ANOMALY DETECTION IN NETWORK TRAFFIC USING UNSUPERVISED MACHINE LEARNING APPROACH

[1]Kichagari Sruthi, [2]B. Charishma
[1]PG Scholar, Department of CSE, Srinivasa Institute of technology and science, Kadapa.
[2]Associate Professor, HOD of CSE, Srinivasa Institute of technology and science, Kadapa.
[1] kichagarisruthi@gmail.com, [2]charishma.bavirisetty@gmail.com

ABSTRACT
Intrusion detection can identify unknown attacks from network traffics and has been an effective means of network security. Nowadays, existing methods for network anomaly detection are usually based on traditional machine learning models, such as KNN, SVM, etc. Although these methods can obtain some outstanding features, they get a relatively low accuracy and rely heavily on manual design of traffic features, which has been obsolete in the age of big data. To solve the problems of low accuracy and feature engineering in intrusion detection, a traffic anomaly detection model is proposed. The model combines LINEAR REGRESSION, attention mechanism. It can well describe the network traffic behaviour and improve the ability of anomaly detection effectively. We test our model on a public benchmark dataset, and the experimental results demonstrate our model has better performance than other comparison methods.
Keywords: Anomaly Detection, Isolation Forest, Machine Learning, Intrusion Detection System, Linear Regression, KNN, SVM.

I. INTRODUCTION

In this paper Network viruses, eavesdropping and malicious attacks are on the rise, causing network security to become the focus of attention of the society and government departments. Fortunately, these problems can be well solved via intrusion detection. Intrusion detection plays an important part in ensuring network information security. However, with the explosive growth of Internet business, traffic types in the network are increasing day by day, and network behaviour characteristics are becoming increasingly complex, which brings great challenges to intrusion detection [1], [2].

Neural network with traffic data as image. This method does not need manual design features, and directly takes the original traffic as the input data to the classifier. In [10], the authors provide an analysis of the viability of recurrent neural networks (RNN) to detect the behaviour of network traffic by modelling it as a sequence of states that change over time. In [11], pacifically, network traffic is a traffic unit composed of multiple data packets. Data packet is a traffic unit composed of multiple bytes. Secondly, traffic features in the same and different packets are significantly different. Sequential features between different packets need to be extracted independently. In other words, not all traffic features are equally important for traffic classification in the process of extracting features on a certain network traffic.

II. LITERATURE SURVEY

[1] 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. 2020 [1]. In this survey, we reviewed various recent works on machine learning (ML) methods that leverage SDN to implement NIDS. More specifically, we evaluated the techniques of deep learning in developing SDN-based NIDS. In the meantime, in this survey, we covered tools that can be used to develop NIDS models in SDN environment. This survey is concluded with a discussion of ongoing challenges in implementing NIDS using ML/DL and future works.

[2] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, ''Deep learning with long short-term memory for time series prediction,'' IEEE Commun. Mag., vol. 57, no. 6, pp. 114–119, Jun. 2019. In this article, we first give a brief introduction to the structure and forward propagation mechanism of the LSTM model. Then, aiming at reducing the considerable computing cost of LSTM, we put forward the Random Connectivity LSTM (RCLSTM) model and test it by predicting traffic and user mobility in telecommunication networks. Compared to LSTM, RCLSTM is formed via stochastic connectivity between neurons, which achieves a significant breakthrough in the architecture formation of neural networks.

[3] K. Wu, Z. Chen, and W. Li, ''A novel intrusion detection model for a massive network using convolutional neural networks,'' IEEE Access, vol. 6, pp. 50850–50859, 2018. In this paper, we propose a novel network intrusion detection model utilizing convolutional neural networks (CNNs). We use CNN to select traffic features from raw data set automatically, and we set the cost function weight coefficient of each class based on its numbers to solve the imbalanced data set problem. The model not only reduces the false alarm rate (FAR) but also improves the accuracy of the class with small numbers. To reduce the calculation cost further, we convert the raw traffic vector format into image format [6].

III. 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 [4].

i) Existing System

- In existing methods analysis of the viability of Recurrent Neural Networks (RNN) to detect the behavior of network traffic by modeling it as a sequence of states that change over time.
- In existing methods verify the performance of Long Short-term memory (LSTM) network in classifying intrusion traffics. Experimental results show that LSTM can learn all the attack classes hidden in the training data.

Disadvantages of Existing System

- All the above methods treat the entire network traffic as a whole consisting of a sequence of traffic bytes. They don't make full use of domain knowledge of network traffics.
- Existing methods treats traffics as independent and ignore the internal relations of network traffics.

ii) Proposed System

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

- We propose an end-to-end deep learning model logistic regression that is composed of logistic regression and attention mechanism. logistic regression can well solve the problem of intrusion detection and provide a new research method for intrusion detection
- We compare the performance of logistic regression with traditional deep learning methods, the model can extract information from each packet. By making full use of the structure information of network traffic, the logistic regression model can capture features more comprehensively. 4) We evaluate our proposed network with a real NSL-KDD dataset. The experimental results show that the performance of algorithm is better than the traditional method.

Advantages

- This method is used to analyses the important degree of packet vectors to obtain fine-grained features which are more salient for malicious traffic detection.
- At the output layer, the features generated by attention mechanism are then imported into a fully connected layer for feature fusion, which obtains the key features that accurately characterize network traffic behavior.

IV. SYSTEM DESIGN

i) Data Preprocessing Layer

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].

ii) Normalization Processing

The value of the original data may be too large, resulting in problems such as ‘‘large numbers to eat decimals’’, data processing overflows, and inconsistent weights so on. We use standard scaler to normalize the continuous data into the range [0, 1]. Normalization processing eliminates the influence of the measurement unit on the model training, and makes the training result more dependent on the characteristics of the data itself.

V. IMPLEMENTATION

i) 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.

ii) 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 exist (CASE tools) to actually automate the production of code using information gathered and produced during the design phase [15].

iii) Input Design

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

iv) Input Stages

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

v) Input Types

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system.
- Interactive, which are inputs entered during a dialogue.

vi) Input Media

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

vii) Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization.
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

viii) Output Stages

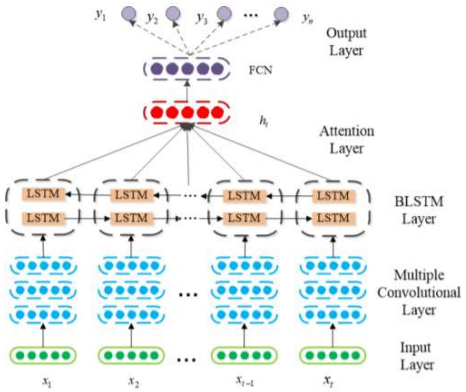
The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

ix) Architecture Analysis

Structured project management techniques (such as an SDLC) enhance management's control over projects by dividing complex tasks into manageable sections. In system design the main target is on distinguishing the modules, whereas throughout careful style the main target is on planning the logic for every of the modules. Here first we collect the data sets and process the data and we remove if there are any impurities in the data sets. Next the data is normalized if needed like it can be converted to smaller volume of data. Next the data is converted to supporting format. And then it is stored in the databases. Next the required method is applied. Now we get the final results.



Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Figure: System Architecture

x) Functional requirements

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization.
- Internal Outputs whose destination is within organization.
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.
- Understanding user's preferences, expertise level and his business requirements through a friendly questionnaire.
- Input data can be in four different forms - Relational DB, text files, .xls and xml files. For testing and demo, you can choose data from any domain. User-B can provide business data as input.

xi) Non-Functional Requirements

- Secure access of confidential data (user's details). SSL can be used.
- 24 X 7 availability.
- Better component design to get better performance at peak time.
- Flexible service-based architecture will be highly desirable for future extension.

VI. RESULTS

i) Test Cases

TABLE: Test Cases

Vol. 21, No. 1, (2024)
ISSN: 1005-0930

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Upload the tasks dataset	Verify either file is loaded or not	If dataset is not uploaded	It cannot display the file loaded message	File is loaded which displays task waiting time	High	High
02	Upload dataset and preprocess	Verify either dataset loaded or not and preprocessed	If dataset is not uploaded and preprocessed	It cannot display dataset reading process completed	It can display dataset reading process completed	low	High
03	Preprocessing	Whether preprocessing on the dataset applied or not	If not applied	It cannot display the necessary data for further process	It can display the necessary data for further process	Medium	High
04	Prediction model	Whether Prediction algorithm applied on the data or not	If not applied	Algorithm model is created	Algorithm model is created	High	High
05	Prediction	Whether predicted data is displayed or not	If not displayed	It cannot view prediction containing fraud in NSL-KDD dataset	It can view prediction containing fraud in NSL-KDD dataset	High	High

VII. CONCLUSION

The current machine 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 Linear Regression & 3 Layer Neural Network and attention on the time series features for intrusion detection using NSL-KDD dataset. 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.

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. 2020.

[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. 2020.
[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 2019.
[9] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, ‘‘Malware traffic classification using convolutional neural network for representation learning,’’ in Proc. Int. Conf. Inf. Netw. (ICOIN), 2017, pp. 712–717.
[10] P. Torres, C. Catania, S. Garcia, and C. G. Garino, ‘‘An analysis of Recurrent Neural Networks for Botnet detection behavior,’’ in Proc. IEEE Biennial Congr. Argentina (ARGENCON), Jun. 2016, pp. 1–6.

Vol. 21, No. 1, (2024)
ISSN: 1005-0930