International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Deep Learning Proactive System Security Based on CNN Algorithm

V.Krishna Reddy[1], S.Srihitha Reddy[2], S. Rajesh Reddy[3],T. Ravi Kiran[4]

[1] Assistant Professor, Department of Information Technology, J. B. Institute of Engineering & Technology

[2,3,4]Student, Department of Information Technology, J. B. Institute of Engineering &Technology

Abstract

Software-Defined Networking is a new architecture technique in the networking sector. This technology enables networks to be controlled centrally and intelligently via unified applications such as traffic categorization and security management. Conventional networks' static nature limits their ability to accommodate the business objectives of organisations. SDNs are evolving designs that provide novel solutions to a variety of networking difficulties. Nowadays, the bulk of this sort of security solution has been built utilising Machine Learning approaches. Deep Learning algorithms have lately been employed to improve accuracy and efficiency. This research provides a new detection technique based on Convolutional Neural Network (CNN) that detects multiple assaults with 100% accuracy and achieves a low degradation rate of 2.3% throughput and 1.8% latency when conducted in a cluster.

Keywords:Deep learning-early warning proactive system (DL-EWPS), convolutional neural network(CNN), software-defined networking (SDN), intrusion detection system (IDS), deep learning (DL), RGBimage, InSDN dataset.

1. Introduction

The security of contemporary communication networks, as well as the integrity and privacy of data, have become more important in recent years. An institution's most important requirement is to protect its own and sensitive data from both internal and external adversaries. Some authorised users may breach their company's data and send it to others for various causes. A continual live flow of data makes it difficult to identify an assault in real-time. Modern networks are made up of a combination of hardware and software entities that are designed to fulfil the needs of each organisation in an arbitrary manner to the observer. These components include dangers, flaws, and security constraints. The software's attack is complex, and it puts the data at danger. Log files may be used by the developer and programmer to ensure system security. The intricacy of contemporary communication networks renders them vulnerable to security breaches. The severity and speed of network assaults vary according to network factors. The fundamental problem in current network security is identifying and neutralising attacks efficiently and in real-time. As a result, the Intrusion Detection System (IDS) technology must be used to detect internal and external attackers and defend systems and networks. manuscript and approving it for publication Tawfik Al-Hadhrami may use log files to guarantee system security. The intricacy of contemporary communication networks renders them vulnerable to security breaches. The severity and speed of

network assaults vary according to network factors. The fundamental problem in current network security is identifying and neutralising attacks efficiently and in real-time. As a result, the Intrusion Detection System (IDS) technology must be used to detect internal and external attackers and defend systems and networks.

SDN allows the network to be controlled by software, resulting in fewer networking devices and simpler physical connectivity and configuration. As a result, network administrators may modify network behaviour to accommodate current services and security applications. As a result, the vast majority of network services will be more adaptable, programmable, and not constrained by a platform.

This paper we introduces the Deep Learning-Early Warning Proactive System, a novel intrusion detection system designed for current SDNs (DL-EWPS). DL-EWPS is a novel technique that uses the CNN classifier to identify multiple assaults throughout the SDN network in real-time. CNN is one of the Deep Learning algorithms used in image classification, document analysis, face recognition, and other tasks [1]. The model identifies different attacks efficiently, including Denial of Service (DoS), Distributed Denial of Service (DDoS), Probe, User to Root (U2R), and Remote to Local (R2L). In existing IDSs, the suggested approach decreases the controller's overhead. As a result, the suggested model provides a lightweight mechanism for SDNs to confirm that the current request has been finished and then make another request without taxing the controller. DL-EWP S also introduces a novel fast approach for converting numerical data to RGB visuals for use by CNN classifiers. In addition, we included new characteristics retrieved from the flow table data to improve the model's dependability. The new features were chosen after an examination and analysis of certain typical assault behaviours. During an assault, these qualities are extremely useful. The proposed system has been implemented in the SDN controller. When tested in a large-scale network, DL-EWPS can identify multiple assaults with 100% accuracy and achieve a low degradation rate of throughput and latency.

The suggested model obtained 99.86% prediction accuracy with a training:testing ratio of 6:4, which is deemed pretty high. Nevertheless, the hybrid model utilised enhanced the model's complexity, necessitating extra resources such as memory and CPU processing. Only DDoS assaults can be detected by the suggested IDS. Furthermore, the suggested model includes numerous extracted characteristics that need extra memory and a lengthy procedure. As a result, it will be a bottleneck for the controller and cannot be implemented in real-time in large-scale networks. The bulk of these extracted traits have nothing to do with DDoSattack techniques. [8] presents a less sophisticated model. The authors developed a Deep Learning (DL) solution for the SDN environment to detect DDoS and DoS assaults in real-time between the controller and end-user devices. This study's suggested model detects attack by utilising thestandard DL algorithm with Relu and Softmax functions.

The suggested model obtained 99.86% prediction accuracy with a training:testing ratio of 6:4, which is deemed pretty high. Nevertheless, the hybrid model utilised enhanced the model's complexity, necessitating extra resources such as memory and CPU processing. Only DDoS assaults can be detected by the suggested IDS. Furthermore, the suggested model includes numerous extracted characteristics that need extra memory and a lengthy procedure. As a result, it will be a bottleneck for the controller and cannot be implemented in real-time in large-scale networks. The bulk of these extracted traits have nothing to do with DDoSattack techniques. [8] presents a less sophisticated model. The authors developed a Deep Learning (DL) solution for the SDN environment to detect DDoS and DoS assaults in real-time between the controller and end-user devices.

Earlier they demonstrated a Deep Learning-based intrusion detection system capable of detecting a DDoS assault in an SDN context. There are three modules in the model: Traffic Collector and Flow Installer (TCFI), FeatureExtractor (FE), and Traffic Classifier (TC). The system examines each packet in the SDN controller, extracts the characteristics, and then sends the collected features to be classified as either regular or malignant. The authors gathered data from their home WiFi network (HWN). To produce DDoS traffic, they used the tcpdump and hping3 tools in this study. The suggested system harvests 68 characteristics to be utilised in packet classification at the Feature Extractor (FE) module. Unfortunately, it could only reach a 95.65% accuracy. Because this method is repeated for each packet, the quantity of retrieved characteristics necessitates a large amount of memory and a lengthy processing time. As a result, the controller experiences a bottleneck. Moreover, the majority of these extracted traits have nothing to do with DDoS attack methods. Moreover, the authors violated the principle of SDN architecture by configuring the controller to require all packets via the controller to be processed while ignoring the flow table function. As a result, when executing such a model in large-scale networks, the controller will crash, resulting in poor performance in tiny networks.

In contrast, a detection strategy was proposed using a Deep Neural Network (DNN) approach to identify DDoS assaults in SDN networks. For the training and testing rounds, the NSL-KDD public dataset was used. The suggested model classified using the six main features, and the controller collects these features in real time for each flow. Nonetheless, the minimal amount of retrieved characteristics resulted in a 75.75% detection accuracy. These attributes were retrieved from the flow's basic statistics information and are insufficient to cover the assaults' behaviours; hence, the attacker may easily bypass the IDS. The authors set up the controller to collect flow data from all OpenFlow switches sequentially at a set time. This procedure strains the controller, but not more than the model given in [9]. [11] presents a thorough IDS. The suggested model is a flow-based anomaly detection solution in the OpenFlow controller that employs Gated Recurrent Unit LongShort-Term Memory (GRU-LSTM DNN). To achieve a high-performance classification, an ANOVA F-TEST feature selection approach was applied. The NSL-KDD public dataset was used for the experiment's training and testing phases. The suggested model identifies several attacks such as Prop, U2R, R2L, and DoS.

The goal of the model is to define the DOS, U2R, R2L, and Probes. The detection rate of the model was 97.4%. Unfortunately, using a modest dataset size during the training step reduces the detection accuracy of the real test. The retrieved characteristics are solely taken from the headerpacket and do not cover the attack behaviours. Also, the dataset utilised has a very redundant record. The controller performed the processing function for each packet involved, resulting in a bottleneck and a single point of failure. Using Machine Learning (ML) techniques with IDS is a significant problem. Furthermore, the suggested model necessitates the use of a feature selection approach to choose the characteristics that are effective when an assault happens.

2. Proposed System

- We use the classic NSL-KDD and the up-to-date benchmark datasets and conduct detailed analysis and data cleaning. (2) This work proposes a machine learning algorithm, reducing the majority samples and augmenting the minority samples in the difficult set, tackling the class imbalance problem in intrusion detection so that the classifier learns the differences better in training. (3) The classification model uses Random Forest (RF), Support Vector Machine (SVM), XGBoost, NLP with other methods, we divide the experiment into 30 methods.

- We propose an end-to-end deep learning model with ml models that is composed of logistic regression and attention mechanism. CNN can well solve the problem of Software Defined Networks and provide a new research method for Early Warning Proactive System
- We compare the performance of ML Modes 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.
- 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 methods.

2.1 WORKING OF CNN ALGORITHM

CNN (Convolutional Neural Network) is a deep learning algorithm commonly used in image recognition, but it can also be used in intrusion detection systems. The process of using CNN in an early warning intrusion detection system typically involves the following steps:

Data Collection: The system collects data from various sources such as network traffic, system logs, and user behavior.

Data Preprocessing: The collected data is preprocessed by removing noise, normalizing the data, and converting it into a format that can be used by the CNN algorithm.

Feature Extraction: The preprocessed data is fed into the CNN algorithm, which automatically extracts important features from the data.

Training: The extracted features and corresponding labels (indicating whether an intrusion occurred or not) are used to train the CNN algorithm. The CNN algorithm learns to identify patterns in the data that are indicative of intrusions.

Testing: The trained CNN model is tested on new data to evaluate its accuracy in detecting intrusions.

Early Warning: The CNN algorithm is integrated into an early warning system that analyzes incoming data in real-time. When the system detects a potential intrusion, it alerts the appropriate security personnel so that action can be taken to prevent the intrusion from succeeding.

The CNN algorithm can be trained on a large dataset of labeled data to improve its accuracy in detecting intrusions. It can also be fine-tuned over time as new data becomes available. The early warning system can be further improved by integrating it with other security measures such as firewalls, antivirus software, and access control systems.

The below diagram explains the working of the CNN Algorithm:

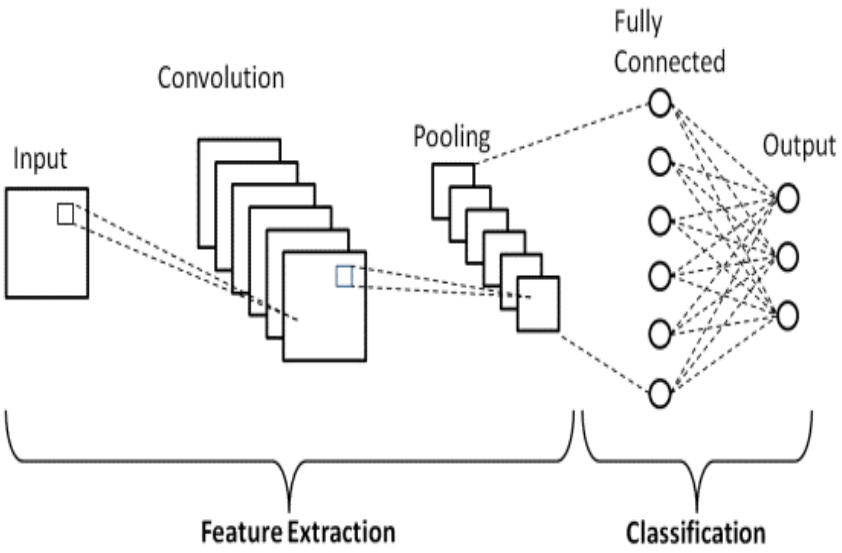

Fig 2.1.1 CNN Algorithm

3. System Architecture

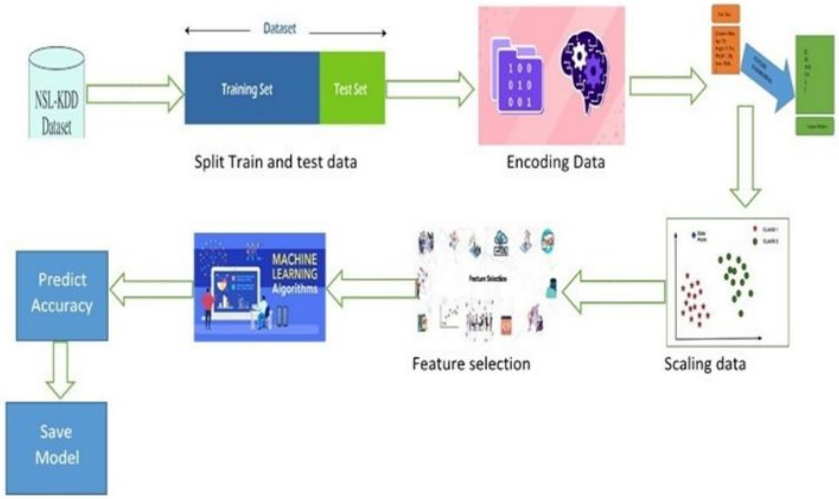

Fig 3.1 System Architecture

The NSL-KDD dataset is a popular data set for deep learning research. It is widely used for research in intrusion detection, network security, and network traffic analysis. The dataset contains around 4.9 million network connections, each of which is labeled as either normal or an attack. The dataset can be used to develop and evaluate deep learning models for intrusion detection, network security, and network traffic analysis. Additionally, the dataset can also be used to identify and classify anomalous network traffic.

After obtaining the dataset, it is important to split it into two parts: training dataset and test dataset. The training dataset is used to train the model, while the test dataset is used to evaluate the performance of the model. When creating the training dataset, it is important to select a subset of the dataset that is

representative of the overall population. This can be done by randomly selecting a certain percentage of the dataset for training. The remaining records can be used for testing.

One way of encoding data in intrusion detection is to use one-hot encoding. This involves taking categorical features and creating a new binary feature for each possible value. For example, if the original data has a categorical feature for country, then one-hot encoding would create a new binary feature for each country in the dataset. This encoding scheme makes it easier for algorithms to process and analyze the data. Other encoding techniques, such as ordinal encoding, label encoding, and hashing, can also be used.

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. It involves selection and construction of features from your dataset, which can be used to train a machine learning model. Scaling data in intrusion detection is important to ensure that malicious behavior is detected regardless of the size of the attack.

4. Conclusion

We have proposed a framework ,deep learning-based CNN algorithms have shown great potential in enhancing proactive system security. By using CNNs, security systems can effectively detect and respond to threats in real-time, ultimately improving the overall security posture of the system. Through the use of various deep learning techniques, such as transfer learning and reinforcement learning, CNNs can learn from vast amounts of data to identify patterns and anomalies that may indicate a security threat. While there are still challenges and limitations to be addressed, the continuous development and refinement of deep learning-based CNN algorithms hold promise for achieving a more secure and resilient system.

5. Future Enhancement

Although the neural networks strengthen data expression, the current public datasets have already extracted the data features in advance, which is more limited for deep learning to learn the preprocessed features and cannot take advantage of its automatic feature extraction. Therefore, in the next step, we plan to directly use the deep learning model for feature extraction and model training on the original network traffic data, performance the advantages of deep learning in feature extraction, reduce the impact of imbalanced data and achieve more accurate classification.

6. References

1. D. E. Denning, ‘‘An intrusion-detection model,’’ IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. 1987.
2. N. B. Amor, S. Benferhat, and Z. Elouedi, ‘‘Naive Bayes vs decision trees in intrusion detection systems,’’ in Proc. ACM Symp. Appl. Comput. (SAC), 2004, pp. 420–424.
3. M. Panda and M. R. Patra, ‘‘Network intrusion detection using Naive Bayes,’’ Int. J. Comput. Sci. Netw. Secur., vol. 7, no. 12, pp. 258–263, 2007.
4. M. A. M. Hasan, M. Nasser, B. Pal, and S. Ahmad, ‘‘Support vector machine and random forest modeling for intrusion detection system (IDS),’’ J. Intell. Learn. Syst. Appl., vol. 6, no. 1, pp. 45–52, 2014.
5. N. Japkowicz, ‘‘The class imbalance problem: Significance and strategies,’’ in Proc. Int. Conf. Artif. Intell., vol. 56, 2000, pp. 111–117.

6. Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ Nature, vol. 521, no. 7553, pp. 436–444, 2015.
7. Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, ‘‘Deep learning for visual understanding: A review,’’ Neurocomputing, vol. 187, pp. 27–48, Apr. 2016.
8. T. Young, D. Hazarika, S. Poria, and E. Cambria, ‘‘Recent trends in deep learning based natural language processing [review article],’’ IEEE Comput. Intell. Mag., vol. 13, no. 3, pp. 55–75, Aug. 2018.
9. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, ‘‘A deep learning approach to network intrusion detection,’’ IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, Feb. 2018.
10. D. A. Cieslak, N. V. Chawla, and A. Striegel, ‘‘Combating imbalance in network intrusion datasets,’’ in Proc. IEEE Int. Conf. Granular Comput., May 2006, pp. 732–737.