WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

NETWORK INTRUSION DETECTION SYSTEM USING ML

***Akshay Dawar*[*1],RVS PRANAV*[*2], Sohen Anil Mondal*[*3]**

[*1] Department of Computer Science Engineering, SRM Institute of Science and Technology, Modinagar, Uttar Pradesh, India
[*2] Department of Computer Science Engineering, SRM Institute of Science and Technology, Modinagar, Uttar Pradesh, India
[*3] Department of Computer Science Engineering, SRM Institute of Science and Technology, Modinagar, Uttar Pradesh, India

ABSTRACT:

Among the most important issues facing modern society is network security. The weaknesses in network security have grown in importance over the last ten years due to the internet's rapid expansion and widespread use. To improve accuracy and efficiency in identifying possible security breaches, this study suggests a Network Intrusion Detection System (NIDS) that makes use of Machine Learning (ML) capabilities. The suggested NIDS seeks to evaluate network traffic patterns and spot unusual behaviors suggestive of cyber threats by utilizing a variety of machine learning approaches, including ensemble methods, supervised learning, and unsupervised learning. Additionally, using carefully labeled datasets, we will train the algorithm to identify patterns linked to both benign and malevolent network activity. This study shows how useful the Knowledge Discovery and Data Mining (KDD) dataset is for testing and evaluating different machine learning techniques. It focuses mostly on the KDD preparation step to provide a credible and fair experimental data set

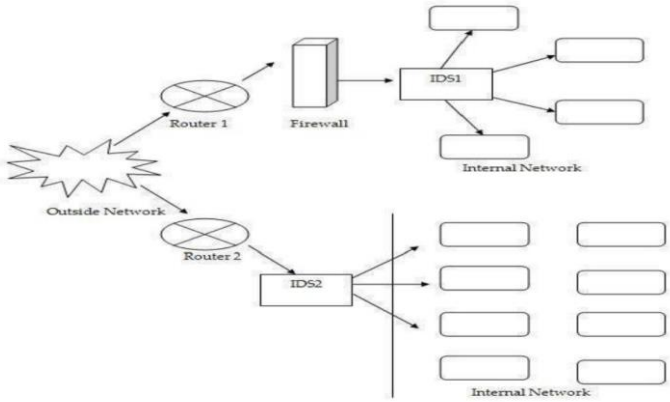
Keywords: Machine Learning, cyber threats, supervised learning, ensemble methods.

INTRODUCTION

In this modern era protection of individual property has become very important. The new ways through which cyber-attacks happen have become very concerning to the authorities. In times like this Machine Learning and NIDS (Network intrusion detection system has come as handy for solving our various problems. The old or traditional methods of tracking intrusion have long been left behind by the new ways how cyber have become prevalent in these times. Therefore, ML-based intrusion detection systems provide a standard way of using algorithms to detect patterns autonomously and detect oddity from traffic networks, providing more accurate and dynamic threat detection. As the rule-based systems are vulnerable to huge amounts of false positive rates, limited scalability, thereby losing their efficacy in detecting intrusions. We would define the underlying rules and principles that govern the many principles of ML driven models in cybersecurity. By deep diving into the details of ML algorithms, feature selection strategies and preprocessing data techniques we provide cybersecurity researchers with tools needed to use ML-based intrusion detection systems. Moreover, we can employ a variety of models rather than just a few to determine the best course of action when dealing with data silos issues. We go over every framework available for successfully refining various machine learning algorithms. We also go through methods for securing our data.

Signature-based detection, Anomaly-based detection, Protocol analysis, Attack categorization, correlations,
False positive,False negative, Packet header analysis, Machine learning algorithms, variance, bias.

1.1. ***Network Structure***



PROCEDURES AND METHODOLOGY

Cleaning and preparing the dataset

This data is KDDCUP'99 dataset, which is widely used as one of the few publicly available datasets for network-based anomaly detection systems.

Basic Exploratory Analysis

We distribute the dataset and explore it based on different parameters: -

1. Protocol Distribution
2. Service Distribution
3. Flag Distribution
4. Attack Distribution
5. Attack Class Distribution

Attack Class is a new parameter created by us, and not previously given in the dataset. It will help in grouping different types of attacks into a class. Attack class features will be the target. It consists of 5 categories which will be predicted using multinomial classification. 0 means normal 1 means DOS 2 means Probe 3 means R2L 4 means U2R.

Variable Reduction

Variable reduction, also known as feature selection or dimensionality reduction, is a critical preprocessing step in machine learning and data analysis aimed at enhancing model performance, interpretability, and computational efficiency.

Variable Reduction is possible using different techniques based on
-low variance
-high missing values
-high correlations

Variable reduction involves selecting a subset of relevant features from the original set of variables. This process is crucial for mitigating the curse of dimensionality, where an excessive number of features can lead to overfitting, increased computational complexity, and reduced model generalization.

Cases where the number of features is prohibitively high, dimensionality reduction techniques are employed to transform the dataset into a lower-dimensional space while preserving most of the essential information.

We will be using Select K-Best Technique for our model. The Select K-Best technique is a feature selection method commonly used in machine learning to select the top k most relevant features from a dataset. Select K-Best works by assigning a score to each feature in the dataset based on a predefined scoring function. The scoring function evaluates the statistical relationship between each feature and the target variable.

Common scoring functions include chi-squared for categorical targets and ANOVA F-value for numerical targets. After computing scores for each feature, Select K-Best selects the top k features with the highest scores. The value of k is determined by the user and depends on factors such as the

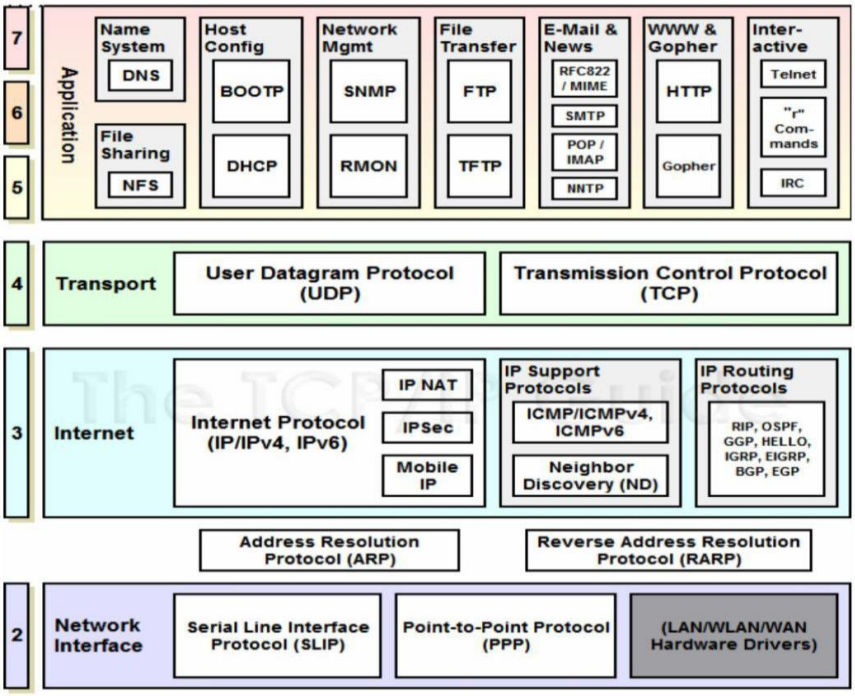
desired model complexity, computational resources, and the nature of the dataset.

Training and testing the model

We will train the model on 5 different machine learning algorithms: -

1. Logistic Regression
2. Gaussian Naïve Bayes
3. Linear SVC
4. Neural Network (Multilayer Perceptron)
5. Support Vector Classifier

TCP/IP HEADER STACK



ANALYSIS AND IMPLEMENTATION

In this section, we will be describing the modeling techniques used in our research.We will be using all supervised algorithms in this research as we are working with clearly labeled data.

Supervised machine learning uses labeled data to generate a function that maps an input to an output. The function is constructed from labeled training data.One of the main advantages of supervised learning is to use previous experiences to produce outputs. In addition, previous results can be used to improve the algorithm by optimizing the performance criteria to reach a precise model.

Supervised learning is used to solve many computational problems. However, the model needs precise and good input during the training phase to produce good outputs. In addition, this training requires a lot of computation time.

We used 5 different machine learning algorithms: -

1. **Logistic Regression**

It is a statistical method used for solving binary classification problems. It is a technique that is used to predict the probability of the event by putting the data against a logistic curve. It includes:

Binary Classification: It is ideal for binary classification problems where the variable has only two outcomes like 1/0, yes/no etc.

Sigmoid Function: Instead of giving a straight answer like linear regression, logistic regression uses a special S-shaped curve called the sigmoid function. It is different from the linear regression wherein it uses a s-shaped curve i.e. sigmoid function to take input and divide it in range of 0 and 1. It switches like a flip-flop between two outcomes.

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Linear Decision Boundary: It is about finding a plane in higher dimensions that best separates the two classes in your data. This line acts as a boundary for decision making.

Training: It is a mode that learns from the data to predict the minimum variation between the predicted probability and actual outcome. It uses the process of maximum likelihood estimation.

Decision Making: If you are using a threshold which is usually 0.5 then it decides which class it would assign the data point based on whether it predicted data above or below it.

It works best when the relationship between the input features and the outcome is roughly linear, and there are no significant interactions between features.

2. Gaussian Naïve Bayes

Gaussian Naïve Bayes (GNB) is a technique which uses Bayes theorem where we calculate the probability from the evidence overserved.
It is different from the Bayes algorithm which follows feature independence, GNB uses features that follow a Gaussian (normal) distribution. The above line means all the data points that forms a class cluster around the mean according to the bell curve shape:

$P(H|U)=(P(U|H)*P(H))/P(U)$

GNB assumes that the features are independent of all class labels and that we cannot find the value of one feature through the feature of another.

3. Support Vector Classifier

The Support Vector Classifier (SVC) is a powerful supervised learning algorithm widely used for classification tasks in machine learning. At its core, SVC aims to find the optimal hyperplane that best separates different classes within a dataset by maximizing the margin between the hyperplane and the nearest data points, known as support vectors. This margin maximization contributes to the algorithm's ability to generalize well on unseen data.It is primarily used to separate data in different groups by finding the best boundary line between them. SVM achieves this by maximizing the margin, or the distance between the boundary line and the nearest data points from each group. SVC can handle both linear and non-linear classification tasks through the use of kernel functions, which map the input data into a higher-dimensional space where a linear separation can be achieved.

The model is trained by solving a constrained optimization problem, typically using quadratic programming techniques, to identify the hyperplane that minimizes classification errors while maximizing the margin. SVC offers several advantages, including robustness against overfitting when the regularization parameter C is appropriately tuned, effectiveness in high-dimensional spaces, and capability to handle non-linear decision boundaries. These characteristics make SVC a popular choice for various applications, ranging from text categorization and image classification to bioinformatics and beyond.

4. Neural Network

A neural network mimics a human brain and forma crucial concept of AI and machine learning which can be used to find different patterns in the data. Neurons are basic units that process input data to produce an output with internal parameters.

A multilayer perceptron is a type of feedforward neural network consisting of fully connected neurons with a nonlinear kind of activation function. It is widely used to distinguish data that is not linearly separable.MLPs have been widely used in various fields, including image recognition, natural language processing, and speech recognition, among others. These layers can be: -
Input layer: - The input layer consists of nodes or neurons that receive the initial input data. Each neuron represents a feature or dimension of the input data. The number of neurons in the input layer is determined by the dimensionality of the input data.
Hidden layer: - Between the input and output layers, there can be one or more layers of neurons. Each neuron in a hidden layer receives inputs from all neurons in the previous layer (either the input layer or another hidden layer) and produces an output that is passed to the next layer.
Outer layer: - This layer consists of neurons that produce the final output of the network. The number of neurons in the output layer depends on the

nature of the task. In binary classification, there may be either one or two neurons depending on the activation function and representing the probability of belonging to one class; while in multi-class classification tasks, there can be multiple neurons in the output layer.

5. Linear Support Vector Machine (SVM)

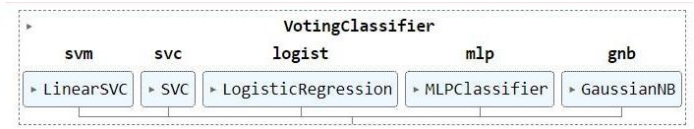
The Linear Support Vector Classifier (Linear SVC) is a variant of the Support Vector Machine (SVM) algorithm designed for linearly separable datasets. It operates by identifying the optimal hyperplane that best divides the classes in the input data space, aiming to maximize the margin between this hyperplane and the nearest data points. Unlike traditional SVC, Linear SVC specifically targets linearly separable data and does not rely on kernel functions for mapping to higher-dimensional spaces. Instead, it optimizes a linear decision boundary directly on the original feature space, making it computationally efficient and well-suited for large-scale datasets.

Linear SVC is trained by solving a convex optimization problem, typically using techniques like coordinate descent or stochastic gradient descent. This classifier is particularly effective when dealing with high-dimensional data or when the number of features exceeds the number of samples. Linear SVC offers simplicity, interpretability, and scalability, making it a preferred choice for applications requiring efficient binary classification with linear decision boundaries.

RESULTS AND DISCUSSIONS

This study underscores the substantial impact of machine learning on enhancing the efficacy of Intrusion Detection Systems (IDSs), emphasizing the pivotal role of dataset quality in determining IDS efficiency. Employing well-curated datasets is imperative, with many reviewed research papers leveraging labeled data to enhance model training. However, the burgeoning size of datasets poses a challenge, as conventional machine learning models may struggle to scale effectively. Consequently, researchers are increasingly turning to deep learning techniques, particularly Convolutional Neural Networks (CNNs), to pioneer innovative solutions. These approaches excel at extracting salient features from raw datasets, bolstering Network Intrusion Detection Systems (NIDS) against zero-day attacks. Moreover, NIDS must be regularly trained with real-time network data, although adopting these advanced methods comes at a cost, demanding more robust computing resources and prolonged processing times to train high-performing models.

Using the voting ensemble algorithm we can also find a final accuracy score: -




```
y_pred=ensemble.predict(X_test)
y_pred
```

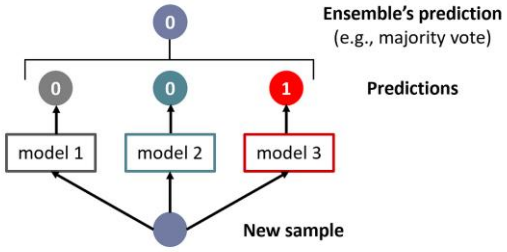
```
array([1., 0., 2., ..., 1., 0., 2.])
```

```
accuracy_score( Y_test, y_pred )
```

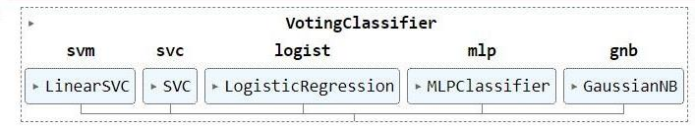
```
0.8379984917712816
```

Ensemble Model Accuracy score –83.79

Voting ensemble methodology:



Voting ensemble methods are often used to enhance classification accuracy and robustness. By combining the collective wisdom of various diverse classifying algorithms, they help in mitigating individual biases and variance.




```
y_pred=ensemble.predict(X_test)
y_pred
```

```
array([1., 0., 2., ..., 1., 0., 2.])
```

```
accuracy_score( Y_test, y_pred )
```

```
0.8379984917712816
```

CONCLUSIONS

Network Intrusion Detection Systems (NIDS) signifies a noteworthy progression in the field of cybersecurity. NIDS can now more effectively and precisely analyze large volumes of network data thanks to AI and ML, which improves threat detection capabilities.

Enhanced detection accuracy, less false positives, adaptability to changing threats, and higher operational efficiency through automation are the main advantages of utilizing AI and ML in NIDS. Using AI, NIDS can spot irregularities and intricate patterns that conventional rule-based systems could miss but that point to possible security breaches.

However, careful consideration of several criteria, such as data quality, model training, scalability, and continuing monitoring for model performance and effectiveness, is necessary for the successful implementation of AI-driven NIDS. In order to preserve confidence and support human decision-making in reaction to risks that have been discovered, organizations also need to address issues pertaining to the interpretability and transparency of AI-driven detections.

Future work in AI and ML research and development will improve NIDS capabilities even more, allowing for proactive threat detection and response to protect networks and digital assets from ever-evolving cyber threats. AI-powered NIDS will be essential for bolstering overall cybersecurity posture and reducing risks in today's digital ecosystem as cyberattacks become more complex.

REFERENCES

1. Buczak, Anna L., and Erhan Guven. "A survey of data mining and machine learning methods for cyber security intrusion detection." IEEE Communications Surveys & Tutorials 18.2 (2016): 1153-1176.
2. Axelsson, Stefan. "Intrusion detection systems: A survey and taxonomy." Technical report, Chalmers University of Technology, 2000.
3. Roesch, Martin. "Snort—lightweight intrusion detection for networks." USENIX Association, 1999.
4. Mukkamala, S., et al. "Intrusion detection using ensemble of soft computing paradigms." Journal of Network and Computer Applications 28.2 (2005): 167-182.
5. Lee, Wenke, and Salvatore J. Stolfo. "Data mining approaches for intrusion detection." USENIX Security Symposium 1998.
6. Amin, Syed Muhammad, et al. "Deep learning for network intrusion detection: A survey." IEEE Communications Surveys & Tutorials 23.1 (2021): 202-231.
7. Mohaisen, Aziz, and Omar Alrawi. "A survey on network anomaly detection using machine learning." Journal of Network and Computer Applications 145 (2020): 102447.
8. Garcia-Teodoro, Pedro, et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges." Computers & Security 28.1-2 (2009): 18-28.
9. Patcha, Animesh, and Jung-Min Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends." Computer Networks 51.12 (2007): 3448-3470.
10. Alazab, Mamoun, et al. "An efficient model for intrusion detection based on artificial immune system and fuzzy clustering." Computers & Security 30.5 (2011): 331-341.
11. Zanero, Stefano, and Matteo Bocchi. "Internet traffic classification using machine learning." ACM SIGCOMM Computer Communication Review 37.4 (2007): 11-22.
12. Lazarevic, Aleksandar, et al. "A comparative study of anomaly detection schemes in network intrusion detection." Proceedings of the Third SIAM International Conference on Data Mining. 2003.
13. Zhao, Jing, et al. "Survey of deep learning-based intrusion detection approaches." IEEEAccess 8 (2020): 17110-17125.

14. Das, Dipankar, et al. "A survey of deep learning in cyber security." Information 10.10 (2019): 307.
15. KD99 CUP DATASET (The UCI KDD Archive Information and Computer Science University of California, Irvine CA 92697-3425 October 28, 1999)
16. Elhag, Samir Mohamed, and Zeyar Aung. "Hybrid intrusion detection with ensemble of feature selection techniques and machine learning algorithms." IEEE Access 7 (2019): 153931-153945.