IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks

1. J.DELPHIN, 2.Dr.M.JANAKI

1. M.Phil Scholar, PG Department of Computer Science, Dr. Umayal Ramanathan College for Women, Karaikudi, Tamilnadu, India
2. Associate Professor, PG Department of Computer Science, Dr.Umayal Ramanathan College for Women, Karaikudi, Tamilnadu, India

Abstract—Intrusion Detection System (IDS) is meant to be a software application which monitors the network or system activities and finds if any malicious operations occur. Tremendous growth and usage of internet raises concerns about how to protect and communicate the digital information in a safe manner. Nowadays, hackers use different types of attacks for getting the valuable information. As the internet emerging into the society, new stuffs like viruses and worms are imported. The malignant so, the users use different techniques like cracking of password, detecting unencrypted text are used to cause vulnerabilities to the system. Hence, security is needed for the users to secure their system from the intruders. Firewall technique is one of the popular protection techniques and it is used to protect the private network from the public network. IDS are used in network related activities, medical applications, credit card frauds, Insurance agency. Many intrusion detection techniques, methods and algorithms help to detect these attacks. This main objective of this project is to provide a comparative study about intrusion detection using hybrid algorithm such as CNN with LSTM have been used to develop Ids in real time network datasets such as Intrusion Detection System (IDS) datasets and UNSW datasets. This algorithm is widely used neural network classifier based on number of classes (output) and number of hidden layers, uses weights for every node at neural network, most effective attributes will get large weights conversely attributes not affect in predictive class. The proposed system can be analyzed in terms of error rate and accuracy values and implement in python tool for performance analysis.

Index Terms—.***Dataset collection, preprocessing, feature extraction, classification, intrusion detection***

I. INTRODUCTION

A network packet is a formatted unit of data carried by a packet-switched network. The physical link technologies of packet network typically limit the size of

packets to a certain maximum transmission unit (MTU). A longer message is fragmented before it is transferred and once the packets arrive, they are reassembled to construct the original message. Packets consist of two types of data: control information and user data (payload). The control information provides data the network needs to deliver the user data, for example, source and destination network addresses, error detection codes, and sequencing information. Typically, control information is found in packet headers and trailers, with payload data in between. With packets, the bandwidth of the transmission medium can be better shared among users than if the network were circuit switched. When one user is not sending packets, the link can be filled with packets from other users, and so the cost can be shared, with relatively little interference, provided the link isn't overused. Often the route a packet needs to take through a network is not immediately available. In that case, the packet is queued and waits until a link is free. A communication protocol is a set of rules for exchanging information over a network. In a protocol stack (also see the OSI model), the protocol is divided up into layers, where each protocol layer leverages the services of the protocol layer below it until the lowest layer controls the hardware that sends information across the media. The use of protocol layering is today ubiquitous across the field of computer networking. An important example of a protocol stack is HTTP (the World Wide Web protocol) running over TCP over IP (the Internet protocols) over IEEE 802.11 (the Wi-Fi protocol). This stack is used between the wireless router and the home user's personal computer when the user is surfing the web. This project is develop the approach aims to leverage the complementary strengths of CNNs in spatial feature extraction and LSTMs in modeling temporal dependencies. By integrating these two deep learning architectures, the system endeavors to achieve a more comprehensive and accurate detection of anomalous patterns within network traffic data. Specifically, the CNN component of the proposed system is designed to analyze the spatial characteristics of network traffic, effectively identifying complex and subtle patterns that may indicate malicious activity. CNNs excel at capturing local patterns and spatial relationships within data, making them well-suited for detecting irregularities in network traffic structures. Meanwhile, the LSTM component focuses on capturing the temporal dynamics inherent in network traffic data. By modeling the sequential nature of network events over time, LSTMs can effectively recognize patterns that evolve and unfold gradually, enabling the system to detect sophisticated intrusion attempts that may span across multiple network packets or sessions. Through the integration of CNNs and LSTMs, the project aims to enhance the overall performance and accuracy of intrusion detection, enabling system administrators to proactively identify and respond to security threats in real-time. By providing a robust and adaptive defense mechanism against network intrusions, the proposed system ultimately seeks to bolster the cybersecurity posture of organizations and safeguard their critical assets and resources from potential cyber threats.

II. LITERATURE SURVEY

ShadiAljawarneh, et.al,…[1]Efficiently detecting network intrusions requires the gathering of sensitive information. This means that one has to collect large amounts of network transactions including high details of

recent network transactions. Assessments based on meta-heuristic anomaly are important in the intrusion related network transaction data's exploratory analysis. These assessments are needed to make and deliver predictions related to the intrusion possibility based on the available attribute details that are involved in the network transaction. We were able to utilize the NSL-KDD data set, the binary and multiclass problem with a 20% testing dataset. This paper develops a new hybrid model that can be used to estimate the intrusion scope threshold degree based on the network transaction data's optimal features that were made available for training. The experimental results revealed that the hybrid approach had a significant effect on the minimization of the computational and time complexity involved when determining the feature association impact scale. The accuracy of the proposed model was measured as 99.81% and 98.56% for the binary class and multiclass NSL-KDD data sets, respectively. Intrusion detection systems (IDS) are generally divided into two types (see Fig. 1): misuse and anomaly intrusion detection systems. For a misuse IDS, instructions are identified based on parameters of system weaknesses and known attack signatures. However, it does not recognised attacks that are new or unfamiliar. On the other hand, anomaly IDS is based on normal behaviour parameters and utilizes them to pinpoint any action that deviates significantly from normal behaviour. The misuse intrusion detection mechanism identifies intrusions by matching existing intrusion patterns in consideration for examination with previously identified patterns.

Nasrin Sultana, et.al,…[2]Network Intrusion Detection systems (NIDS) have been developed rapidly in academia and industry in response to the increasing cyber-attacks against governments and commercial enterprises globally. The annual cost of cybercrime is continuously rising. Organizations can lose their intellectual property with such malicious software crept into the system which may lead to disruptions to a country's critical national infrastructure. Organizations deploy a firewall, antivirus software, and an intrusion detection system (NIDS) to secure computer systems from unauthorized access. Software Defined Networking Technology (SDN) provides a prospect to effectively detect and monitor network security problems ascribing to the emergence of the programmable features. Recently, Machine Learning (ML) approaches have been implemented in the SDN-based Network Intrusion Detection Systems (NIDS) to protect computer networks and to overcome network security issues. A stream of advanced machine learning approaches – the deep learning technology (DL) commences to emerge in the SDN context. 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. Software-defined network is an emerging architecture that decouples network control and forwarding functions so that the network control can be directly programmable. The segregation of the control plane from the data plane enables easy network management.

Kai Peng, et.al,…[3]Intrusion detection system (IDS) provides an

important basis for the network defence. Due to the development of the cloud computing and social network, massive amounts of data are generated, which inevitably brings much pressure to IDS. And therefore, it becomes crucial to efficiently divide the data into different classes over big data according to data features. Moreover, we can further determine whether one is normal behaviour or not based on the classes information. Although the clustering approach based on K-means for IDS has been well studied, unfortunately directly using it in big data environment may suffer from inappropriateness. On the one hand, the efficiency of data clustering needs to be improved. On the other hand, differ from the classification, there is no unified evaluation indicator for clustering issue, and thus, it is necessary to study which indicator is more suitable for evaluating the clustering results of IDS. In this paper, we propose a clustering method for IDS based on Mini Batch K-means combined with principal component analysis. First, a pre-processing method is proposed to digitize the strings and then the data set is normalized so as to improve the clustering efficiency. Second, the principal component analysis method is used to reduce the dimension of the processed data set aiming to further improve the clustering efficiency, and then mini batch K-means method is used for data clustering. More specifically, we use K-means++ to initialize the centres of cluster in order to avoid the algorithm getting into the local optimum, in addition, we choose the CalsskiHarabasz indicator so that the clustering result is more easily determined. Compared with the other methods, the experimental results and the time complexity analysis show that our proposed method is effective and efficient. Above all, our proposed clustering method can be used for IDS over big data environment.

FahimehFarahnakian, et.al,…[4]In recent years, significant research has been focused on developing Intrusion Detection Systems (IDSs) to improve software and system security. Generally, IDSs can be divided into two main categories: misuse-based IDSs and anomaly based IDSs. Misuse-based IDSs detect known attacks based on the predetermined signature. Therefore, dynamic signature updating is so important and new attack definitions are frequently released by IDS vendors. However, the misuse based IDS cannot incorporate the rapidly growing number of vulnerabilities and exploits. Anomaly-based IDSs are designed to capture any deviation from profiles of normal behaviour. Therefore, they are more suitable than misuse-based detection systems for detecting unknown or novel attacks without any prior knowledge. One of the most challenging problems facing network operators today is network attacks identification due to extensive number of vulnerabilities in computer systems and creativity of attackers. To address this problem, we present a deep learning approach for intrusion detection systems. Our approach uses Deep Auto-Encoder (DAE) as one of the most well-known deep learning models. The proposed DAE model is trained in a greedy layer-wise fashion in order to avoid over fitting and local optima. The experimental results on the KDD-CUP'99 dataset show that our approach provides substantial improvement over other deep learning-based approaches in terms of accuracy, detection rate and false alarm rate.

Mohamed Idhammad, et.al,…[5]Intrusion and attack tools have become more sophisticated challenging

existing Cloud IDSs by large volumes of network traffic data, dynamic and complex behaviours and new types of attacks. It is clear that IDS for Cloud should analyse large volumes of network traffic data, detect efficiently the new attack behaviours and reach high accuracy with low false. However pre-processing, analysing and detecting intrusions in Cloud environments using traditional techniques have become very costly in terms of computation, time and budget. However, many security issues arise with the transition to this computing paradigm including intrusions detection. Regardless the important evolution of the information security technologies in recent years, intrusions and attacks continue to defeat existing intrusion detection systems in Cloud environment. Attackers developed new sophisticated techniques able to bring down an entire Cloud platform or even many within minutes. New records are breached each year by attacker. Recently a destructive DDoS attack has brought down more than 70 vital services of Internet including Github, Twitter, Amazon, Paypal, etc. Attackers have taken advantages of Cloud Computing and Internet of Things technologies to generate a huge amount of attack traffic; more than 665 Gb/s. tem is designed to be inserted in the Cloud side by side with the edge network components of the Cloud provider. This allows intercepting incoming network traffic to the edge network routers of the physical layer. A time-based sliding window algorithm is used to pre-process the captured network traffic on each Cloud router and pass it to an anomaly detection module using Naive Bayes classifier. A set of commodity server nodes based on Hadoop and MapReduce are available for each anomaly detection module to use when the network congestion increases.

III. BACKGROUND OF THE PROJECT

The IDS can be distinguished on the basis of where the detection is taking place and how or by which technique it is being detected. The IDS is classified into two niche segment one being Network Intrusion Detection System (NIDS) and the other being Host Intrusion Detection System (HIDS). The first system mentioned helps in the analysis the incoming networking traffic whereas the HIDS functioning is based on the activity of the operating system. The main aspects of data mining on IDS that were dealt with originally were termed as clustering and classification. Since there exist no label for the initial data set for clustering issue, the object created for the clustering algorithm was allocated the same class with similar data records. The behavior of the packet was termed as a normal class or abnormal class according to the features and characteristics of already existing data. In Classification, this works on mining from the already clustered data. This implies that the data is labelled. Classification is a data mining technique which is used for examining a data set. In this world of continuous streaming data, classification plays an important role in classifying the data. Many algorithms such as decision tree, rule-based induction, Bayesian network, genetic algorithm etc are used to classify the data. In existing framework implement, machine learning techniques such as Random forest, Naives Bayes, Support Vector machine algorithms are implemented to detect the intrusion from network datasets. In existing framework can be provide high false alarm and low accuracy.

IV. PROPOSED METHODOLOGY

Deep learning is learns features from the data. If large amount of data is

available, it can reduce the performance of system. For achieving better accuracy in terms of performance deep learning is well suited learning mechanism. Learning is varies in three major categories i.e. supervised, semi-supervised and unsupervised. Here, the intrusion detection is carried out with respect to the deep learning approach. Intrusion is the term that can violate security of computer system or network. And another is intrusion detection is the process to identify intrusion. Intrusion detection technique is classified in two methods i.e. anomaly detection or misuse detection. With the rapid expansion of computer networks during the past decade, security has become a crucial issue for computer systems. Hybrid based methods have been proposed in recent years for the development of intrusion detection systems. This project presents a neural network approach to intrusion detection. CNN with LSTM is used for intrusion detection based on an off-line analysis approach. While most of the previous studies have focused on classification of records in one of the two general classes - normal and attack, this research aims to solve a multi-class problem in which the type of attack is also detected by the neural network and it is a layered feed forward network typically trained with static back propagation (BP). Such networks have found their way into countless applications requiring static pattern classification. The hybrid algorithm is a flexible type for composed of one input layer, one or more hidden layers, and one output layer.

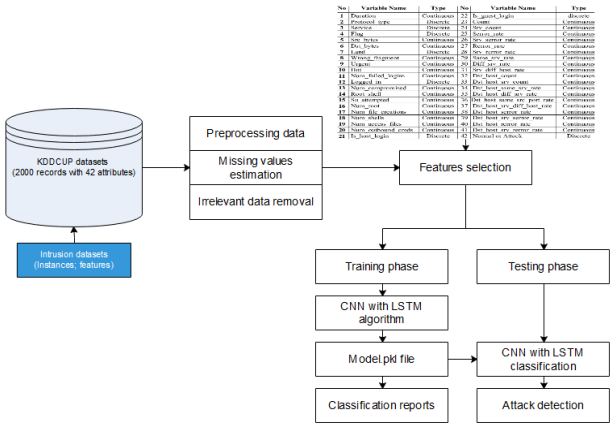

Fig 3: Architecture of Proposed Framework

DATASETS ACQUISITION

The KDD Cup dataset, utilized for benchmarking intrusion detection issues, is used in our experiments. The dataset is a gathering of simulated crude TCP dump data over a time of 9 weeks on a LAN. The training data was processed to about 5 million connections records from seven weeks of network traffic and two weeks of testing data yielded around 2 million connection records. And also upload the UNSW datasets. The raw network packets of the UNSW-NB 15 dataset was created by the IXIA Perfect Storm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviours. In this module, we can upload the network datasets in the form of CSV file.

PREPROCESSING

Data pre-processing is an important step in the [data mining] process. The phrase "garbage in, garbage out" is particularly applicable to data mining

and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations, missing values, etc. Thus, the representation and quality of data is first and foremost before running an analysis. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. In this module, eliminate the irrelevant and missing values in uploaded datasets.

FEATURES EXTRACTION

Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction. Determining a subset of the initial features is called feature selection. The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data. In this module, we can select the many attributes from pre-processed datasets**:**

CLASSIFICATION

This model aims to capture both spatial and temporal patterns in network traffic data. Initially, the data undergoes preprocessing, including feature extraction and partitioning into training and testing sets. Subsequently, CNN layers are employed to extract spatial features from the network data, treating it as a 1D image. These extracted features are then fed into LSTM layers to capture temporal dependencies in the sequence of network packets. By combining the outputs of the CNN and LSTM layers, spatial and temporal features are effectively merged. Finally, the combined representation undergoes classification using fully connected layers, enabling the system to identify potential intrusions with enhanced accuracy and reliability

PERFORMANCE EVALUATION

In this module, performance can be evaluated in terms of accuracy rate. Proposed work provide improved accuracy rate than the existing systems

CNN ALGORITHM

CNNs represent feed-forward neural networks whichencompass diverse combos of the convolutional layers, max pooling layers, and completely related layers and Take advantage of spatially neighborhood correlation by way of way of imposing a nearby connectivity pattern among neurons of adjacent layers. Convolutional layers alternate with max pooling layers mimicking the individual of complex and clean cells in mammalian seen cortex .A CNN includes one or extra pairs of convolution and max pooling layers and ultimately ends with completely related neural networks. In our proposed CNN structure, multiple features can be extracted from each original hyperspectral, and each feature has $n3$ dimensions

Constructing the CNN Model

function INITCNNMODEL (θ, [$n1$–5])

layerType = [convolution, max-pooling, fully-connected, fully-connected];

layerActivation = [tanh(), max(), tanh(), softmax()]

model = new Model();

for$i$=1 to 4 do

layer = new Layer();

layer.type = layerType[i];
layer.inputSize = ni
layer.neurons = new Neuron [ni+1];
layer.params = θi;
model.addLayer(layer);
end for
return model;
end function
Training the CNN Model
Initialize learning rate α, number of max iteration ITERmax, min error ERRmin, training batchsBATCHEStraining, bach size SIZEbatch, and so on;
Compute $n2$, $n3$, $n4$, $k1$, $k2$, according to $n1$ and $n5$;
Generate random weights θ of the CNN;
cnnModel = InitCNNModel(θ, [$n1$–5]);
iter = 0; err = +inf;
while err >ERRmin and iter<ITERmax do
err = 0;
forbach = 1 to BATCHEStraining do
[$\nabla\theta J(\theta)$, $J(\theta)$] = cnnModel.train (TrainingDatas, TrainingLabels), as (4) and (8); Update θ using (7);
err = err + mean($J(\theta)$);
end for err = err/BATCHEStraining;
iter++;
end while
Save parameters θ of the CNN.

V. RESULT AND DISCUSSION

The hybrid CNN-LSTM algorithm demonstrated promising results in intrusion detection, exhibiting superior performance compared to traditional methods. The model effectively learned spatial features through the CNN layers, capturing patterns in network traffic data akin to image recognition. Additionally, the LSTM layers successfully captured temporal dependencies, enabling the model to discern sequential patterns indicative of intrusion activities. Through comprehensive evaluation on benchmark datasets, the hybrid model achieved high accuracy and robustness in identifying various types of intrusions while minimizing false positives. Moreover, the model showcased scalability and adaptability, maintaining consistent performance across different network environments and traffic volumes. However, challenges such as computational complexity and parameter tuning were encountered, necessitating optimization strategies for real-time deployment. Overall, the hybrid CNN-LSTM algorithm presents a promising approach for enhancing intrusion detection systems, offering a potent combination of spatial and temporal analysis for improved security in network environments .

VI. CONCLUSION

In conclusion, the use of a hybrid algorithm combining Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks shows promise in enhancing intrusion detection systems. By leveraging the strengths of CNNs in extracting spatial features from network traffic data and LSTM's ability to capture temporal dependencies, the hybrid model can effectively detect complex patterns indicative of malicious activities. This approach offers several advantages, including improved accuracy in identifying both known and unknown threats, enhanced scalability to handle large volumes of network data, and better adaptability to evolving attack techniques. Additionally, the interpretability of the model can be enhanced by visualizing the learned features, providing insights into the underlying mechanisms of intrusion detection.

REFERENCES

[1] Aljawarneh, Shadi, Monther Aldwairi, and Muneer Bani Yassein. "Anomaly-

based intrusion detection system through feature selection analysis and building hybrid efficient model." Journal of Computational Science 25 (2018): 152-160.

[2] Sultana, Nasrin, et al. "Survey on SDN based network intrusion detection system using machine learning approaches." Peer-to-Peer Networking and Applications 12.2 (2019): 493-501.

[3] Peng, Kai, Victor CM Leung, and Qingjia Huang. "Clustering approach based on mini batch kmeans for intrusion detection system over big data." IEEE Access 6 (2018): 11897-11906.

[4] Farahnakian, Fahimeh, and Jukka Heikkonen. "A deep auto-encoder based approach for intrusion detection system." 2018 20th International Conference on Advanced Communication Technology (ICACT). IEEE, 2018.

[5] Idhammad, Mohamed, Karim Afdel, and Mustapha Belouch. "Distributed intrusion detection system for cloud environments based on data mining techniques." Procedia Computer Science 127 (2018): 35-41.

[6] Rustam, Zuherman, and Durrabida Zahras. "Comparison between support vector machine and fuzzy c-means as classifier for intrusion detection system." Journal of Physics: Conference Series. Vol. 1028. No. 1. IOP Publishing, 2018.

[7] Peng, Kai, et al. "Intrusion detection system based on decision tree over big data in fog environment." Wireless Communications and Mobile Computing 2018 (2018).

[8] Pham, Ngoc Tu, et al. "Improving performance of intrusion detection system using ensemble methods and feature selection." Proceedings of the Australasian Computer Science Week Multiconference. 2018.

[9] Ahmad, Iftikhar, et al. "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection." IEEE access 6 (2018): 33789-33795.

[10] Sahani, Roma, et al. "Classification of intrusion detection using data mining techniques." Progress in computing, analytics and networking. Springer, Singapore, 2018. 753-764.

.