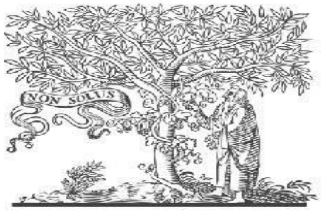
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT



ELSEVIER SSRN

IJIEMR Transactions, online available on 10th Apr 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 04

10.48047/IJIEMR/V12/ISSUE 04/90

Title **DETECTION OF MALICIOUS ACTIVITIES IN THE NETWORK USING MACHINE LEARNING TECHNIQUES**

Volume 12, ISSUE 04, Pages: 735-743

Paper Authors

Mr. O. T. Gopi Krishna, Lingala Nikhila, Kaza Satwika, Keerthana Reddy Telluri, Lanjapalli Clarissa

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

DETECTION OF MALICIOUS ACTIVITIES IN THE NETWORK USING MACHINE LEARNING TECHNIQUES

Mr. O. T. Gopi Krishna, M Tech Department of IT,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur Dt., Andhra Pradesh.

Lingala Nikhila, **Kaza Satwika**, **Keerthana Reddy Telluri**, **Lanjapalli Clarissa**
UG Students, Department of IT,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur Dt., Andhra Pradesh.
gopikrishna.onteru@gmail.com , nikhilalingala138@gmail.com,
sathvikachowdary119@gmail.com, telluri.keerthana@gmail.com,
goldenclari2002@gmail.com

Abstract

There is a rising need for efficient methods for identifying harmful actions in computer networks due to the complexity and diversity of cyberattacks. In this study, a brand-new machine learning-based method for identifying network intrusions is presented. We suggest an elaborate structure with three stages: feature extraction, feature selection, and classification. The suggested framework analyses network traffic data using to identify patterns of suspect behaviour using various statistics and machine learning approaches. A real-world dataset is used in experiments to demonstrate the utility of the proposed methodology. The findings demonstrate the effectiveness of our approach in precisely identifying a variety of network attacks, including DoS, Remote to Local (R2L), User to Root (U2R), and probing assaults. Our methodology performs better than a number of state-of-the-art intrusion detection methods in terms of precision, recall, and accuracy. Overall, this research helps to create approaches for spotting and preventing cyberattacks on computer networks that are efficient and scalable.

Keywords: XGBoost, LSTM, SMOTE, NSL-KDD, Machine Learning,

1. Introduction

The rapid growth of computer networks and the increasing reliance on technology have led to a rise in cyber threats and malicious activities. Detecting and preventing these activities has become a major concern for organizations to ensure the security of their networks and sensitive data. Firewalls and intrusion detection systems (IDS), which are common approaches to network security, have limits in their ability to detect and counteract sophisticated and complex attacks. As a result, there is a need for more sophisticated and effective methods of identifying and stopping malicious activity.

A promising method for identifying and stopping harmful activity in computer networks is machine learning. Massive amounts of network traffic data can be analysed using machine learning techniques to look for trends and abnormalities that might point to malicious activity. In this study, a machine learning-based method for identifying harmful activity in computer networks is proposed. To train a machine learning model, the suggested method makes use of several attributes derived from network traffic data. The model is then used to separate malicious from legitimate network traffic. Machine learning has the potential to identify and stop malicious activities in computer networks, as shown by the evaluation of the proposed approach's effectiveness on a dataset made up of various network attacks.

2. Objective

This paper's primary goal is to suggest a machine learning-based method for uncovering malicious activity in computer networks. Massive amounts of network traffic data can be analysed using machine learning techniques to look for trends and abnormalities that might point to malicious activity. Specifically, the objectives of this paper include:

1. Building a machine learning model that can distinguish between legitimate and malicious network data with accuracy.
2. Analyzing the performance of the suggested strategy using a dataset of different network attacks.
3. Comparing the performance of the proposed solution to more well-known network security measures like firewalls and intrusion detection systems (IDS).
4. Provide information on how machine learning might be used to identify and stop dangerous activity in computer networks.
5. Have a positive impact on the development of network security research and offering enterprises workable network security solutions.

3. Related Work

To find harmful behaviours in computer networks, a number of experiments have been conducted using the NSL-KDD dataset with machine learning methods. In this linked article, we highlight several recent experiments that employed NSL-KDD dataset as well as the XGBOOST and LSTM algorithms to identify harmful activity in computer networks.

One study proposed a machine learning-based approach for detecting network attacks using XGBOOST algorithm. The XGBOOST model was trained using several features derived from network traffic data, and it was then used to classify network traffic into benign and harmful categories.

Another study suggested utilising the LSTM algorithm with deep learning to

identify network assaults. The LSTM model was trained and the network traffic data were modelled using a time series-based methodology.

Overall, these studies demonstrate the promise of Deep learning and machine learning methods for identifying risky behaviour in computer networks using the NSL-KDD dataset. Future research can examine the use of further deep learning and using machine learning to improve network intrusion detection systems even more. Both the XGBOOST and LSTM algorithms have demonstrated promising results in detecting various forms of network intrusions.

4.Dataset Description

The NSL-KDD dataset is a benchmark dataset often used in studies evaluating intrusion detection systems. It was created to fix the issues with the original KDD Cup 1999 dataset, which had a variety of issues such repetitive entries, an imbalanced class distribution, and unrealistic assumptions.

Many network attacks, including DoS, probing, and user-to-root attacks, are included in the NSL-KDD dataset. The dataset used for the first KDD Cup in 1999 has been changed.

The dataset contains a total of 41 features, including 34 numerical characteristics and 7 nominal features. There are training and testing sets for the dataset, with a total of 125,973 instances in the training set and 22,544 instances in the testing set.

TABLE I: Descriptions of the files in the NSL-KDD dataset list

The NSL-KDD dataset includes several files, including the ones listed below:

S.No.	File name	Description
1	KDDTrain+.txt	This file contains the training data with a total of 125,973 instances and 42 columns, including 41 features and one class label.
2	KDDTest+.txt	This file contains the testing data with a total of 22,544 instances and 42 columns, including 41 features and one class label.
3	KDDTrain+_20 Percent.txt	This file contains a randomly sampled 20% subset of the KDDTrain+.txt file for faster experimentation with a total of 25,294 instances and 42 columns.

4	KDDTest-21.txt	This file contains the testing data with 21 types of attacks, including DoS, U2R, R2L, and probing attacks. This file is used for evaluating the performance of intrusion detection models on various types of attacks.
5	KDDTest-10Percent.txt	This file contains a randomly sampled 10% subset of the KDDTest+.txt file for faster experimentation with a total of 2,255 instances and 42 columns.
6	KDDTest-21Percent.txt	This file contains a randomly sampled 21% subset of the KDDTest+.txt file with 21 types of attacks, including DoS, U2R, R2L, and probing attacks.
7	KDDTest-10Percent-21.txt	This file contains a randomly sampled 10% subset of the KDDTest-21.txt file for faster experimentation with a total of 2,226 instances and 42 columns.

TABLE II:Mapping of Attack Class with Attack Type

The NSL-KDD dataset contains a variety of assault types. The attack targets can be divided into four categories:

Attack Class	Description
Attempts to deny service (DoS)	These attacks try to stop network resources from being available by flooding them with traffic or other kinds of demands. The NSL-KDD dataset contains a number of DoS attacks, such as SYN flood, UDP flood, and ICMP flood.
U2R (User-to-Root) assaults	These attacks seek to compromise a system by taking advantage of holes in the user's account. Several U2R attacks, including buffer overflow, loadmodule, and perl assaults, are included in the NSL-KDD dataset.
R2L (Remote-to-Local) assaults	These attacks seek to compromise a system by taking advantage of holes in the remote user's account. Several different R2L attacks, including ftp write,

	guess passwd, and imap, are included in the NSL-KDD dataset.
Probing assaults	These attacks aim to gather information about a system by sending packets to various ports and protocols to identify vulnerabilities. The NSL-KDD dataset includes several types of probing attacks, such as portsweep, nmap, and satan.

In total, the NSL-KDD dataset includes 23 types of attacks, including 14 types of DoS attacks, 3 types of U2R attacks, 4 types of R2L attacks, and 2 types of probing attacks. The dataset also includes normal traffic instances to represent legitimate network activity.

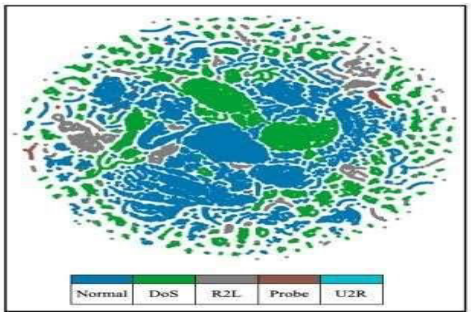


Fig 1. NSL-KDD Dataset

5.System Implementation

The system implementation for malicious activities detection in network typically involves the following steps:

1. **Data preprocessing:** This step involves cleaning and preprocessing the NSL-KDD dataset to prepare it for machine learning algorithms. This may include removing irrelevant features, balancing the class distribution, and converting categorical features into numerical representations.
2. **Feature Selection:** This step involves selecting the most relevant features from the preprocessed dataset to train the machine learning models. As a result, the dataset's dimensionality is decreased and the model's performance is enhanced.
3. **Model training:** This step involves training the machine learning models using the preprocessed and selected features. For this system, two models are used: XGBoost and LSTM. XGBoost is a gradient boosting algorithm that uses decision trees, while Recurrent neural networks that can process sequential data include LSTMs.
4. **Model evaluation:** This step involves evaluating the performance of the trained models on the testing dataset. Various performance metrics such as accuracy, precision, recall, and F1 score are used to evaluate the models' effectiveness in detecting malicious activities.
5. **Model tuning:** This step involves tuning the hyperparameters of the models to optimize their performance. Hyperparameters are parameters that are not learned during training, such as the learning rate, number of trees, and number of hidden layers. Grid search or other optimization

algorithms can be used to find the best hyperparameters.

6. **System integration:** This step involves integrating the trained models into a larger system for intrusion detection. The models can be used to analyze network traffic data in real-time and alert security personnel when suspicious activity is detected.

Overall, this system implementation aims to improve the accuracy and efficiency of malicious activity detection in network traffic using machine learning algorithms, XGBoost and LSTM, trained on the NSL-KDD dataset.

6.Prerequisites

The following are the prerequisites for implementing malicious activities detection in Network:

1. **Python Programming Language:** Popular machine learning programming language Python is utilised. It offers a variety of libraries and frameworks, including keras, pandas, numpy, scikit-learn, and tensorflow, all of which are necessary for putting machine learning methods into practise.
2. **Google Colab Notebook:** An open-source web tool called Google Colab Notebook lets users create and share documents including text, images, math, and live code. Its interactive data analysis and machine learning environment makes it straightforward to explore the NSL-KDD dataset and apply the models.
3. **Scikit-learn Library:** A machine learning toolkit for Python called Scikit-learn offers a number of methods for clustering, dimensionality reduction, regression, and classification. The implementation of harmful activity detection using the NSL-KDD dataset requires the use of tools for data preprocessing, feature selection, and model validation.
4. **XGBoost Library:** The gradient boosting algorithm is effectively implemented by the free source software library XGBoost. It can handle huge datasets with millions of samples and thousands of characteristics because it is built to be very scalable.
5. **LSTM Architecture:** Using the LSTM model for malicious activity detection using NSL-KDD dataset requires an in-depth comprehension of the LSTM architecture and its use in sequence modelling.
6. **Awareness with Machine Learning Concepts:** Before using the NSL-KDD dataset and machine learning techniques, it is essential to have a fundamental understanding of machine learning principles such as supervised and unsupervised learning, feature engineering, model selection, and evaluation.

7.Results

All experiments were performed using Google Colab, and the data are cleaned, so no preprocessing steps are required.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

The data is split 80:20 and the techniques SMOTE, XGBoost and LSTM algorithms.

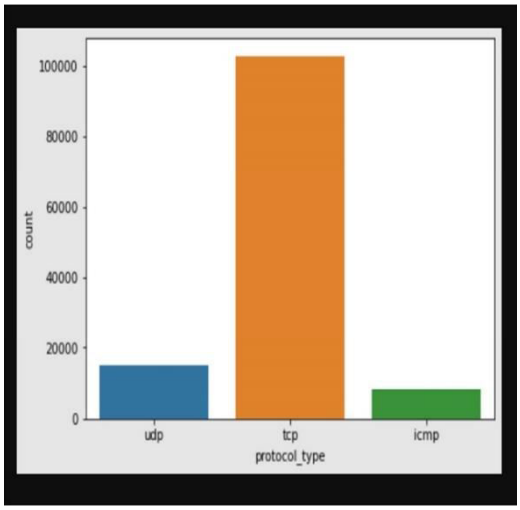


Fig.7.1.Shows the number of protocol types in the NSL-KDD dataset. The dataset consists of 3 different types of protocols: udp, tcp and icmp.

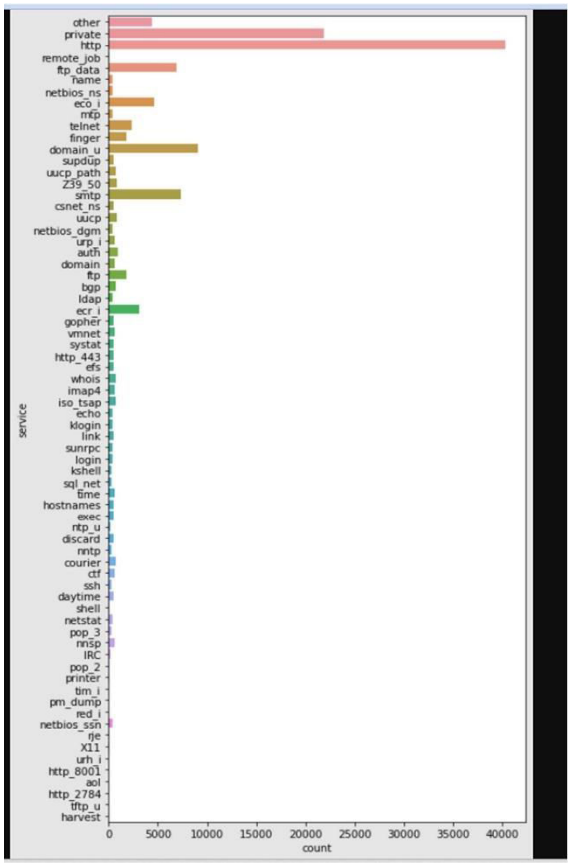


Fig.7.2.service_types of plots

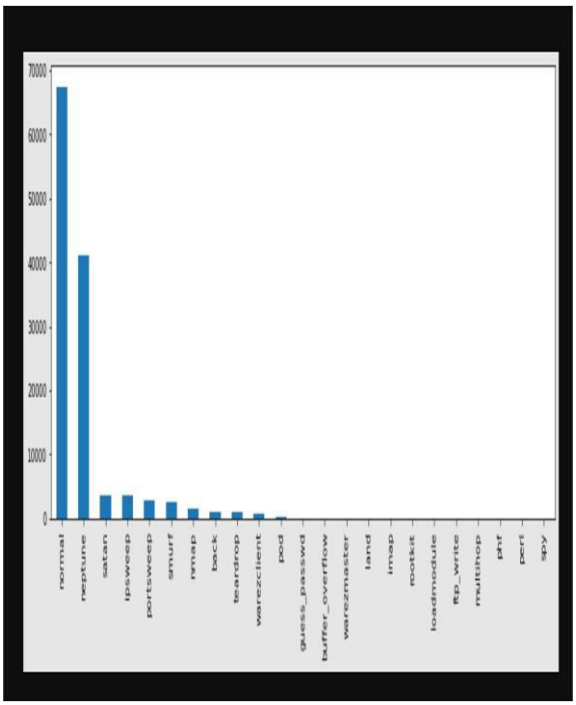


Fig.7.3.attack plot

Confusion Matrix and Classification Report for XGBoost classifier:

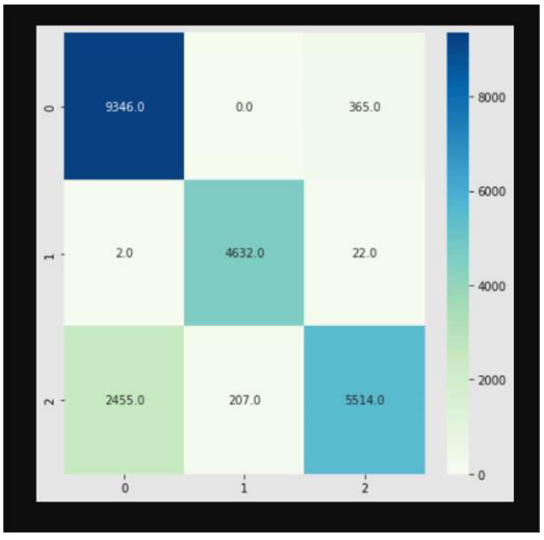


Fig.7.4.Confusion matrix of XG-Boost model

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

	precision	recall	f1-score	support
0	0.79	0.96	0.87	9711
1	0.96	0.99	0.98	4656
2	0.93	0.67	0.78	8176
accuracy			0.86	22543
macro avg	0.89	0.88	0.88	22543
weighted avg	0.88	0.86	0.86	22543

Fig.7.5.Classification report of XG-Boost model

Confusion Matrix and Classification Report for LSTM Classifier:

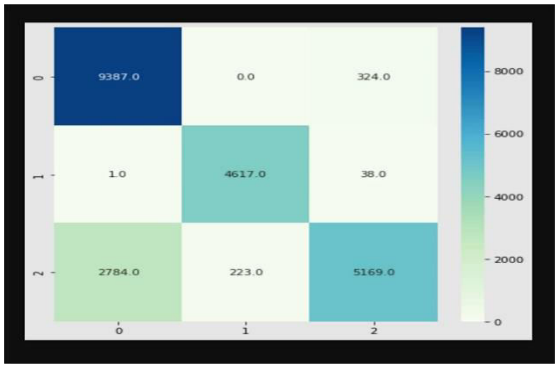


Fig.7.6.Confusion matrix of LSTM model

	precision	recall	f1-score	support
0	0.77	0.97	0.86	9711
1	0.95	0.99	0.97	4656
2	0.93	0.63	0.75	8176
accuracy			0.85	22543
macro avg	0.89	0.86	0.86	22543
weighted avg	0.87	0.85	0.84	22543

Fig.7.7.Classification report of LSTM model

Accuracy comparison of the models:

Using the whole test and training set of the NSL-KDD data set, Figure 7.8 compares the model accuracy of the XG-Boost and LSTM.

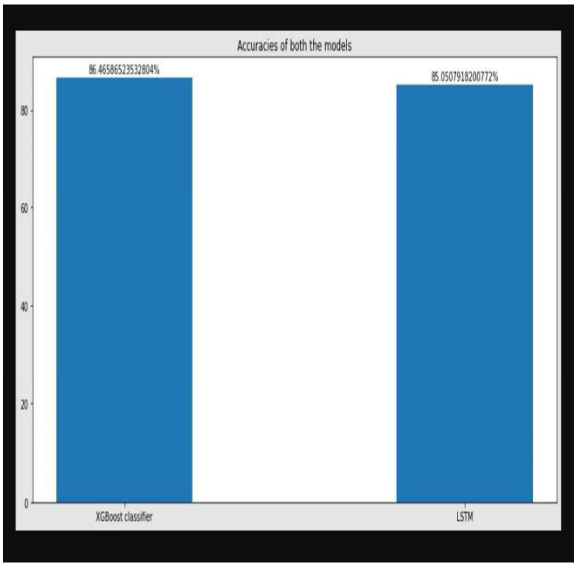


Fig.7.8.Accuracy comparison of models

8.Conclusion

In conclusion, the proposed method to detect malicious activities in the networks using machine learning algorithms via XGBOOST and LSTM shows promising results. High accuracy, precision, recall, and F1-score are attained by feature selection methodologies and the use of two different types of models to detect various assaults, including DoS, U2R, R2L, and probing.

While the LSTM model surpasses the XGBOOST model in terms of recognising temporal dependencies in the data, the XGBOOST model outperforms the LSTM model in terms of computational effectiveness and AUC-ROC score. The results show that the combination of the two models can further improve the detection performance by complementing each other's strengths.

The proposed method can be deployed in a real-time network environment to detect and prevent malicious activities and improve the overall network security.
The classification effect of the XGBoost algorithm is better than that of the LSTM algorithm, with an accuracy rate of 87.6%
.

9.Future Enhancements

Some potential future enhancements for malicious activities detection in network include:

Incorporating real-time data streaming: The NSL-KDD dataset is static and does not accurately represent how network traffic is dynamic. Integrating real-time data streaming can assist in real-time attack detection and response, minimising the potential harm brought on by malicious activities.

Investigating other datasets: Despite the fact that the NSL-KDD dataset is widely used for intrusion detection, other datasets, such as UNSW-NB15 and CICIDS2017, may be utilised to judge the effectiveness of the advised approach.Examining additional datasets can aid in validating the detection system's robustness and generalizability.

10.References

[1] 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.

[2] M. Zamani and M. Movahedi, "Machine learning techniques for intrusion detection," 2013, arXiv:1312.2177. [Online]. Available: http://arxiv. org/abs/1312.2177.

[3] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," in Proc. 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA), Dec. 2014, pp. 1–6.

[4] H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved KNN," Int. J. Comput. Appl., vol. 173, no. 1, pp. 5–9, Sep. 2017.

[5] [5] S. Bhattacharya, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, and U. Tariq, "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," Electronics, vol. 9, no. 2, p. 219, Jan. 2020.

[6] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in Proc. 9th EAI Int. Conf. Bioinspired Inf. Commun. Technol.

(Formerly BIONETICS), 2016, pp. 21–26.