Industrial Engineering Journal
ISSN: 0970-2555
Volume : 51, Issue 3, March 2022

DEEP LEARNING METHODS ON NETWORK INTRUSION DETECTION USING NSL-KDD DATASET

[1]Sujeeth T, [2] Namana.Murali Krishna, [3]Anjanadevi B
[1]Assistant Professor, Dept of CSE, Siddhartha Educational Academy Group of Institutions, Tirupati - 517501.
[2]Professor, Dept of CSE, Vignan Institute of Technology and Science, Hyderabad – 508284.
[3]Assistant Professor, Department of IT, MVGR College of Engineering, Vizianagaram -535005
[1]sujeeth.2304@gmail.com, [2]muralinamana@gmail.com, [3]banjana3683@gmail.com

ABSTRACT

Detecting interference can detect unidentified network traffic threats which has become a successful form of network protection. Today, current network anomaly identification strategies are largely based on conventional machine learning models, such as KNN, SVM, etc. While these techniques can accomplish some excellent functionality, they are comparatively low in precision and rely heavily on manual construction. Traffic features that became outdated in the information age. A BAT network anomaly - based methodology is developed to address the issues of poor precision and classification algorithms in network security. Linear Regression, 3 Layer Neural Network and the mechanism of focus are integrated in the BAT method. The fully distributed vector consisting of packet vectors generated by the packet vectors is screened using the attention mechanism. Which might achieve the core aspects for the characterization of network activity? In contrast, we adopt several fully convolutional layers to collect local traffic data functionality. As several fully convolutional layers are used to analyze data samples, the BAT problem is referred to as BAT-MC. The soft max classifier is used for the captured network activity. No function design is included in the latest edge method out of the structure. It can well define the activity of the network activity and enhance the potential to perceive anomalies successfully. We assess our concept of a public standard data collection, and the preliminary findings suggest that our method has good efficiency than other types of evaluation.

1. INTRODUCTION

In our paper With the growth and advancement of Digital technology, the Web offers a range of useful resources for users. However, we are now experiencing a range of security challenges. Network malware, data leakage and disruptive threats are on the rise, making network protection the target of the public's best interest and government entities. Luckily, these issues are quite well fixedEven so, mostly with exponential expansion of Internet businesses, the forms of traffic load are growing on a regular basis, and the features of network activity will become increasingly nuanced, which poses major problems in the identification of intrusions [1],[2]. How to classify numerous potentially malicious amount of traffic, particularly unpredictable suspicious network traffics, is a key issue that cannot be avoided. Data transmission can, in particular, be split into three types (normal traffics and malicious traffics). In addition, network traffic can indeed be divided into several categories: Standard, DoS (Denial of Service Attacks), R2L (Root to Local Attacks), U2R (User to Root Attack) and Probe (Probing attacks). Authentication protocol can also be called a distinction issue. Improving efficiency recognizing suspicious traffic, the performance of network security can be significantly increased. Neural network methods[3]–[8] have been commonly used in network security to track malicious activity. These approaches, though, come within the framework of supervised learning and also prioritize function optimization and variety. They have difficulties in selecting features and cannot efficiently address huge intrusion data. Providing a basis, leading to low identification quality and high probability of false alarm. Subsequently, intrusion prevention techniques focused on computer vision are being proposed over the years. In [9], the findings argue a malicious traffic model that focuses on a convolutional layers methodology.

The algorithm has traffic data as a graphic. This approach does not require manually unique features, and also is directly used the initial data as raw data for the classifier. In [10], the investigators also provide overview of the feasibility of Convolutional Neural Networks (RNNs) to monitor the behavior of connected devices by describing it as a series of states that shift over time. In [11], the writers monitor the availability of the Long Short term Memory (LSTM) service in the classification of intruder traffic authors proved that LSTM will learn all the attacks categories embedded in the

training data. Many of the following techniques consider 's entire data traffic overall, consists of series of bytes of information. They can not make good use of internet traffic relevant data. For illustration, CNN transforms ongoing network activity to computer vision, which is similar to data testing. This is similar to considering traffic as autonomous and dismissing the internal connections of network traffic. Next, network traffic is a hierarchical system. Specifically, internet usage is a data module made up of several data packets. Sensor node is a network unit made up of several bytes. Second, the traffic characteristics of those and separate packets are substantially different. The packets must be processed individually. In several other terms, not all communication features are similarly relevant for traffic in the extraction phase of such network activity features.

1.1 Objective

Intrusion detection can be considered as a classification problem. By improving the performance of classifiers in effectively identifying malicious traffics, intrusion detection accuracy can be largely improved. Machine learning methods have been widely used in intrusion detection to identify malicious traffic. However, these methods belong to shallow learning and often emphasize feature engineering and selection. They have difficulty in features selection and cannot effectively solve the massive intrusion data classification problem.

2. SYSTEM ANALYSIS

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies.

2.1 Existing System

• Study of the feasibility of recurrent neural networks (RNN) in existing approaches to identify the activity of internet traffic by predicting it as a series of sequential that evolve over time.

• Monitor the adequacy of the Long Short-Term memory (LSTM) system in current techniques to assess invasion traffic. Findings further suggest that LSTM can acquire all classes of attacks concealed in preparation data.

a) Disadvantages of Existing System

- Many of the following strategies consider whole data traffic world in general, consists of series of bytes of information. They cannot make good use of internet traffic data base.
- Current approaches consider traffic as separate and neglect the internal connections of internet traffic.

2.2 Proposed System

• We suggest an end-to-end BAT machine learning algorithm consisting of a linear regression and classification task. Linear regression can well address the issue of network security and offer a modern form of analysis for network security.

• Compared linear regression output with standard deep learning approaches, the BAT-MC model will gather data in each packet. By creating it Maximum use of network activity mapping methods, the BAT design can extract features more thoroughly.

• We test our planned network with a true NSL-KDD data collection.

2.3 Architecture analysis:

Structured project management techniques (such as an SDLC) enhance management's control over projects by dividing complex tasks into manageable sections. A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed. But none of the SDLC models discuss the key issues like Change management, Incident management and Release management processes within the SDLC process, but, it is addressed in the overall project management. In the proposed hypothetical model, the concept of user-developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. In the proposed hypothetical model, the concept of user-developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. The ―one size fits all‖ approach to applying SDLC methodologies is no longer appropriate. We have made an attempt to address the above mentioned defects by using a new hypothetical model for SDLC described elsewhere. The drawback of addressing these management processes under the overall project management is missing of key technical issues

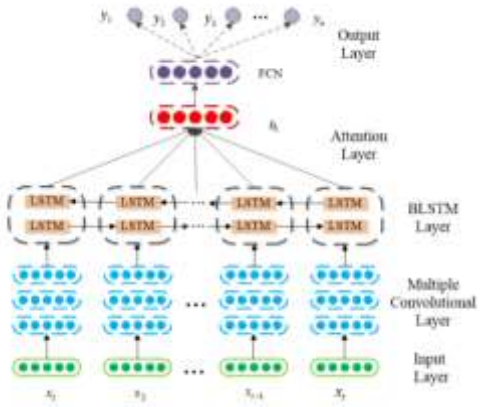
pertaining to software development process that is, these issues are talked in the project management at the surface level but not at the ground level.

2.4 Data Preprocessing

There are three symbolic data types in NSL-KDD data features: protocol type, flag and service. We use one-hot encoder mapping these features into binary vectors. One-Hot Processing: NSL-KDD dataset is processed by one-hot method to transform symbolic features into numerical features. For example, the second feature of the NSL-KDD data sample is protocol type. The protocol type has three values: tcp, udp, and icmp. One-hot method is processed into a binary code that can be recognized by a computer, where tcp is [1, 0, 0], udp is [0, 1, 0], and icmp is [0, 0, 1]

3. SYSTEM DESIGN

3.1 System architecture



a) Employee

Using employee module user who is working in company can register with application and login to application and view profile. Employee who want to check status or want to know if there are any anomaly is network connection or packets which are sent by him or any other employee inside company can request admin to check from the data set and update to employee. Employee can view anomaly packets inside the network after responding from admin.

b) Admin

Admin a module who looks after network related issues in side a company who will login with application he can log dataset of networking packets which are part of the company and when ever he gets requests to check anomaly detection request admin will upload data set , preprocess data, divide data in to test and train and then create model and predict from the test data and predicted result is displayed to admin and sent to requested user.

c) Dataset Collection

In this step data set is collected from Kaggle website. Data set has features and labels. Features are used as input and labels for output.

3.2 Data Preprocessing

In this step data is pre processed by removing unwanted data and NAN values and using features and labels which are useful to fit in to algorithm and then process data for prediction.

a) Data split Test training

In this stage data is divided in to test and train values using train test split function and store features and labels in to test train values. Train set is 30 percent of test set data which is used for checking accuracy of the dataset.

b) Model Training

In this stage different algorithms are used to check which algorithm provides best accuracy and select one algorithm to use that for fitting features and labels and then run algorithm in this way model is trained.

c) Prediction and accuracy

In this stage new input or test set is taken as input and given as input to predict function of the algorithm and then result of labels are as output of the algorithm.

3.3 Input Design

The input design is a component of the overall system design. The following is the main goal of the input design:

1. Create a low-cost input method.
2. To achieve the highest possible level of accuracy.
3. Ascertain that the user accepts and comprehends the input.

3.4 Output Design

Computer system outputs are primarily used to communicate the results of processing to users. They are also used to save a permanent copy of the results for future reference. In general, the various types of outputs are as follows:

- External Outputs, which have a destination outside of the organization
- Internal outputs have a destination within the organization and serve as the primary interface between the user and the computer.
- Operational outputs that are only used within the computer department.
- Interface outputs that engage the user in direct communication.

4. IMPLEMENTATION

a) Design

The software system design is produced from the results of the requirements phase. Architects have the ball in their court during this phase and this is the phase in which their focus lies. This is where the details on how the system will work is produced. Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.

b) Implementation

Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle. For a developer, this is the main focus of the life cycle because this is where the code is produced. Implementation my overlap with both the design and testing phases. Many tools exists (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

5. TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole.

So in a nutshell, that is a very basic overview of the general software development life cycle model. Now let's delve into some of the traditional and widely used variations.

5.1 System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

5.2 Module Testing

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

5.3 Integration Testing

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing

results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

5.4 Acceptance Testing

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

6. OUTPUT SCREENS

Figure: Home Page

7. CONCLUSION

The current deep learning methods in the network traffic classification research don't make full use of the network traffic structured information. Drawing on the application methods of deep learning in the field of natural language processing, we propose a novel model BAT-MC via the two phase's learning of BLSTM and attention on the time series features for intrusion detection using NSL-KDD dataset. BLSTM layer which connects the forward LSTM and the backward LSTM is used to extract features on the the traffic bytes of each packet. Each data packet can produce a packet vector. These packet vectors are arranged to form a network flow vector. Attention layer is used to perform feature learning on the network flow vector composed of packet vectors. The above feature learning process is automatically completed by deep neural network without any feature engineering technology.

7.1 Future Enhancements

The developed system faces difficulties in classifying faces covered by hands since it almost looks like the person wearing a mask. While any person without a face mask is traveling on any vehicle, the system cannot locate that person correctly. For a very densely populated area, distinguishing the face of each person is very difficult. For this type of scenario, identifying people without face mask would be very difficult for our proposed system. In order to get the best result out of this system, the city must have a large number of CCTV cameras to monitor the whole city as well as dedicated manpower to enforce proper laws on the violators. Since the information about the violator is sent via SMS, the system fails when there is a problem in the network.

REFERENCES

[1] B. B. Zarpelo, R. S Miani, C. T. Kawakani, and S. C. de Alvarenga, ''A survey of intrusion detection in Internet of Things,'' J. Netw. Comput. Appl., vol. 84, pp. 25–37, Apr. 2017.

[2] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, ''Network intrusion detection,'' IEEE Netw., vol. 8, no. 3, pp. 26–41, May 1994.

[3] S. Kishorwagh, V. K. Pachghare, and S. R. Kolhe, ''Survey on intrusion detection system using machine learning techniques,'' Int. J. Control Automat., vol. 78, no. 16, pp. 30–37, Sep. 2013.

[4] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, ''Survey on SDN based network intrusion detection system using machine learning approaches,'' Peer-to-Peer Netw. Appl., vol. 12, no. 2, pp. 493–501, Mar. 2019.

[5] M. Panda, A. Abraham, S. Das, and M. R. Patra, ''Network intrusion detection system: A machine learning approach,'' Intell. Decis. Technol., vol. 5, no. 4, pp. 347–356, 2011.

[6] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, ''A new intrusion detection system based on KNN classification algorithm in wireless sensor network,'' J. Electr. Comput. Eng., vol. 2014, pp. 1–8, Jun. 2014.

[7] S. Garg and S. Batra, ''A novel ensembled technique for anomaly detection,'' Int. J. Commun. Syst., vol. 30, no. 11, p. e3248, Jul. 2017.

[8] F. Kuang, W. Xu, and S. Zhang, ''A novel hybrid KPCA and SVM with GA model for intrusion detection,'' Appl. Soft Comput.., vol. 18, pp. 178–184, May 2014