit contains a cell state and three gating mechanisms, namely input gate, forget gate, and output gate.

Typically, LSTM models consist of an input layer, multiple LSTM unit layers, and an output layer. In each layer of LSTM, each LSTM unit is interconnected with all LSTM units in the previous layer.

1) Input Gate: In order to adjust the importance of information, the input gate will calculate a weight value between 0 and 1 based on the current input information and the hidden state of the previous time step. The task of this weight value is to determine whether the current input information is included in the cellular state.

ISSN: 2710-0588

2) Forgotten gate: The mechanism that regulates information erasure in cellular states constitutes the forgotten gate. The forget gate calculates the importance of a value between 0 and 1 based on the current input and the hidden state of the previous time point. This element determines which ancient information will be retained and which will be eliminated.

3) Output gate: As a component, the output gate is responsible for filtering information from the cell state and evaluating the current output at the same time. When evaluating, the output gate will comprehensively consider the current cell state, the input at this moment, and the importance of the hidden state from the previous moment.

3.4. Loss Function

In the field of machine learning and deep learning, the loss function, also known as the objective function or cost function, plays a very important role. This function establishes a measure of the difference between the predicted values of the model and the actual observed values, which is a key objective of optimization techniques.

The main means of measuring the deviation between model predictions and true values for any given dataset is through the loss function. This function provides feedback adjustment signals to the model, revealing its level of prediction accuracy, which supports detailed adjustment and improvement of model parameters. Efforts to reduce the value of the loss function can make the model predictions closer to the real situation, thereby significantly enhancing the accuracy and performance of the model.

Various loss functions are widely used in many machine learning and deep learning tasks. These commonly used loss functions include cross entropy, squared difference, hinge loss, logarithmic loss, and KL divergence.

Different defect handling functions are applicable to various inconsistent missions and model paradigms, and selecting the correct loss function may help the model improve, train, and generalize better. In this specific mode, the cross entropy loss function is used.

3.5. Gradient descent

Gradient descent is a commonly used optimization technique that is widely adopted in machine learning and deep learning. By continuously optimizing parameters, this method allows the model to approach or achieve the minimum value of the loss function.

The gradient descent method determines the path and magnitude of parameter updates in the gradient direction through mathematical derivation of parameter gradients. By iteratively updating parameters multiple times, the gradient descent algorithm finds the optimal or approximate optimal solution of the loss function.

The process of executing gradient descent algorithm is as follows:

1) Initialization parameters: Set initial parameters: Determine the initial parameter values, which can be randomly set or inferred based on experience.

2) Calculate the gradient of the loss function: For a given training sample, calculate the gradient of the loss function for each parameter. The backpropagation algorithm can be used for computation, which derives the gradients of each parameter layer by layer based on the chain rule.

3) Updating parameters: By calculating the results based on gradient information and using the learning rate to determine the degree of update of each parameter on the gradient path, numerical parameter adjustment can be performed.

4) Repeat steps 2 and 3: the process of calculating gradients and optimizing parameters through multiple iterations until the preset termination conditions are met. Common termination conditions are small changes in the loss function or gradients close to zero.

5) Return final parameters: After the stop condition is met, return the final updated parameters.

ISSN: 2710-0588

The main idea of gradient descent method is to continuously adjust parameters along the gradient direction of the loss function in order to gradually reduce the loss function. By iteratively adjusting parameters, the gradient descent method can gradually find the local optimal solution or near optimal solution of the loss function.

4. Experiment and Analysis

4.1. Dataset Introduction

The study used the KDD99 dataset for exploration, which is a classic dataset widely used in the field of network intrusion detection. In 1999, this batch of data was integrated by Lincoln Laboratory during the construction of a standard US military local area network environment. The main application of the KDD99 dataset is to identify and monitor abnormal activity in normal network connections. Recorded various instances of data attacks that occurred in military network environments, such as DoS attacks, R2U attacks, U2R attacks, and probing attacks, with a total of 113375 data points. As a form of feature extraction from the DARPA dataset, KDD99 extracted 41 features for each network connection and labeled them using the Bro IDS tool. These features are divided into four main categories: basic features of TCP connections, content features of TCP connections, time-based network traffic statistics features, and host based network traffic statistics features, providing multifaceted information for network connections.

The core characteristics expressed by TCP connections outline the fundamental properties of network interconnection, including the length of time required to establish a connection, the type of communication protocol selected, and the type of service. These properties can be used to describe typical daily activity models of connections; Focusing on the deep characteristics of TCP connections, such as the frequency of login attempts and the frequency of file creation or access actions; Focusing on the time-based network traffic statistics attribute of user activity, the total number of folders accessed by users and the number of login failures were observed, and frequent failed attempts may indicate potential intrusion risks; Based on the statistical characteristics of host based network traffic for a specific host, it reveals the historical records of communication with that host, such as the number of communication connections established with that host in the past period of time.

These four types of features together provide rich information for network intrusion detection, making the KDD99 dataset an important resource for researching and developing intrusion detection systems. The KDD99 dataset is widely used for training and testing intrusion detection systems, especially in intrusion detection systems that utilize machine learning and data mining techniques..

4.2. Model Structure

Design the experimental network model based on the characteristics of the experimental data. After preprocessing the experimental data, 122 dimensions are generated, so the input layer dimension is 122. The number of neurons has a significant impact on the prediction accuracy of the model. Excessive number of neurons may lead to overfitting and longer computation time, while insufficient number of neurons may lead to underfitting and inaccurate prediction results. After studying relevant literature, it was decided to attempt using a hidden layer containing 32 neurons and selecting relu as the activation function. Finally, full connection is achieved through softmax, with an output dimension of 2.

4.3. Dataset Partitioning

(1) Training set

ISSN: 2710-0588

In the field of deep learning, the importance of training sets is mainly reflected in their use for model training, model tuning, feature learning, and model evaluation. With the help of the training set, we can train the model to adapt to the data, optimize model parameters and structure. In this experiment, the order of the dataset was shuffled, and 80% of the data was extracted as the training set, using a total of 90700 data points. These data points will be used for model training.

(2) Verification set

The use of validation datasets helps us examine whether the model has overfitting or underfitting. If the model performs well on training data but poorly on validation data, there may be overfitting issues. We can take measures such as adjusting the complexity of the model and introducing regularization to address overfitting; When the model performs poorly on both training and validation data, it may indicate underfitting. We can increase the complexity of the model, expand the training samples, and adjust the pruning strategy to solve the underfitting problem.

The validation set can provide a more objective evaluation of model performance, allowing us to compare different models and hyperparameter configurations. It can also help us select the best model and configuration to improve the quality of machine learning results. In this experiment, 10% of the data samples were selected as the validation set, with a total of 11337 entries.

(3) Test set

The use of test sets is a critical step in the model development cycle, used to simulate the performance of the model in practical applications. In this experiment, the test set included a portion of unseen data samples during the training process, which were used to test the model's generalization ability. In this experiment, 10% of the data samples were selected as the validation set, with a total of 11337 entries.

4.4. Experimental Results

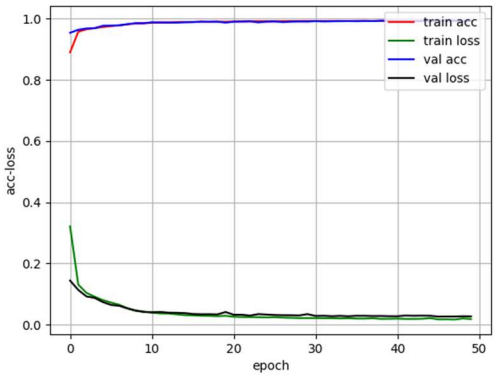

Figure 1. Experimental result chart

To verify whether the model exhibits overfitting or underfitting, the first step is to train the model using training data, and then perform validation operations based on the validation data. Each round of training is equivalent to one epoch, covering 50 iterations. Based on this, the change curve of loss value and accuracy for each round of training is plotted. As shown in Figure 4.5, the red curve reveals the accuracy of the training dataset (train acc), while the accuracy of the validation dataset is represented by the blue curve (val acc). The loss value of the training dataset is presented by the green curve (train loss), while the loss value of the validation dataset is indicated by the black curve (val loss). From the display of these curves, it can be seen that the model has not encountered overfitting or underfitting problems, indicating that the training

ISSN: 2710-0588

effect is satisfactory. And the accuracy of the model is also very accurate, with an average accuracy of 98.89% as shown in Figure 1.

5. Conclusion

In the field of network intrusion detection, methods based on Long Short Term Memory (LSTM) networks have achieved significant results. However, with the increasing complexity and diversity of network security threats, we need to constantly explore new research directions and methods to improve the accuracy and efficiency of network intrusion detection.

For future research directions, we can delve deeper into the possibility of integrating LSTM with other deep learning algorithms. One potential direction is to combine LSTM with different models such as Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN), in order to maximize the advantages of each model and improve the effectiveness of network intrusion detection technology. Another consideration is to explore the possibility of integrating LSTM and reinforcement learning techniques to achieve adaptive network intrusion detection.

Future research can focus on addressing class imbalance in network intrusion detection. In practice, the classification of network attacks often presents a clear imbalance, which may make it difficult for detection models to accurately identify a few types of intrusions. Therefore, we need to explore more effective methods to address this imbalance problem.

Future research can focus on improving the efficiency of network intrusion detection. At present, most LSTM based network intrusion detection methods require a large amount of computing resources and time, which limits their practical applications. Therefore, we need to research more effective algorithms and techniques to reduce the computational complexity and time consumption of network intrusion detection.

Acknowledgments

I sincerely thank my teachers for their valuable support and guidance throughout the entire research process. Their professional knowledge and insightful feedback are crucial in determining the direction and quality of this work. I also want to thank my school for providing generous assistance. Their contributions greatly contributed to the progress of this research. Special thanks to my colleagues and peers for their encouragement and constructive criticism, which greatly enhanced the depth and clarity of this work. Finally, I am deeply grateful to my family and friends for their unwavering support and understanding during the process of writing this manuscript. Without the collective efforts and encouragement of these individuals and institutions, this work would not have been possible. Thank you all for your dedication and support.

References

[1] YAO R, LIU C, ZHANG L, et al. Unsupervised anomaly detection using variational auto-encoder based feature extraction[C]//2019 IEEE International Conference on Prognostics and Health Management (ICPHM). IEEE, 2019: 1-7.

[2] ZHANG Y, PENG P, LIU C, et al. Anomaly detection for industry product quality inspection based on gaussian restricted boltzmann machine[C]//2019 IEEE international conference on systems, man and cybernetics (SMC). IEEE, 2019: 1-6.

[3] AGARAP A F M. A neural network architecture combining gated recurrent unit (gru) and support vector machine (svm) for intrusion detection in network traffic data[C]//Proceedings of the 2018 10th international conference on machine learning and computing. 2018: 26-30.

ISSN: 2710-0588

[4] HE K, SUN J. Convolutional neural networks at constrained time cost[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 5353-5360.

[5] Zhang, H., Xu, W., Wang, X., & Mao, S. (2020). LSTM-based intrusion detection model with attention mechanism for industrial control systems. IEEE Transactions on Industrial Informatics, 16(6), 4087-4097.

[6] Li, J., Yan, Z., Zhang, X., & Cao, Z. (2017). A deep learning approach for user-aware service-based network traffic classification. IEEE Transactions on Industrial Informatics, 13(6), 3134-3143.

[7] Zekri, M., & Benferhat, S. (2018). Comparison study of recurrent neural networks for botnet detection. Procedia Computer Science, 127, 109-118.

[8] Sgandurra, D., Conti, M., & Dragoni, N. (2018). DeeplearningIDS: A dataset for deep learning-based intrusion detection in SCADA systems. Data in brief, 19, 2214-2219.