# Threat Detection and Neutralization

1st: Shikhar
*dept. name : Computer Science*
*(of Aktu)*
email: shikhar.2024cs1068@kiet.edu

2nd : Shubhi
*dept. name : Computer Dcience*
*(of Aktu)*
email : shubhi.2024cs1047@kiet.edu

2nd : Abhijeet Kannaujia
*dept. name : Computer Dcience*
*(of Aktu)*
email : abhijeet.2024cs1148@kiet.edu

***1: Abstract—***

The network size and related data have greatly increased as a result of the quick developments in the internet and communication areas. As a result, a lot of new assaults are created, which has made it difficult for network security to precisely identify breaches. Additionally, the fact that the invaders are there cannot be disregarded with the intention of launching numerous attacks within the network. Strong Intrusion Detection Systems (IDS) are now essential due to the quick spread of networked systems and the growing sophistication of cyberattacks. This work presents a sophisticated intrusion detection system that can dynamically classify the type of attack in real time in addition to detecting possible incursions. Utilizing machine learning techniques, the system improves its capacity to adjust to changing cyberthreats. The key innovation lies in the incorporation of machine learning classifiers, which enable the system to dynamically categorize detected intrusions into specific attack types.
The system can generalist and adjust to new and emerging threats since it has been trained on past defense data covering a variety of assault scenarios. By employing ensemble learning, the classification accuracy is significantly improved and a strong defense against false positives and false negatives is provided.
To validate the proposed system, extensive experiments are conducted using benchmark datasets and real-world network traffic. The results demonstrate the system's efficacy in accurately identifying and classifying a wide range of attacks, including Denial of Service (DoS), Probe, R2L etc.
By offering a dynamic and flexible defense mechanism, the intelligent intrusion detection system described in this study represents a substantial leap in the detection of cyber threats. Modern network infrastructures can be more secure thanks to the system's ability to adapt to the constantly changing cyber threat landscape thanks to the integration of machine learning.

**Keywords—** Some of the related keywords are Intrusion Detection System, Firewall, Key logger, Network Packets, Packets breaker, Random Forest Classifier.

***2: Introduction :***

The increasing danger of cyberattacks is a serious threat to information network security in today's world of networked technologies and digital communication. An essential line of defence against these attacks is provided by intrusion detection systems (IDS), which are designed to recognise and react to unauthorised access, questionable activity, and possible security breaches. Traditionally, intrusion detection systems (IDS) have not delved into the finer points of attack type classification, instead concentrating on identifying predetermined attack patterns or abnormalities.
This research aims to address the limitations of existing Intrusion Detection Systems by proposing an intelligent system capable of not only detecting intrusions but also dynamically classifying the specific type of attack. One cannot stress how important it is to be able to distinguish between different types of attacks in order to respond in a focused and efficient manner. Security experts that comprehend the type of intrusion are better able to modify their mitigation tactics, use resources wisely, and strengthen network defenses against certain threats. Extensive experiments with synthetic and real-world datasets are carried out to verify the efficacy of the suggested system. The outcomes of these tests show that the system can correctly detect and categorize a wide range of assaults, such as Denial of Service (DoS), Probe, R2L, and other types of malware distribution.
The use of machine learning classifiers to dynamically classify detected incursions into particular attack types is what makes this research novel. Through the examination of several feature sets derived from packet payloads, system logs, and network traffic, the system adjusts to the constantly changing threat environment. This flexibility is essential because cyber adversaries are always improving their strategies, methods, and processes.

3: **Literature Survey**

I. TITLE: Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT

AUTHORS: Muhammad Aslam,
Dengpan Ye,
Muhammad Hanif & Muhammad Asad

DESCRIPTION: The Internet of Things (IoT) is developing smart network infrastructure, but advanced Distributed Denial-of-Service (DDoS) security attacks pose a serious threat to its development. Enterprise networks' current network security solutions are prohibitively expensive and not IoT scalable.
Additional security measures are made possible by the inclusion of recently developed Software Defined Networking (SDN), which also significantly decreases computational overhead for Internet of Things network devices. As of right now, the sampling-based security approach to SDN-enabled IoT network infrastructure yields low accuracy and low detection of DDoS attacks. We provide in this research an SDN-enabled system for the detection and mitigation of distributed denial-of-service threats using adaptive machine learning (AMLSDM). In order to successfully detect and mitigate DDoS attacks, the suggested AMLSDM framework creates an SDN-enabled security mechanism for Internet of Things devices with the use of an adaptive machine learning classification model. By analysing the static aspects of the inspected network traffic, the proposed framework applies machine learning techniques in an adaptive multilayered feed-forwarding architecture to successfully detect DDoS attacks. The first layer of the proposed adaptive multilayered feed-forwarding framework builds a model for detecting DDoS attacks from training and testing environment-specific datasets using classifiers such as Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), k-Nearest Neighbour (kNN), and Logistic Regression (LR). An Ensemble Voting (EV) method receives the first layer's output and adds up the first layer classifiers' performances. Adaptive frameworks in the third layer monitor live, real-time network traffic in order to identify DDoS attacks in that traffic. In order to mitigate the detected DDoS attacks over Open Flow (OF) switches and modify the network resources for authorised network hosts, the suggested framework makes use of a remote SDN controller. The experimental results demonstrate that the suggested framework

outperforms current state-of-the-art methods in terms of lower false alarm rates and increased DDoS detection accuracy.
In order to detect DDoS attacks for network traffic of SDN-enabled IoT, we developed an AMLSDM framework in this paper that is based on an adaptive machine learning classification model. Additionally, the AMLSDM framework offers a DDoS mitigation mechanism for switching networks.
We run the DDoS inspection, DDoS mitigation, adaptive classification, and feature extraction modules in each phase. Our comprehensive simulation findings verify the AMLSDM framework's intelligent DDoS detection and mitigation for categorizing real-time network traffic produced by two SDN-enabled IoT networks. When compared to static AMLSDM-SVM, AMLSDM-NB, AMLSDM-kNN, AMLSDM-LR, and AMLSDM-RF classification settings, performance metrics such as accuracy, precision, recall, and F1 score validate that adaptive setup AMLSDM-EV performs better in classification. In order to verify the superior performance of the AMLSDM framework, we also compare its simulation results with those of the most recent versions of the LEDEM and CONA frameworks.
Future research will focus further on the mitigation of DDoS assaults against SDN controllers because this is a crucial area for advancing the use of SDN controllers in Internet of Things networks. Additionally, we'll expand the use of our suggested methodology to identify phishing attempts.

II. TITLE: Detection of Phishing Attacks with Machine Learning Techniques in Cognitive Security Architecture
AUTHORS: Ivan Ortiz-Garc ´ es´ ∗,
Roberto O. Andrade† & Mar´ıa Cazares‡

DESCRIPTION: In Latin America, there has been a surge in phishing attempts that surpass the capabilities of cybersecurity analysts in terms of operation. The application for cognitive security suggests using big data, machine learning, and data analytics to enhance threat detection response times. This paper examines the examination of unusual behavior associated with phishing online attacks and discusses how machine learning techniques may be used as a solution. In order to develop machine learning for phishing attack detection, this analysis makes use of contaminated data sets and Python tools. It analyses URLs to determine whether they are good or bad based on certain characteristics. The ultimate goal is to provide real-time information so that proactive decisions can be made to minimize the impact of an attack.
We can conclude that artificial intelligence, which is quicker, more effective, and enabled by modern technology to create better applications, is a useful instrument in the fight against anomalous behavior.
Some phishing tactics, such as shortening URLs, may run into problems with tools like this machine learning application, which can determine whether a URL is acceptable or harmful and then add it to a blacklist. Even though this machine learning may not always be accurate, we may verify those URLs using a web tool that shortens URLs. which leads us to our next suggestion. It is advised not to click on a truncated URL prior to We already know that a lot of those URLs may be malware, phishing scams, and other threats, so make sure to check it. With this knowledge of our shortcomings, the community can attempt to create new instruments that assist us in strengthening our inadequate methods.

III. TITLE: Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems
AUTHORS: Moses Garuba,
Chunmei Liu, and Duane Fraites

DESCRIPTION: Security systems for organisations need to be adaptive and agile in order to handle the growing threats posed by malicious software, virus attacks, and other code in addition to assaults from within. Effective network intrusion detection systems, which are a component of the layered defense strategy, must be able to achieve certain organizational goals. Heuristic-based network intrusion detection systems are able to fully satisfy the organization's aims, even while signature-based systems accomplish a number of organizational security goals. This research examines various organizational security goals through a comparative theoretical investigation to identify the network intrusion detection solution that most successfully satisfies these goals. Heuristic-based systems outperform signature-based systems in meeting organizational goals, according to a convincing analysis of the study.
Which technology gave clear security objectives and the flexibility, adaptability, and decreased vulnerability that an organization needs was the basis for the analysis. After going over the key comparison goals, a summary and conclusive analysis are covered to help identify which system works best for a certain organization.
The organizational objectives as stated in the document should be suitably satisfied by the NIDS for implementation, in line with the article's aim. In conclusion, both NIDS struggle with the primary goal of false positives and negatives. False positives are acceptable in terms of security, while false negatives are unacceptable since they let attacks pass unnoticed. Update goals, competency, and attack vulnerability are all closely related. These goals have to do with signature based NDS's reliance on knowledgeable staff for upgrades, which makes it vulnerable to intrusions. Heuristic-based NIDS, on the other hand, rely on behavioral patterns rather than updates and skilled staff. Its capacity to improve its detection analysis by ongoing sampling of typical behavior also lessens its vulnerability to threats.
The goal of coverage suggests that the system based on heuristics covers all facet of the network, both externally and internally. Signature-based solutions leave the organization open to internal risks because they only protect against assaults that use signatures, which are usually external attacks. Lastly, the intrinsic constraints show that signature-based NIDS are perpetually plagued by unknown assaults. Heuristic-based NIDS, on the other hand, are constrained by the requirement that attacks display anomalous behavior patterns. The technology cannot identify threats that don't behave abnormally.
In conclusion, it is clear that the thesis is supported since, in contrast to signature-based NIDS, heuristic-based NIDS successfully satisfy an organization's organizational objectives. The different compositions of the two NIDSs serve as the foundation for this kind of study. The fundamental cause of a signature-based NIDS's weakness is its reliance on knowledgeable staff. In short, if a detection system's effectiveness depends just on its workforce, it is not practical. Furthermore, regardless of the staff members' level of competency, vulnerabilities are introduced.
As a result, a system that is prone to attack by design is one that an organisation looking to implement an effective security defence should ignore. A heuristic-based network intrusion detection system's common characteristic, on the other hand, is its flexibility and adaptability in identifying known, unknown, or evasive threats within the network.
The system's capacity to acquire knowledge through continuous sampling of statistical and behavioural patterns enables it to identify potential dangers, regardless of their age.
As a result, an organisation should use this kind of system for its security defence since it offers a system that is independent of staff and upgrades, as well as the built-in capacity to identify the majority of network attacks, whether they are fresh and altered or originate from within or outside

**4: Methodology and Model**

The following is the Basic Pipeline of our model: - The attacks can be broadly categorized into four types: -

1. DOS-In this assault, the hacker aims to construct a device or network resource unavailable to its intended consumers by momentarily or permanently interfering with a host's ability to provide services over the Internet.
2. R2L - An attacker uses this kind of attack to get unapproved access to a victim computer over the network.
3. U2R - When gaining legal access to a local machine, this kind of assault is performed illegally in order to achieve the root's rights.
4. Probe: A probe is an assault that is intentionally designed to be detected by its target and report it using a distinguishable "fingerprint" in the report. The position and defensive capabilities of the detector are subsequently revealed to the attacker via the collaborative infrastructure.
A package is categorized as regular or safe if it does not fall under one of these four categories of attack.
Following are the methodologies:-

1. Feature Selection - Feature selection is a process of identifying relevant characteristics and eliminating unnecessary ones to accurately characterize a problem. This approach enhances machine learning algorithms' performance, helps understand data, reduces storage requirements, and potentially saves costs. It also offers simplicity, allowing for more basic models and acceleration. Overall, feature selection is a crucial aspect of machine learning.

2. Data Preprocessing -The data consists of 43 columns with various attributes like duration, protocol type, service, and flag. The data is identified using column names and summarizes each numerical feature in the num_summary file. Outlier capping is applied to restrict extreme values to the first and 99th percentiles to address outliers and enhance model resilience. This process helps maintain model resilience.

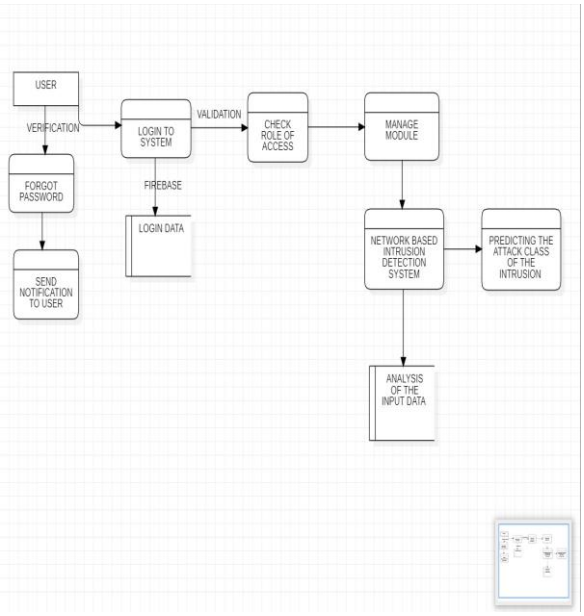
3. Model Selection - The Random Forest Classifier is an ensemble machine learning model that can handle both classification and regression tasks. It is built using a series of steps in the code. The train_test_split function from Scikit-Learn is used to divide the data into training and testing sets. The hyperparameters are used to generate the model, which is imported from Scikit-Learn. Using the fit() method, the model is trained on the training set of data, gaining the ability to recognize patterns in the data and generate predictions. Following training, the accuracy_score function is used to evaluate the model's performance on the testing set of data. This guarantees that the model can efficiently handle complicated datasets. It uses 70% for the training and remaining 30 percent for testing.

4. Intrusion Detection - The finalized ML model is then used to predict whether the packet is regular or attack type. Basically the attack classification is based on normal , dos , probe , U2r and R2L.

**Dataset**

A common dataset for auditing incursions simulated in a military network context is KDD CUP 1999. This dataset is widely employed in the tracking and analysis of intrusions, enabling the distinction between malicious and legitimate connections. This dataset was created in 1998 by the Défense Advanced Research Project Agency (DARPA), which stands for knowledge discovery and data mining. DARPA and MIT Lincoln Labs worked together to create a thorough and accurate IDS benchmarking environment that was centred on real-world scenarios. Network threat detection systems are benchmarked using datasets of this kind. The pre-processing step of the dataset's initial stage involves the selection of key features, the most important of which is feature selection. In this paper, we have included 11 features for experimental work that would give high accuracy with low false-positive rates.

5.


**5: Applied Algorithm**

1. Decision tree - Decision Tree is a supervised machine learning algorithm used for classification and regression of datasets. It uses a tree structure with nodes representing attributes,

branches representing decisions, and leaves representing outcomes. DT models include CART, C4.5, and ID3.58, with advanced learning algorithms like Random Forest and XGBoost also derived from multiple decision trees.

2. Logistic Regression - A logistic regression model, lr_clf, is created using scikit-learns Logistic Regression class and trained using training data. The model predicts class labels for test data, generating predictions stored in the y_pred variable. This model is useful for multi-class classification tasks.

3. K- Nearest Neighbour - KNN is a straightforward supervised machine learning technique that predicts the class of data samples based on feature similarity. Model performance is influenced by the parameter k; greater values result in misclassification and smaller values cause overfitting. By employing the Synthetic Minority Oversampling Technique to solve dataset imbalance, Karatas et al. were able to increase the detection rate of minority class attacks.

4. Adaboost - An ensemble learning technique called AdaBoost enhances the performance of underperforming learners to produce a powerful predictive model. It can be used to categorize network traffic into normal and anomalous patterns for the Network Intrusion Detection System (NIDS). The setup, training, and assessment of a group of inexperienced learners serves as an example of how AdaBoost is used. Robust performance evaluation is ensured via cross-validation, and the model's accuracy in classifying network traffic is reflected in the accuracy score on test data. AdaBoost is hence a useful tool for intrusion detection in a variety of network contexts.

5. Random Forest Classifier - Network intrusion detection systems (NIDS) use a machine learning method called the Random Forest Classifier to distinguish between potentially malicious and legitimate network data. It uses ensemble learning to improve accuracy, handles a variety of characteristics, and randomizes data to prevent overfitting. Cross-validation is a technique for evaluating performance between datasets. To simulate real-world situations, the model is tested on a different test set after being trained on training data. The efficacy of the model in classifying network traffic is evaluated using the accuracy score.

6. Neural Network - A deep learning model with multiple layers called DNN is used to simulate intricate nonlinear functions. It is made up of hidden layers, an output, and an input. Jia et al.'s network intrusion detection system (IDS) model demonstrated higher detection rates for most attack classes with four hidden layers. Wang et al. examined the role of individual attributes in generating adversarial scenarios using the NSL-KDD dataset. Vinayakumar et al. introduced the hybrid scalable DNN framework called Scale-hybrid-IDS-AlertNet for host and network intrusion detection. The model fared better when compared against different machine learning techniques utilizing publicly available datasets.

7. Support vector machine - SVM is a supervised machine learning algorithm that uses a max-margin separation hyper-plane in n-dimensional feature space to solve linear and nonlinear problems. It maps low-dimensional input vectors into high-dimensional feature spaces, enhancing efficiency and accuracy in predicting normal and malicious classes.

8. Bernoulli Naïve Bayes - The Bernoulli Naive Bayes classifier is a probabilistic classifier that uses binary features to classify network traffic into normal and anomalous patterns. It is used in the Network Intrusion Detection System (NIDS) to classify network traffic into normal and anomalous patterns. The model is trained on a training dataset and used to make predictions on test data. A confusion matrix is created to evaluate the model's performance, and a heatmap is displayed for better interpretation. The accuracy score is calculated by comparing predicted labels with the true labels of the test data.

**6: Result**

We used the Random Forest Classifier to train our dataset after extracting the features from WEKA. We also trained our model using KNN and Decision Trees, but the Random Forest Classifier produced the best results. Our model was trained using 65% of the KDD dataset. The trained model was evaluated on the remaining 35% of the dataset; it produced the best accuracy of 99% with a low false positive rate.
We developed a packet sniffer to collect real-time packets from the network for real-time analysis of the model. Next, our feature extractor script decodes the collected data to extract the necessary features and stores it in a CSV file. Finally, our trained model evaluates the real-time data and groups the packets into five categories: Normal, DOS, U2R, R2L, and Probe.
The code, as it is, will create a heatmap that visually represents the confusion matrix, making it easier to understand how well the Bernoulli Naive Bayes classifier's predictions align with the actual labels. It helps in evaluating the classifier's performance in terms of true positives, true negatives, false positives, and false negatives.
In summary, the code demonstrates the use of the SGD Classifier, an efficient and scalable classifier, for training a linear SVM model using hinge loss and L2 regularization. It then evaluates the model's performance by calculating accuracy and exploring the effect of changing the number of iterations on the model's performance.

The precision, recall, and f1 score calculations, which are provided below, are the most accurate ways to gauge accuracy:

| Category | Value |
|---|---|
| | |
| DOS | |
| Accurate Positive | 25479 |
| Accurate Negative | 9291 |
| Inaccurate Positive | 29 |
| Inaccurate Negative | 17 |
| | |
| NORMAL | |
| Accurate Positive | 8927 |
| Accurate Negative | 26503 |
| Inaccurate Positive | 11 |
| Inaccurate Negative | 5 |
| | |
| PROBE | |
| Accurate Positive | 229 |
| Accurate Negative | 34566 |
| Inaccurate Positive | 9 |
| Inaccurate Negative | 11 |
| | |
| R2L | |
| Accurate Positive | 119 |

| | |
|---|---|
| Accurate Negative | 34674 |
| Inaccurate Positive | 5 |
| Inaccurate Negative | 18 |
| | |
| U2R | |
| Accurate Positive | 6 |
| Accurate Negative | 34804 |
| Inaccurate Positive | 1 |
| Inaccurate Negative | 5 |
| | |

$$Precision = \frac{Accurate\ Positives}{Accurate\ Positives + Inaccurate\ Positives}$$

Precision :-

DOS-~0.99
PROBE-~0.05
R2L-~0.96
U2R-~0.86
NORMAL-~0.99

$$Recall = \frac{Accurate\ Positives}{Accurate\ Positives + Inaccurate\ Negatives}$$

Recall :-

DOS-~0.99
PROBE-~0.96
R2L-~0.88
U2R-~0.55
NORMAL-~0.99

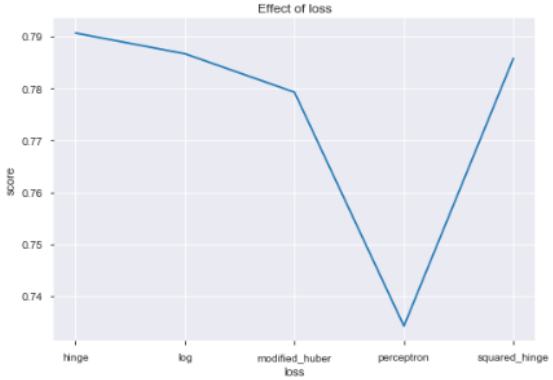
$$F1 = 2 * \frac{precision * recall}{precision * recall}$$

F1 Score :-

DOS-~0.99
PROBE-~0.96
R2L-~0.92
U2R-~0.67
NORMAL-~0.99

9.

```
[128]: [<matplotlib.lines.Line2D at 0x247866f4b08>]
```




In this context, it's showing that a line plot has been created, and the [<matplotlib.lines.Line2D at 0x247866f4b08>] is just the default output that Jupyter Notebook or IPython provides to indicate that a plot has been generated. The actual plot should be displayed in your Jupyter Notebook interface, or the plot may be saved as an image file, depending on your configuration. You can typically view the plot by executing the cell with the plt.plot(x, scores) line in a Jupyter Notebook, and the plot should appear below the cell output.

**7-Conclusion**

Our network intrusion detector model, which has an accuracy of 99% and can be used in an organization for security purposes, can be used here to predict the legitimacy of the packets because we live in a digital age where we frequently use the Internet and connect to networks with ports and services. As a result, there is a high risk of being attacked by a hacker who uses these kinds of attacks to gain access to monitors.
We have presented an overview of IDS detection techniques, strategies, and technologies. Every technique has its advantages and disadvantages, so we should exercise caution when choosing the methods. Consider the pattern-based intrusion detection system (IDS). While it is easy to use and highly efficient in monitoring known attacks, it is not very good at identifying unknown attacks, attacks that are hidden by evasion techniques, and numerous variations of known attacks. Also, a number of rule-based strategies have been put forth to identify unknown attacks. These methods, however, could lead to the issue of hardening and updating the knowledge for given attacks. Heuristic-based techniques also have the advantage of not requiring prior knowledge of attacks, but their high computational complexity prevents them from performing well in real-time applications. Thus, before making any practical use, it is essential to have a thorough understanding of IDSs and application requirements. Furthermore, we suggest a more thorough analysis of IDSs. A summary of the tables and figures makes it easy to understand the overall picture. Additionally, we briefly introduce two well-known, open-source tools for IDS research.
In this work, the Random Forest (RF) method is used to identify four different kinds of attacks: DOS, probe, U2R, and R2L.Ten cross-validation applications were used for classification. The data set is subjected to feature selection in order to eliminate superfluous and pointless characteristics and reduce dimensionality. We used symmetrical uncertainty of attributes to solve the information gain issues. The NSL KDD data set is used to assess the suggested methodology. We evaluated the accuracy, DR, FAR, and MCC of our random forest modeling to those of the j48 classier. Our testing results demonstrate that our suggested strategy increases accuracy, DR, and MCC for four different types of attacks. In order to increase the classifier's accuracy even further, we will use evolutionary computing as a feature selection metric in further work

**8-References**

1. Bolon-Canedo, V., Sanchez-Marono, N., & Alonso-Betanzos, A. (2011). Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset. Expert Systems with Applications, 38(5), 5947-5957.

2.Sekar, R., Guang, Y., Verma, S., & Shanbhag, T. (1999, November). A high-performance network intrusion detection system. In Proceedings of the 6th ACM Conference on Computer and Communications Security (pp. 8-17).

3.Raghunath, B. R., & Mahadeo, S. N. (2008, July). Network intrusion detection system (NIDS). In 2008 First International

Conference on Emerging Trends in Engineering and Technology (pp. 1272-1277). IEEE.

4. Modi, C. N., Patel, D. R., Patel, A., & Rajarajan, M. (2012). Integrating signature apriori based network intrusion detection system (NIDS) in cloud computing. Procedia Technology, 6, 905-912.

5. Samrin, R., & Vasumathi, D. (2017, December). Review on anomaly based network intrusion detection system. In 2017 international conference on electrical, electronics, communication, computer, and optimization techniques (ICEECCOT) (pp. 141-147). IEEE.

6. Garuba, M., Liu, C., & Fraites, D. (2008, April). Intrusion techniques: Comparative study of network intrusion detection systems. In Fifth International Conference on Information Technology: New Generations (itng 2008) (pp. 592-598). IEEE.

7.Shun, J., & Malki, H. A. (2008, October). Network intrusion detection system using neural networks. In 2008 fourth international conference on natural computation (Vol. 5, pp. 242-246). IEEE.

8. Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217.

9. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150.

10. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1), 1-22.