JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Implementing Deep Learning for Intrusion Detection with NIDS-CNN LSTM

Mrs P.Navya[1], G.Anil Reddy [2], G.Koushik [3] R.Lohith Sai [4]
Assistant Professor[1], Student/Research Scholar[2], Student/Research Scholar[3], Student/Research Scholar[4] Department of Artificial Intelligence and Machine Learning, Vignana Bharathi Institute of Technology, Hyderabad, India

Abstract: Due to the central focus of network security research is intrusion detection, and deep learning-based intrusion detection algorithms have emerged as prominent research topics in this field. To efficiently separate and identify network traffic data and guarantee the security of the IIoT's equipment and operation, a deep learning-based network intrusion detection classification model (NIDS-CNNLSTM) is built for the wireless sensing scenario of the Industrial Internet of Things (IIoT) in this paper. With the help of time series data and the potent learning capabilities of long short-term memory neural networks, NIDS-CNNLSTM learns and categorizes the features chosen by the convolutional neural network and validates its applicability using scenarios involving binary and multi-classification. The KDD CUP99, NSL_KDD, and UNSW_NB15 classic datasets are used to train the model. As the three datasets' verification accuracy, training loss, and accuracy rate all exhibit good convergence and level, and the accuracy rate while classifying different types of traffic is high. The models suggested in earlier research have not been able to match the overall performance of NIDS-CNNLSTM. Based on experimental data, the effectiveness demonstrates a low false alarm rate, a high detection rate, and classification accuracy. Large-scale, multi-scenario network data in the IIoT is better suited for it.

Index Terms **– Network Security, Industrial Internet of Things, Convolutional Neural Networks, Long Short-Term Memory.**

I. INTRODUCTION

People can now access a variety of useful services on the Internet thanks to advancements in technology. We do, however, also face some security risks. Because of the increase in network infections, eavesdropping, and malicious assaults, government agencies and society as a whole are paying more attention to network security. Fortunately, intrusion detection is a good way to handle these issues. To ensure the security of network information, intrusion detection is crucial. However, as Internet commerce grows at an exponential rate, more and more different forms of traffic are entering networks, and the behavior characteristics of these networks are getting more complicated, which presents significant hurdles for intrusion detection. One major issue that cannot be avoided is how to recognize different types of malicious network traffic, especially unanticipated hostile network traffic. In actuality, there are two types of network traffic: harmful and regular traffic. Additionally, there are five categories into which network traffic can be separated: Standard, R2L (Root to Local), U2R (User to Root), DoS (Denial of Service), and Probe (Probing) attacks. Therefore, one may think of intrusion detection as a classification problem. Based on experimental findings, LSTM can identify every assault class that is concealed in the training set. A data packet is a unit of transmission made up of many bytes. Second, there are notable differences between traffic features in the same packet and distinct packets. It is necessary to extract sequential information between several packets individually. Stated differently, not every traffic feature holds identical significance for traffic classification during the feature extraction process of a given network traffic.

II. LITERATURE SURVEY

A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks by [1] Chuanlong Yin: This research investigates the use of recurrent neural networks (RNNs) in deep learning for intrusion detection systems (IDS). The effectiveness of the suggested model can be affected by the number of neurons and various learning rates, the authors discover. They also discovered that binary and multiclass classification are used to study the model's performance.

A bit-vector algorithm, Myers pattern matching of signature-based IDS using Myers algorithm under MapReduce framework by [2] Moath Jarrah: Using the Myers algorithm and the MapReduce architecture, this paper provides a bit-vector approach for signature-based intrusion detection systems. The approach is found to be four times faster than the serial version when it is parallelized on a multi-core CPU, according to the authors. Lower in-memory sharing in the model, however, may result in poorer performance

A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks by [3] Kehe We: This study provides a bit-vector approach based on the Myers algorithm for signature-based intrusion detection systems within the MapReduce framework. The approach is found to be four times faster than the serial version when it is parallelized on a multi-core CPU, according to the authors. Lower in-memory sharing in the model, however, may result in poorer performance.

Efficacy of Heterogeneous Ensemble Assisted Machine Learning Model for Binary and Multi-Class Network Intrusion Detection by [4] Dr. Arun Khosi: The study presents a novel approach to network intrusion detection that combines many machine learning techniques. We refer to this method as heterogeneous ensemble learning. The new model is said to be more accurate than earlier models in the paper, particularly for multi-class intrusion detection.

Machine Learning Techniques for Classifying Network Anomalies and Intrusions Algorithm by [5] Zhida Li: The study covers machine learning methods for categorizing intrusions and abnormalities in networks. Support Vector Machine (SVM), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Broad Learning System (BLS) are among the algorithms that the authors compare. They discover that while all of the algorithms work well, SVM and BLS produce the best outcomes.

METHODOLOGY

A. Data Collection:

The first step in this process is to collect the necessary data, train and test the NIDS-CNNLSTM model, and compile a sizable collection of network traffic data.

B. Data Preprocessing:

Preprocessing the data includes cleaning, standardizing, and converting it into a format that is appropriate for training models.

C. Model Architecture:

Convolutional neural networks (CNN) and long short-term memory (LSTM) networks are combined in the NIDS-CNNLSTM model architecture to enable efficient intrusion detection

D. Training:

Once Using the preprocessed dataset, train the NIDS-CNNLSTM model, modifying hyperparameters to maximize performance.

Evaluation:

To determine how well the model performs in classifying network intrusions by utilizing metrics such as accuracy, precision, recall, and F1-score. Analyze the NIDS-CNNLSTM model's performance against that of other intrusion detection systems, emphasizing both of its advantages and disadvantages.

In conclusion, Examined the outcomes of the model's assessment and training to make insightful judgments regarding its effectiveness in detecting network intrusions. Talk about the results' ramifications, possible uses for the NIDS-CNNLSTM model, and directions for further study and development.

IMPLEMENTATION

A sample dataset was fetched from Kaggle.

duration	protocol_t	service	flag	src_bytes	dst_bytes	land	wrong_fra	urgent	hot	num_faile	logged_in	num_com	root_shell	su_attemp	num_root	num_file_	num_shell	num_acce	num_outb	is_host_lo	is_guest_l	count	sr
0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	
0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	
0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	
0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	30	
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	121	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	
0	tcp	remote_jo	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	270	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	205	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	
0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	
0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	
0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	233	
0	tcp	netbios_n	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96	
0	tcp	http	SF	300	13788	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	

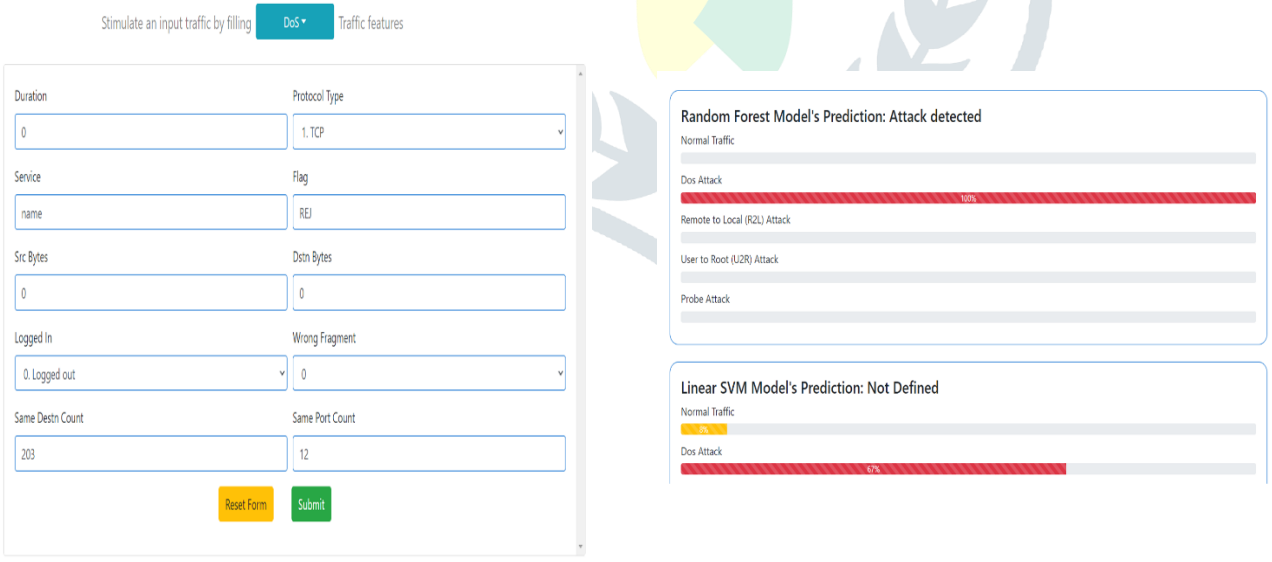
Figure. 1. Sample Dataset

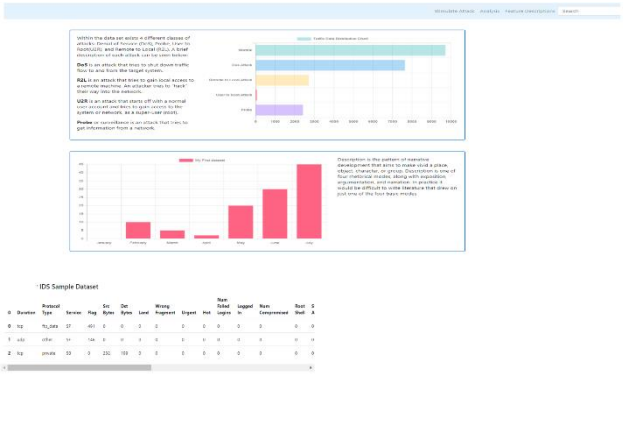
will be used for training and testing. In total, around 25000 training samples will be taken, After the training phase is completed, a testing dataset is used, which will have 22000 resumes to test our system.

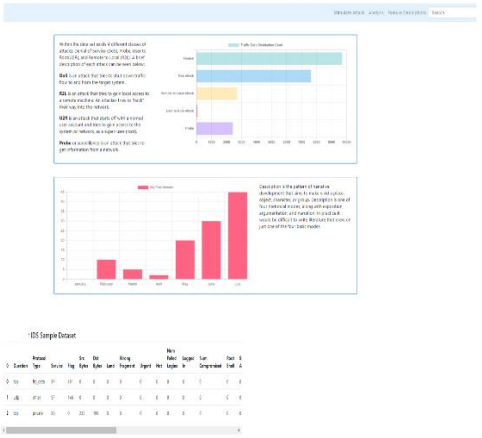
Training phase:
In the initial phase of dataset processing, a dataset of samples is utilized for training, Utilizing the training data, train the NIDS-CNNLSTM model. To minimize the specified loss function, batches of input data must be fed into the model, and iterative adjustments must be made to the model's parameters (weights and biases).To avoid overfitting, keep an eye on the training procedure using the validation data. If the performance no longer improves on the validation set, training can be stopped early using approaches like early stopping. To maximize the performance of the model, experiment with various hyperparameters, including learning rate, batch size, and dropout rate.
.
Modules:
Load the data: Pandas facilitates importing data from various sources like CSV, Excel, JSON, databases (MySQL, PostgreSQL), and web scraping into data frames. It offers a versatile approach for handling data, enabling seamless integration from static files to databases, and empowering efficient data manipulation and analysis.
Data collection:
Data collection involves gathering information from various sources, both online and offline, which is essential for deep learning projects. It's crucial for building accurate predictive models by identifying recurring patterns and trends. Effective data collection.
Data pre-processing: Data preprocessing is essential, as raw data often contains noise, missing values, and formatting issues. It involves cleaning and formatting data to make it suitable for analysis and improving accuracy. Utilizing the training data, train the NIDS-CNNLSTM model. To minimize the specified loss function, batches of input data must be fed into the model, and iterative adjustments must be made to the model's parameters (weights and biases).
To avoid overfitting, keep an eye on the training procedure using the validation data. If the performance no longer improves on the validation set, training can be stopped early using approaches like early stopping.
To maximize the performance of the model, experiment with various hyperparameters, including learning rate, batch size, and dropout rate. TensorFlow, also known as PyTorch: TensorFlow, also known as PyTorch, is the main deep learning framework used to construct and train neural networks. The tools required to define network architectures, compute on tensors, and optimize model parameters during training are provided by these frameworks. Keras is a high-level API for neural networks that may be used with Microsoft Cognitive Toolkit (CNTK), TensorFlow, or Theano. It makes it simpler to design complex architectures and oversee training procedures by offering an intuitive interface for creating and refining deep learning models.NumPy: NumPy is a core package for Python scientific computing. Support for multidimensional arrays and mathematical functions is offered, which is necessary for preprocessing and data manipulation activities related to getting the dataset ready for training.

Feature extraction, a component of dimensionality reduction, partitions raw data into manageable groups to ease processing. It selects and combines variables into features, reducing the computational burden. Color features, derived from image histograms, offer succinct descriptions of color statistics, maintaining accuracy and originality in large datasets.

RESULTS AND DISCUSSION







CONCLUSION:

In conclusion, the strain on network intrusion detection is rising as network intrusion continues to change. Cyberspace security is seriously threatened, in part because of issues with unbalanced network traffic that make it hard for intrusion detection systems to forecast the path of hostile attacks.

ACKNOWLEDGMENT

We would like to thank Mrs P.Navya for her valuable comments and suggestions to improve the quality of the paper. We are also grateful to Mrs P.Navya for helping us review our work regularly. We would also like to thank the Department of Computer Science Engineering (AI and ML), VBIT Hyderabad for providing us with Jetson Nano Boards and all the needed help.

REFERENCES

[1] Chuanlong Yin, Ashwina Pereira, A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks.

[2] Moath Jarrah., A bit-vector algorithm, Myers pattern matching of signature-based IDS using Myers algorithm under MapReduce framework.

[3] Kehe We, A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks.

[4] Dr. Arun Khosi, Efficacy of Heterogeneous Ensemble Assisted Machine Learning Model for Binary and Multi-Class Network Intrusion Detection.

[5] Zhida Li, Machine Learning Techniques for Classifying Network Anomalies and Intrusions.

[6] N.B.Amor,S.Benferhat, and.Elouedi, Naive Bayes vs decision trees in the intrusion detection system.

[7] D. A. Cieslak, N. V. Chawla, and A. Striegel, Combating imbalance in network intrusion datasets.

[8] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, ‘‘Deep learning for visual understanding: A review.

[9] N. Japkowicz, ‘‘The class imbalance problem: Significance and strategies,’’ in Proc.

[10] M. A. M. Hasan, M. Nasser, B. Pal, and S. Ahmad, ‘‘Support vector machine and random forest modeling for intrusion detection system (IDS).