iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VII **Month of publication:** July 2024

DOI: https://doi.org/10.22214/ijraset.2024.63565

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VII July 2024- Available at www.ijraset.com

Machine Learning Approaches for Network Intrusion Detection: An Evaluation of their Efficacy in Bolstering Security

Gaurav Kumar[1], Jawahar Thakur[2]
[1, 2]*Department of computer science, Himachal Pradesh University, Shimla, H.P*

Abstract: ***In today's digital environment, securing networked systems is critical due to the increasing sophistication and frequency of cyberattacks. Network Intrusion Detection Systems (NIDS) are essential for detecting and addressing unauthorized and harmful activities within a network. NIDS function by continuously monitoring network traffic and analyzing data packets for signs of suspicious behavior or known attack patterns. This paper provides an in-depth examination of NIDS, focusing on the application of three machine learning algorithms—K Neighbors Classifier, Logistic Regression, and Random Forest Classifier—to create a robust model for network intrusion detection using the NSL-KDD dataset. Our findings highlight the superior performance of the Random Forest Classifier, which achieved an accuracy of 99.31%, proving its effectiveness in differentiating between normal and malicious traffic. The analysis of ICMP, TCP, and UDP protocols reveals unique attack patterns, underscoring the need for protocol-specific security measures. Additionally, the study emphasizes the importance of integrating NIDS with other security systems for a multi-layered defense strategy and the crucial role of skilled personnel in managing and interpreting NIDS alerts. The results advocate for ongoing innovation and adaptation in NIDS technologies and strategies to effectively counter evolving cyber threats.***

I. INTRODUCTION

In today's digital age, securing network systems is of utmost importance. As cyberattacks become more complex and frequent, organizations must adopt robust security measures to protect their information assets. Network Intrusion Detection Systems (NIDS) are essential in detecting and responding to unauthorized and malicious activities within a network. These systems monitor network traffic and analyze data packets for suspicious behavior or known attack patterns. By examining traffic in real-time, NIDS can identify various attacks, from common threats like malware and phishing to sophisticated intrusions such as zero-day exploits and Advanced Persistent Threats (APTs).

NIDS operate by providing a defense layer at the network level, inspecting packets flowing through the network infrastructure. They use several techniques, including signature-based detection, anomaly-based detection, and stateful protocol analysis. Signature-based detection compares network traffic against a database of known attack signatures, effectively identifying documented threats but struggling with new, unknown attacks. Anomaly-based detection establishes a baseline of normal network behavior, flagging deviations as potential intrusions. This method is particularly useful for detecting novel or polymorphic attacks but can have higher false-positive rates. Stateful protocol analysis looks for deviations from standard protocol behavior, identifying misuse or abnormalities that may indicate an intrusion. Integrating NIDS with other security systems, such as firewalls, Security Information and Event Management (SIEM) systems, and endpoint detection and response (EDR) tools, is crucial for a comprehensive security posture. This integration enables a multi-layered defense strategy, where insights from NIDS are correlated with data from other sources to provide a holistic view of the security landscape. Effective communication and collaboration among these systems enhance detection and response efforts, reducing the time to identify and mitigate threats. Deploying and managing NIDS require skilled personnel who can interpret alerts, fine-tune the system to reduce false positives, and respond appropriately to detected incidents. Training and awareness programs for network administrators and security professionals are essential to maximize NIDS effectiveness. As organizations adopt advanced technologies like cloud computing and the Internet of Things (IoT), the role of NIDS will expand to cover these new environments, necessitating continuous innovation and adaptation in NIDS technology and strategies. Network Intrusion Detection Systems are a vital component of modern cybersecurity frameworks. By continuously monitoring network traffic and identifying suspicious activities, NIDS help organizations defend against a wide range of cyber threats.

©IJRASET: All Rights are Reserved

Despite the challenges, advancements in detection methodologies, integration with other security tools, and the evolution of threat intelligence enhance the capabilities and effectiveness of NIDS, making them indispensable in securing networked environments.

II. METHODOLOGY

A powerful network intrusion detection model can be developed by integrating the strengths of three machine learning algorithms: KNeighborsClassifier, LogisticRegression, and RandomForestClassifier. This ensemble method trains each algorithm on the KDD dataset, which includes features of network traffic and their corresponding classifications as normal or attack types. KNeighborsClassifier is adept at identifying anomalies by comparing them to known attack patterns. LogisticRegression offers a probabilistic assessment of the likelihood of an attack. RandomForestClassifier utilizes multiple decision trees to boost accuracy and minimize overfitting. By combining their outputs, this ensemble model can provide robust and generalized detection of network attacks. Attack flags are crucial in Network Intrusion Detection Systems (NIDS) as they indicate potentially malicious activities within network traffic. These flags, found in packet headers of protocols like TCP, UDP, and ICMP, help NIDS identify abnormal patterns and behaviors that deviate from typical network operations. For example, a surge in TCP SYN packets without corresponding SYN-ACK responses might signal a SYN flood attack, while an unusual number of ICMP echo requests could indicate a Smurf attack. By monitoring these flags, NIDS can detect and alert administrators to various forms of network attacks, such as denial of service, port scanning, and attempts to exploit specific vulnerabilities. This proactive detection capability is vital for addressing threats, reducing the risk of breaches, and maintaining the integrity and availability of network services. The analysis of attack flags enables NIDS to provide robust, real-time defense mechanisms against a wide range of cyber threats, thereby enhancing overall network security.

A. Dataset Used

The NSL-KDD dataset is an enhanced version of the KDD Cup 1999 dataset, designed to improve the evaluation of network intrusion detection systems. It addresses the original dataset's issues, such as redundant records that could skew learning algorithms and lead to inaccurate performance metrics. By reducing these redundancies, the NSL-KDD dataset offers a more realistic and balanced environment for assessment. It includes various types of network traffic data, encompassing normal connections and different attack types, which are categorized into four main classes: Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R). Due to its improved quality and better representation of network traffic behavior, the NSL-KDD dataset has become a widely used benchmark in intrusion detection system research.

B. Improvements Over the KDD'99 Dataset

The NSL-KDD dataset offers several advantages over the original KDD dataset:

1) It eliminates redundant records in the training set, preventing classifiers from being biased towards more frequent records.
2) The proposed test sets do not contain duplicate records, ensuring that performance evaluations are not skewed by methods with higher detection rates for frequent records.
3) The number of records from each difficulty level group is inversely proportional to their presence in the original KDD dataset, allowing for a broader range of classification rates among different machine learning methods. This facilitates more accurate evaluations of various learning techniques.
4) The size of the training and test sets is manageable, enabling comprehensive experiments without the need to sample smaller subsets. This consistency ensures that evaluation results across different research works are comparable.

C. Statistical Observations

One significant deficiency of the KDD dataset is the large number of redundant records, which biases learning algorithms towards frequent records and hinders their ability to learn from infrequent but potentially more harmful records, such as U2R and R2L attacks. Additionally, the presence of these redundant records in the test set biases evaluation results towards methods that perform better on frequent records.

Table 1: Statistics of redundant records in the KDD train set

	Original Records	Distinct Records	Reduction Rate
Attacks	3,925,650	262,178	93.32%
Normal	972,781	812,814	16.44%
Total	4,898,431	1,074,992	78.05%

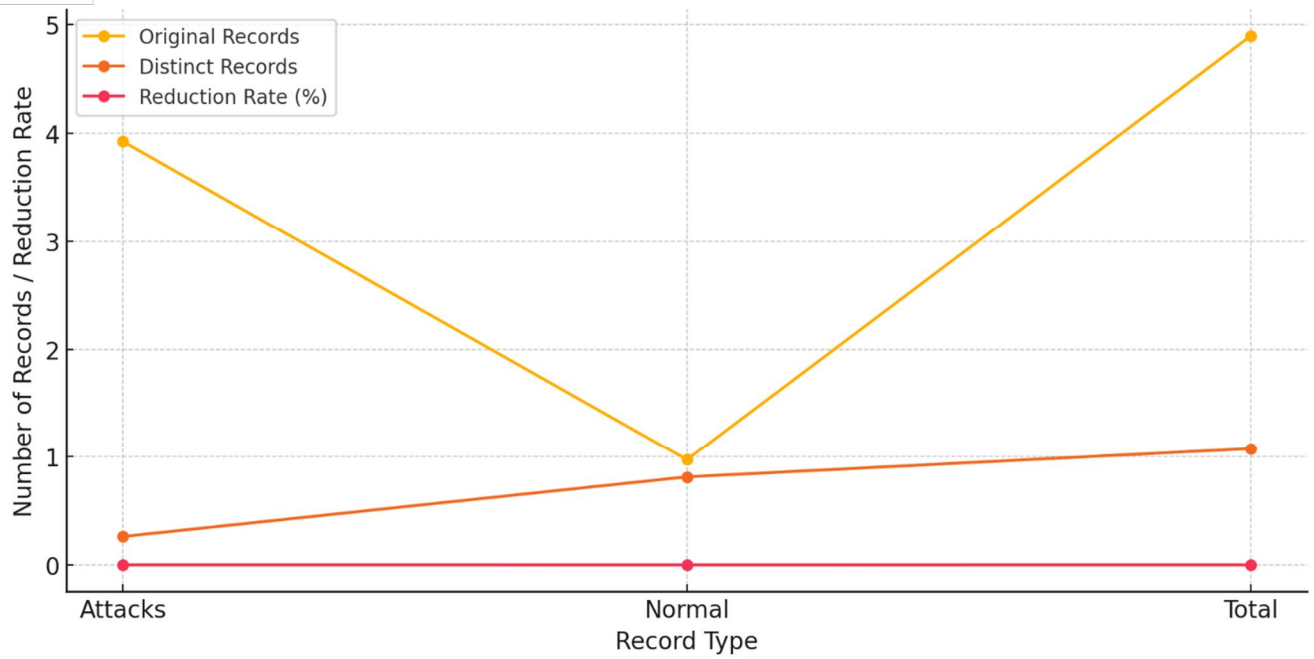


Fig. 1 Statistics of redundant records in the KDD train set

Table 2: Statistics of redundant records in the KDD test set

	Original Records	Distinct Records	Reduction Rate
Attacks	250,436	29,378	88.26%
Normal	60,591	47,911	20.92%
Total	311,027	77,289	75.15%

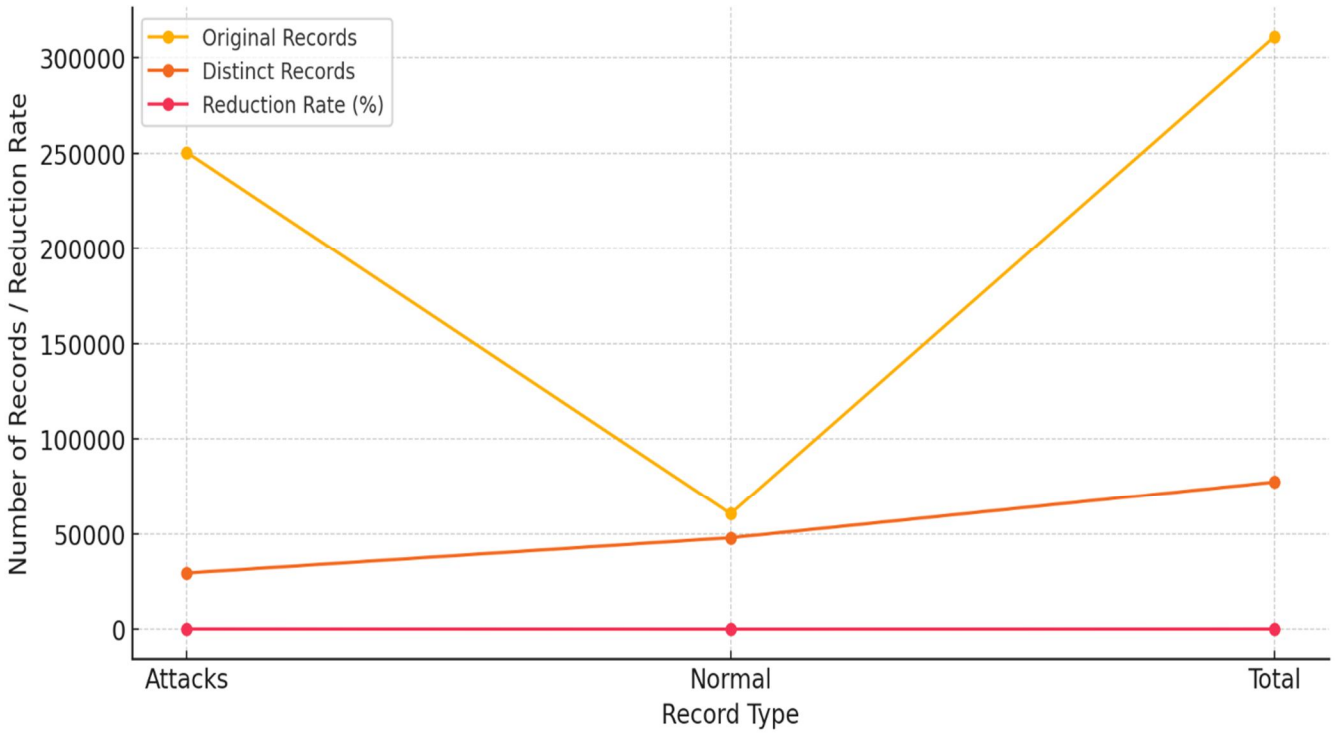


Fig. 2 Statistics of redundant records in the KDD test set

The complexity of the records in the KDD dataset. Remarkably, around 98% of the training set records and 86% of the test set records were accurately classified by all 21 models. For our experiments, we created three smaller subsets from the KDD training set, each containing fifty thousand records. Each model was trained on these subsets. Subsequently, we used the 21 trained models (7 models, each trained 3 times) to classify the records in the entire KDD training and test sets, resulting in 21 predicted labels for each record. We assigned a #successfulPrediction value to each record, initially set to zero. Given that the KDD dataset includes the correct label for each record, we compared the predicted labels from each model with the actual labels, increasing the #successfulPrediction value by one for each match. This allowed us to determine the number of models that accurately labeled each record. The maximum value for #successfulPrediction is 21, indicating that all models correctly predicted the label for that record.

III. EXPERIMENTAL SETUP

This involves examining network traffic data categorized by different protocols and types of attacks. The key findings are visualized using pie charts, which help illustrate the distribution and prevalence of various attack types across different network protocols. This section provides a detailed examination of these findings. The below pie charts shown for different network protocols (ICMP, TCP, and UDP) and the distribution of various types of network traffic or attack flags associated with each protocol.

A. ICMP Traffic Analysis

The ICMP traffic is categorized into normal traffic and various attack types, including Smurf, IPSweep, and Portsweep. The pie chart for ICMP traffic shows that normal diagnostic messages like echo requests and replies constitute a significant portion of the traffic. However, attacks like Smurf, which involves sending ICMP echo requests with a spoofed source IP address, also have a notable presence. IPSweep and Portsweep activities are evident, indicating efforts to map network topology and identify open ports using ICMP.

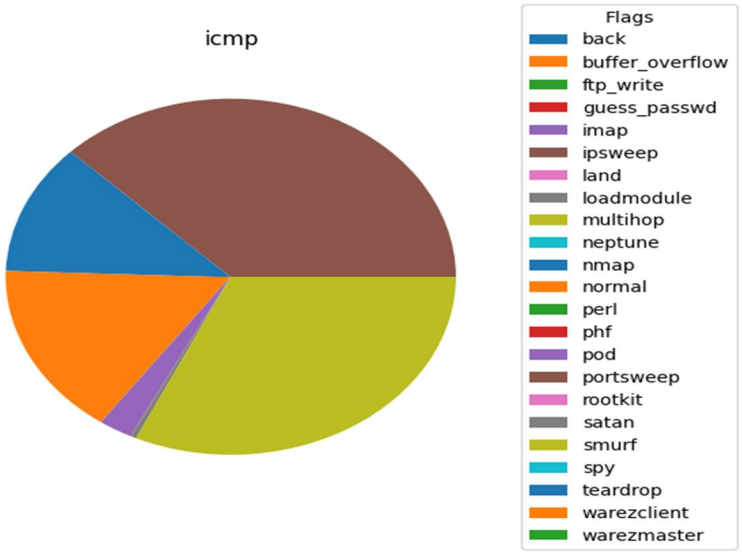


Fig. 3 Distribution of ICMP Traffic by Attack Type

B. TCP Traffic Analysis

TCP traffic analysis reveals a more complex landscape. The normal TCP traffic includes regular web traffic (HTTP/HTTPS), file transfers (FTP), and other standard services. However, the pie chart highlights several attack types, such as Back, Neptune, and Satan, which collectively represent a substantial portion of the traffic.

1) Back Attack: Involves sending continuous SYN packets to a web server, attempting to exploit system vulnerabilities.
2) Neptune Attack: A form of SYN flood attack that disrupts the TCP handshake process by overwhelming the server with connection requests.
3) Satan Attack: Utilizes a network scanner tool to identify system vulnerabilities.

Other attacks like Portsweep, Buffer_overflow, and Teardrop are also present, indicating various methods used to exploit network weaknesses.

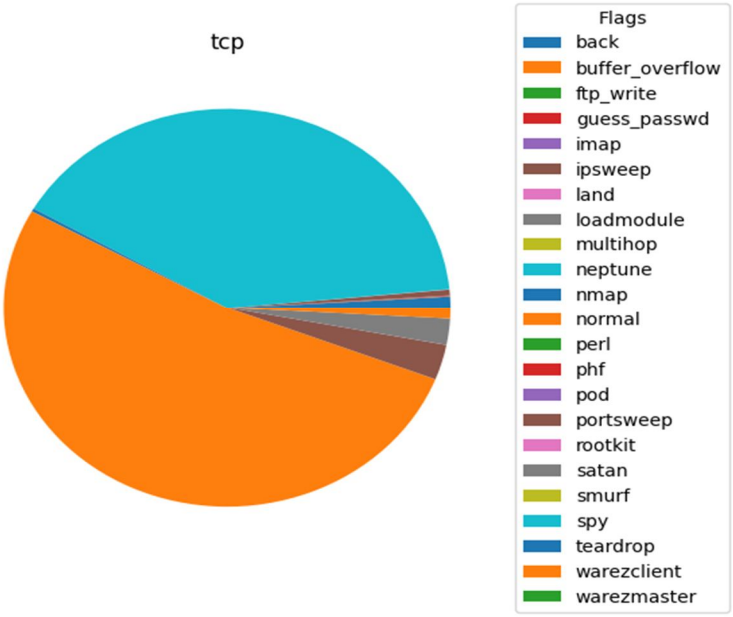


Fig. 4 Distribution of TCP Traffic by Attack Type

C. UDP Traffic Analysis

UDP traffic is typically faster but less reliable than TCP. The normal UDP traffic includes DNS queries and streaming data. The pie chart for UDP traffic shows that attacks like Back and Portsweep are significant. Back in the context of UDP represents overwhelming the target with a continuous stream of packets. The presence of Neptune-like flooding activities in UDP traffic suggests attempts to disrupt services through high-volume packet flows.

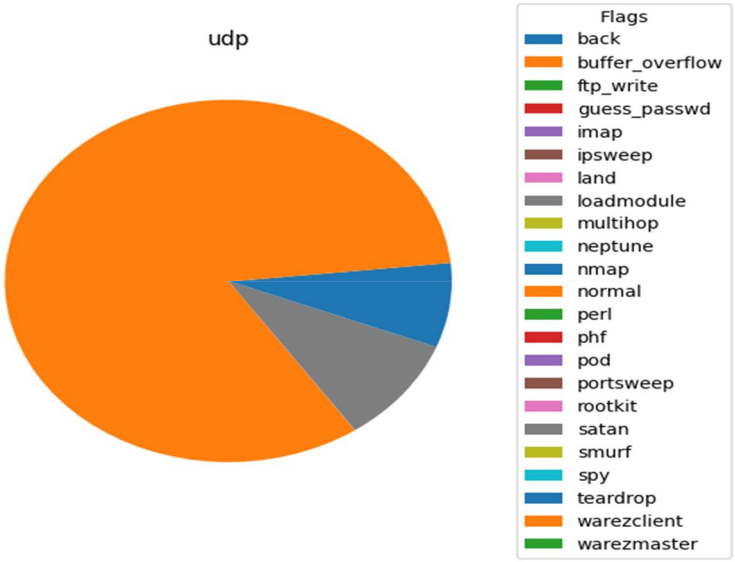


Fig. 5 Distribution of UDP Traffic by Attack Type

D. Comparison of Normal and Attack Traffic

A detailed comparison of normal and attack traffic reveals distinct patterns. In normal traffic, TCP connections are mostly successfully established and terminated without issues, as indicated by the dominance of the SF (Successful Establishment) flag. REJ (Rejected) connections are fewer, aligning with expected network behavior where most connections are legitimate.

In attack scenarios, the S0 flag (SYN packets without SYN-ACK responses) is predominant, a hallmark of SYN flood attacks. The presence of various reset flags (RSTR, RSTO) and other anomalous connection states (S1, S2, S3, OTH) is significantly higher in attack traffic, highlighting disrupted and abnormal connection attempts.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VII July 2024- Available at www.ijraset.com

E. Service Distribution in Normal vs. Attack Traffic

The analysis of service distribution in normal and attack traffic shows a stark contrast. Normal traffic is dominated by HTTP, reflecting typical web browsing and service interactions. Other services like DNS (domain_u), email (smtp), and file transfer (ftp_data) also contribute but to a lesser extent.

The figure 6 below describes the distribution of connection flags in normal traffic.

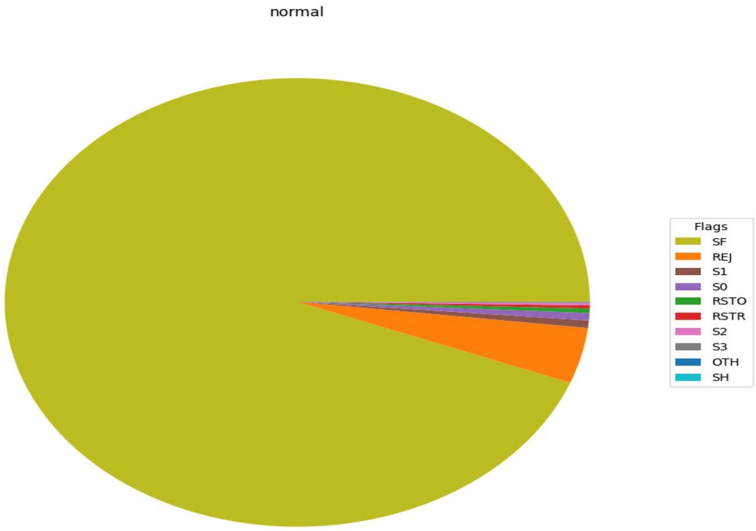


Fig. 6 Distribution of Flags in Normal Dataset

Attack traffic, however, is characterized by a more diverse distribution of services. The "private" service category is particularly dominant, suggesting that attackers often exploit non-standard or custom applications. This indicates that attacks target a wide range of services, exploiting vulnerabilities across different areas, including FTP, Telnet, SMTP, and others.

The figure 7 below describes the distribution of connection flags in ttack traffic.

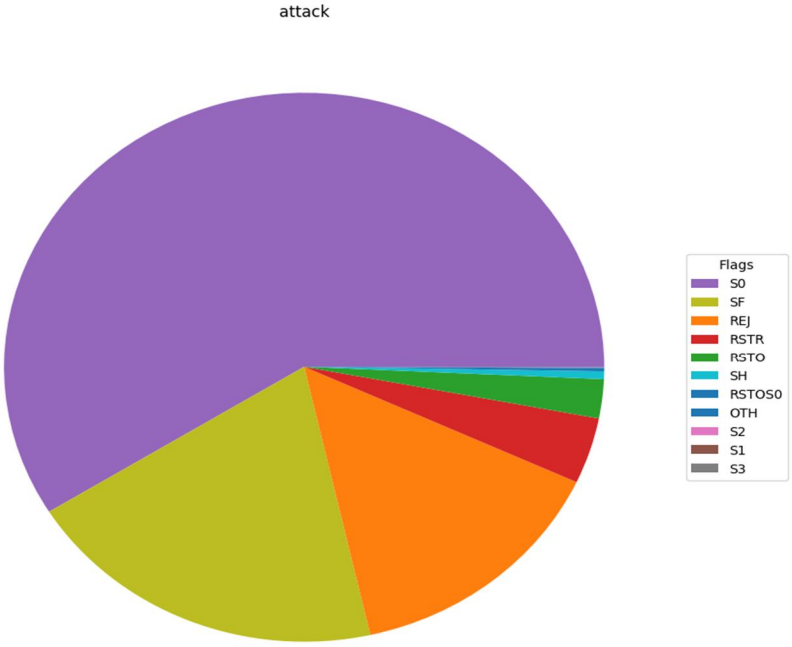


Fig. 7 Distribution of Flags in Attack Dataset

©IJRASET: All Rights are Reserved

IV. SIGNIFICANCE AND IMPLICATIONS

The analysis highlights the importance of monitoring and securing a broad spectrum of network services. Normal traffic patterns provide a baseline for identifying anomalies, while attack traffic patterns reveal common strategies and vectors used by attackers. The findings underscore the need for comprehensive security measures that encompass not only standard services but also custom and less commonly used applications.

The figure 8 below describes the distribution of Flags in NormalMCS And Attack Dataset.

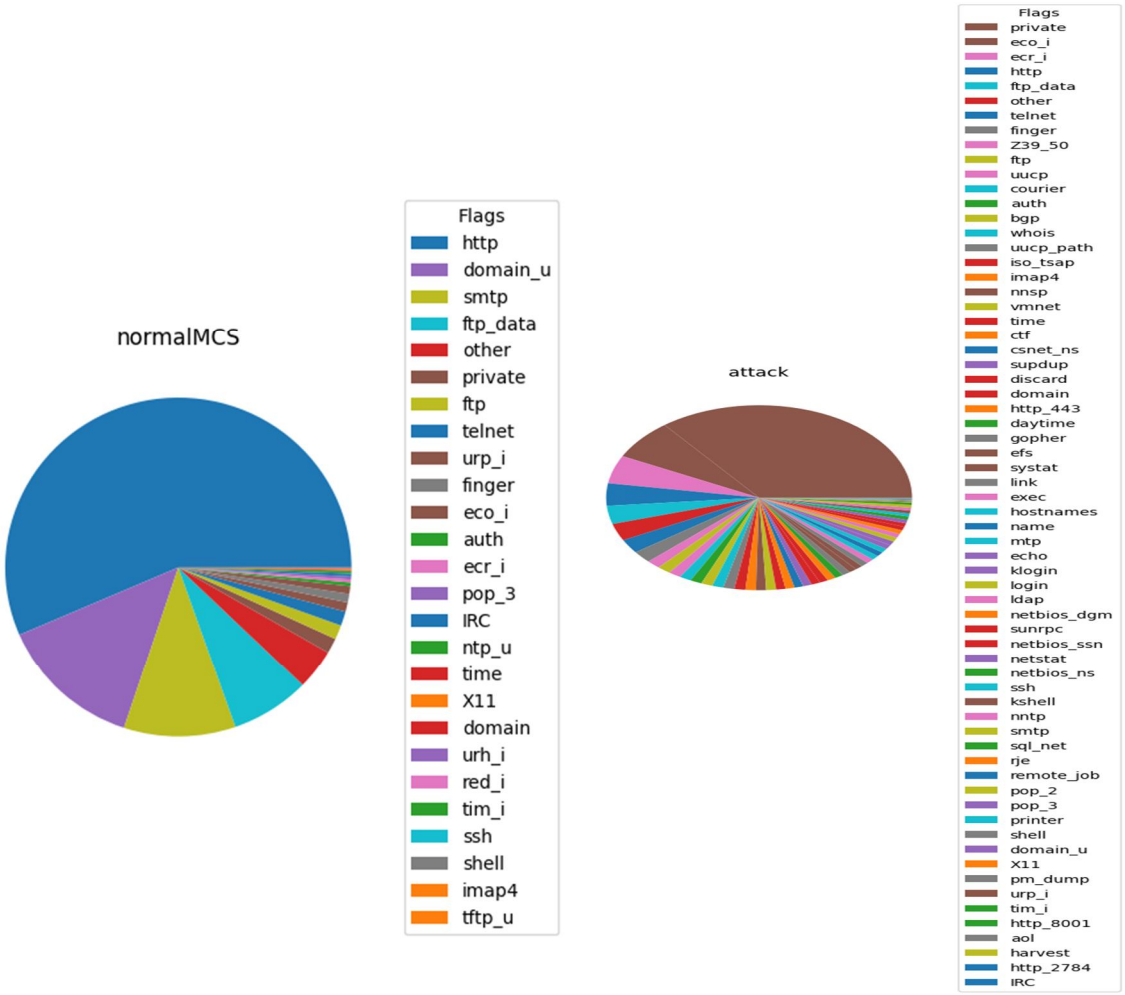


Fig. 8 Distribution of Flags in NormalMCS And Attack Dataset

The stark difference between the two charts highlights how attack traffic is more varied and targets a broader range of services compared to normal traffic. In normal network activity, http is the dominant service, indicating routine web browsing and interactions. However, in attack scenarios, the prevalence of "private" services suggests that attackers frequently target custom or non-standard services, which may be less secure or monitored less closely.

The variety in the attack traffic chart shows that attackers often use multiple vectors to exploit vulnerabilities across different services. This can include services like ftp (file transfer), telnet (remote access), smtp (email), and many others, indicating a comprehensive approach to finding weak points in the network.

By comparing the two pie charts, we can infer that normal traffic is heavily dominated by web traffic, with other services playing smaller roles. In contrast, attack traffic is more evenly distributed among a wide range of services, with a significant focus on custom or less common services. This information is crucial for network administrators and cybersecurity professionals to understand common attack patterns and to implement security measures across a broad spectrum of services to protect against potential threats.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VII July 2024- Available at www.ijraset.com

V. RESULT ANALYSIS

The fig. 9 shows a box plot comparing the performance of three different classifiers: KNeighborsClassifier, LogisticRegression, and RandomForestClassifier. The y-axis represents the performance metric, likely accuracy, with values ranging from 0.4 to 1.0.

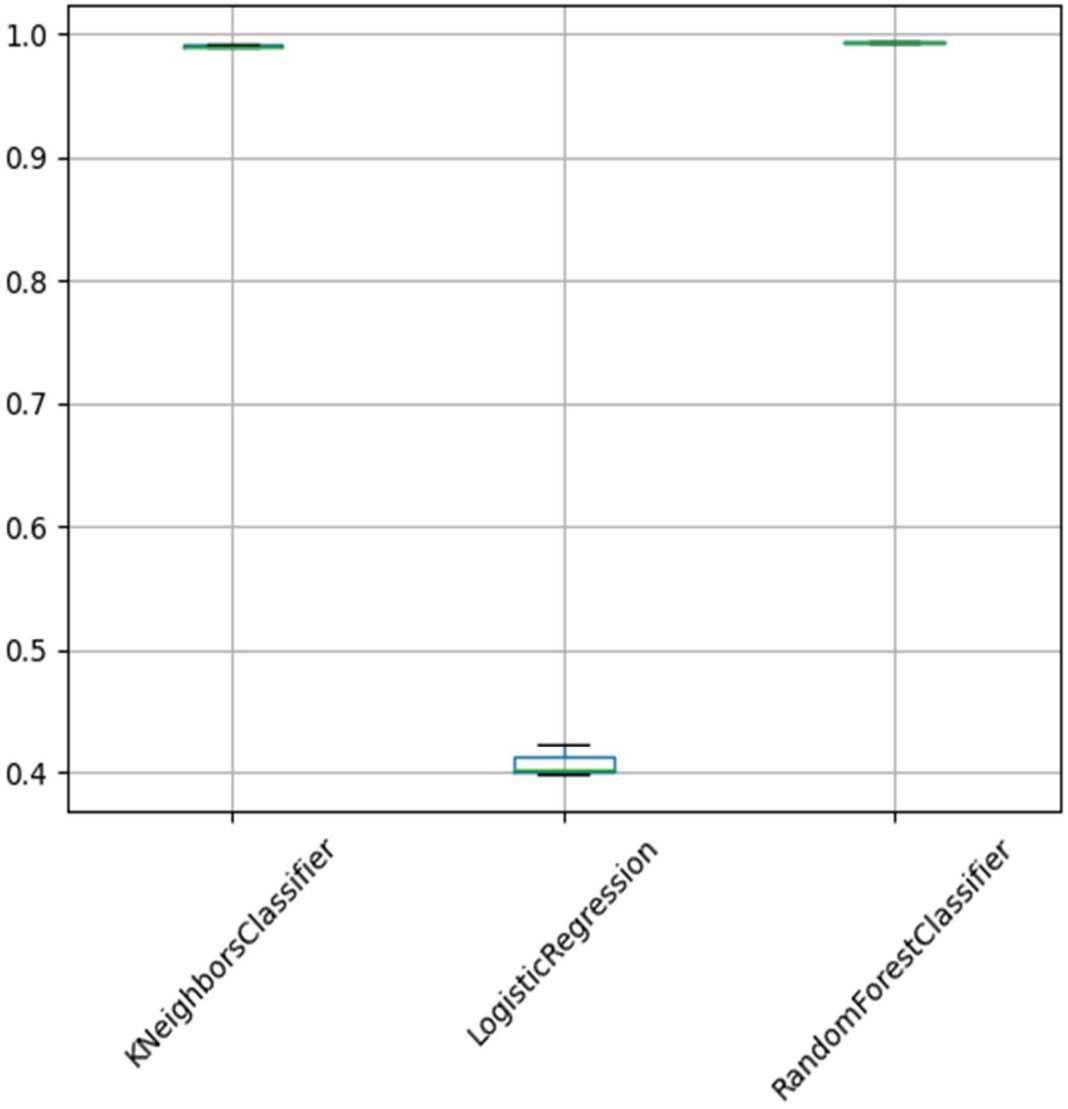


Fig. 9 Box Plot of Classifier Performance Metrics

1) *KNeighborsClassifier:* The box plot for this classifier shows a performance metric very close to 1.0, indicating high accuracy. The small interquartile range suggests that the performance is consistently high across different runs or folds.
2) *LogisticRegression:* The box plot for this classifier shows a significantly lower performance, centered around 0.4. The interquartile range is also small, indicating that the performance is consistently low.
3) *RandomForestClassifier:* Similar to KNeighborsClassifier, the box plot for this classifier shows performance close to 1.0, indicating high accuracy. The small interquartile range again suggests consistent performance.

Overall, the box plot reveals that both KNeighborsClassifier and RandomForestClassifier perform very well with high and consistent accuracy, while LogisticRegression performs poorly with low and consistent accuracy. The consistency across the classifiers can be inferred from the narrowness of the boxes, indicating low variability in their performance. The tilt of the x-axis labels suggests a standard formatting to make the labels more readable. Fig. 10 shows the confusion matrix for classification model

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VII July 2024- Available at www.ijraset.com

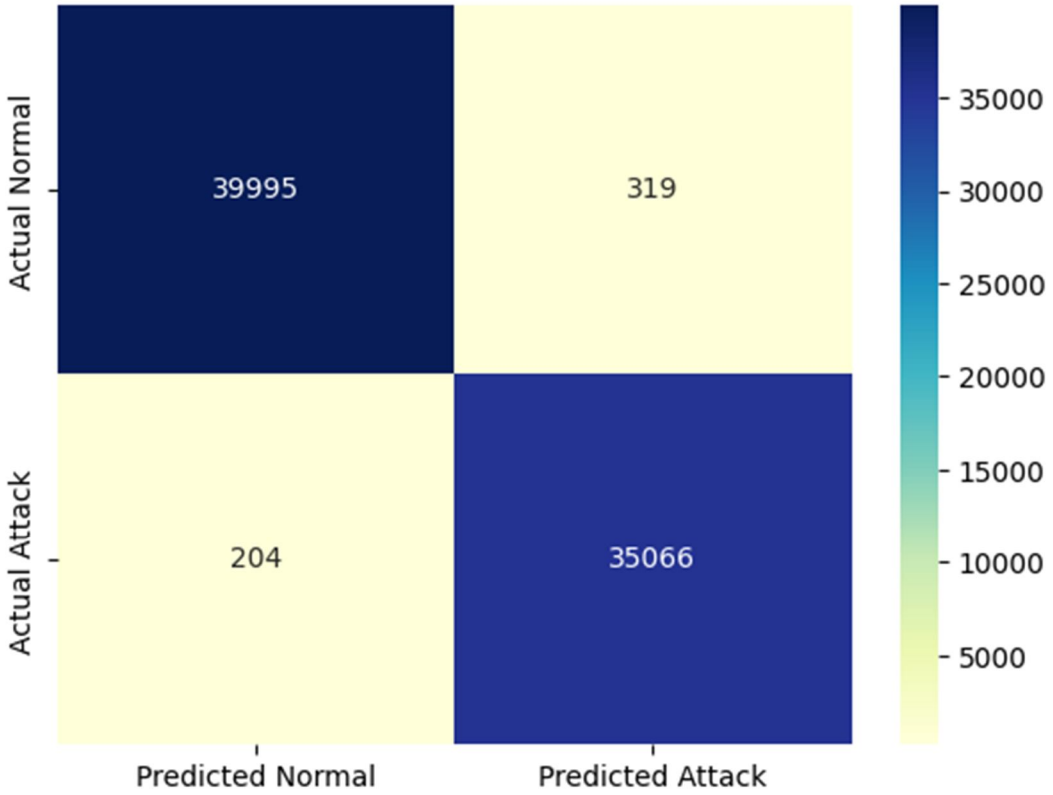


Fig. 10 Confusion Matrix for Classification Model

The figure 10 is a confusion matrix used to assess the performance of a classification model, probably in identifying normal and attack records in a dataset. The matrix is a 2x2 grid that shows the actual versus predicted classifications:

- True Positives (Predicted Attack, Actual Attack): The bottom-right cell shows 35,066 instances where the model correctly identified attacks.
- True Negatives (Predicted Normal, Actual Normal): The top-left cell shows 39,995 instances where the model correctly identified normal behavior.
- False Positives (Predicted Attack, Actual Normal): The top-right cell shows 319 instances where the model incorrectly identified normal behavior as attacks.
- False Negatives (Predicted Normal, Actual Attack): The bottom-left cell shows 204 instances where the model incorrectly identified attacks as normal behavior.

Darker colors in the matrix indicate higher counts. The model demonstrates high accuracy for both normal and attack classifications, with high numbers of true positives and true negatives and relatively low false positives and false negatives. This suggests that the model effectively distinguishes between normal and attack records, which aligns with the strong performance observed in the KNeighborsClassifier and RandomForestClassifier in the previous analysis.

The RandomForestClassifier achieved an impressive accuracy of 0.993, or 99.31%, during its evaluation. This high accuracy indicates the model's exceptional ability to correctly classify instances in the dataset, distinguishing effectively between normal and attack records.

The confusion matrix from the second uploaded image further illustrates this performance. It shows that out of 40,514 actual normal instances, 39,995 were correctly classified, while only 319 were misclassified as attacks. Similarly, out of 35,270 actual attack instances, 35,066 were correctly identified, with only 204 incorrectly labeled as normal. This high true positive and true negative rate confirms the model's robustness and precision.

Additionally, the box plot from the first uploaded image highlights the RandomForestClassifier's consistency and reliability. Unlike the LogisticRegression, which had low and consistent accuracy around 40%, the RandomForestClassifier consistently achieved near-perfect scores close to 1.0 across different runs or folds.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This consistency in high performance aligns with the observed confusion matrix results, reinforcing the RandomForestClassifier's suitability for the binary classification task of identifying normal and attack records.

Overall, an accuracy of 99.31% not only demonstrates the model's effectiveness in this specific task but also underscores its potential for application in real-world scenarios where distinguishing between normal and malicious activity is critical.

VI. CONCLUSION

The comprehensive analysis conducted in this study underscores the critical role of Network Intrusion Detection Systems (NIDS) in safeguarding networked environments from a plethora of cyber threats. Through the meticulous evaluation of network traffic, the NIDS framework demonstrates its efficacy in identifying both common and sophisticated attacks. The study's findings highlight that robust security measures at the network level are indispensable for modern organizations, given the escalating complexity and frequency of cyberattacks. One of the pivotal discoveries of this research is the exceptional performance of the RandomForestClassifier. With an impressive accuracy rate of 99.31%, this classifier not only showcases its capability to distinguish between normal and malicious traffic but also emphasizes the importance of leveraging advanced machine learning algorithms in NIDS. The consistency of the RandomForestClassifier, as evidenced by its narrow interquartile range, further validates its reliability across different datasets and experimental setups.

The detailed traffic analysis, encompassing ICMP, TCP, and UDP protocols, reveals distinct patterns that are crucial for developing targeted defense strategies. The dominance of certain attack types in specific protocols, such as the prevalence of Smurf and Portsweep attacks in ICMP traffic, or the diverse array of attacks in TCP traffic, underscores the necessity for protocol-specific security measures. The nuanced understanding of these patterns enables more precise anomaly detection and mitigation strategies, enhancing the overall effectiveness of NIDS.

Moreover, the study's examination of service distribution in normal versus attack traffic provides valuable insights into attacker behavior. The stark contrast between the service usage in normal traffic and the diverse exploitation of services in attack traffic, particularly the frequent targeting of non-standard services, highlights the attackers' adaptive strategies. This insight is critical for network administrators to prioritize the security of lesser-monitored services and implement comprehensive security policies. The practical implications of these findings are profound. Network administrators and cybersecurity professionals must not only focus on deploying robust NIDS but also ensure continuous monitoring and adaptation of these systems to evolving threats. Integrating Network Intrusion Detection Systems (NIDS) with other security measures like firewalls and Security Information and Event Management (SIEM) tools is crucial for a robust defense strategy. This integration offers a comprehensive view of the security landscape, enhancing the efficiency of threat detection and response.

Additionally, the effectiveness of NIDS heavily relies on the expertise of the personnel managing them. It is vital for network administrators to receive continuous training and skill development to accurately interpret alerts, refine detection algorithms, and respond promptly to incidents. As technology evolves and the network environment becomes more complex with the rise of IoT and cloud computing, the importance of skilled personnel in managing NIDS will grow.

In conclusion, this study reinforces the indispensable role of Network Intrusion Detection Systems in modern cybersecurity frameworks. By providing a detailed analysis of network traffic and demonstrating the efficacy of advanced machine learning algorithms like RandomForestClassifier, the research offers valuable contributions to the field. The insights gained from the traffic and service distribution analyses inform the development of more effective security measures, emphasizing the need for continuous innovation and adaptation in NIDS technology and strategies. As cyber threats continue to evolve, so too must the defenses, ensuring that organizations can protect their critical information assets against an ever-changing threat landscape.

This underscores the study's key findings on the critical role of NIDS in modern cybersecurity, the effectiveness of machine learning classifiers, and the necessity for ongoing adaptation and expert management to safeguard networked environments.

REFERENCES

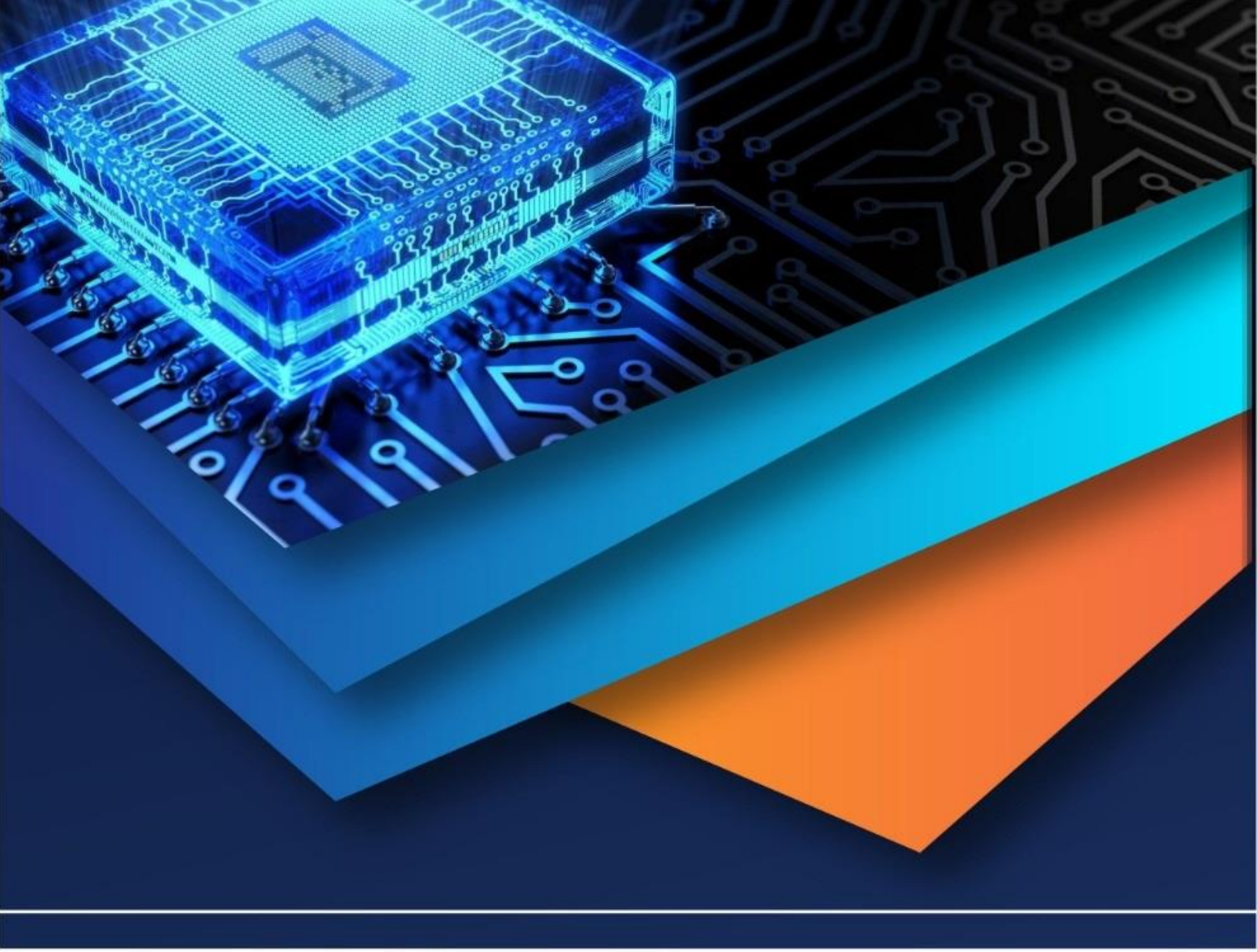
[1] Adeshina, A. M. (2023). Prediction of Diabetes Mellitus using Machine Learning Algorithms: Comparative Analysis of K-Nearest Neighbor, Random Forest and Logistic Regression. SLU Journal of Science and Technology, 205–213. https://doi.org/10.56471/slujst.v6i.319

[2] Akhtar, N., & Mian, A. (2018). Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. IEEE Access, 6, 14410–14430. https://doi.org/10.1109/ACCESS.2018.2807385

[3] Alshamy, R., & Ghurab, M. (2020). A Review of Big Data in Network Intrusion Detection System: Challenges, Approaches, Datasets, and Tools. International Journal of Computer Sciences and Engineering.

[4] Anjum, N., & Chowdhury, M. R. (2024). International Journal of Advanced Research in Computer and Communication Engineering. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4847308

[5] Esmaeili, M., Goki, S. H., Masjidi, B. H. K., Sameh, M., Gharagozlou, H., & Mohammed, A. S. (2022). ML-DDoSnet: IoT Intrusion Detection Based on Denial-of-Service Attacks Using Machine Learning Methods and NSL-KDD. Wireless Communications and Mobile Computing, 2022, 1–16. https://doi.org/10.1155/2022/8481452
[6] Ibitoye, O., Abou-Khamis, R., Shehaby, M. el, Matrawy, A., & Shafiq, M. O. (2023). The Threat of Adversarial Attacks on Machine Learning in Network Security—A Survey (arXiv:1911.02621). arXiv. http://arxiv.org/abs/1911.02621
[7] Iqbal, M. S., Mukhtar, R., Saleem, M., Kamran, M., Hussain, M., Ali, S. Y., & Umair, M. (2024). Comparative Analysis of Machine Learning Algorithms for Breast Cancer Detec- tion: A Study of Support Vector Classification, Logistic Regression, and K-Nearest Neighbors. Biomedical Informatics.
[8] Kumar, S., Gupta, S., & Arora, S. (2021). Research Trends in Network-Based Intrusion Detection Systems: A Review. IEEE Access, 9, 157761–157779. https://doi.org/10.1109/ACCESS.2021.3129775
[9] Liang, X., & Xiao, Y. (2013). Game Theory for Network Security. IEEE Communications Surveys & Tutorials, 15(1), 472–486. https://doi.org/10.1109/SURV.2012.062612.00056
[10] Liu, J., Kantarci, B., & Adams, C. (2020). Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset. Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, 25–30. https://doi.org/10.1145/3395352.3402621
[11] McHugh, J. (2000). Testing Intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security, 3(4), 262–294. https://doi.org/10.1145/382912.382923
[12] Meena, G., & Choudhary, R. R. (2017). A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA. 2017 International Conference on Computer, Communications and Electronics (Comptelix), 553–558. https://doi.org/10.1109/COMPTELIX.2017.8004032
[13] Pawar, M. V., & Anuradha, J. (2015). Network Security and Types of Attacks in Network. Procedia Computer Science, 48, 503–506. https://doi.org/10.1016/j.procs.2015.04.126
[14] Sarhan, M., Layeghy, S., & Portmann, M. (2022). Towards a Standard Feature Set for Network Intrusion Detection System Datasets. Mobile Networks and Applications, 27(1), 357–370. https://doi.org/10.1007/s11036-021-01843-0
[15] Sawant, N., & Khadapkar, D. R. (2022). Comparison of the performance of GaussianNB Algorithm, the K Neighbors Classifier Algorithm, the Logistic Regression Algorithm, the Linear Discriminant Analysis Algorithm, and the Decision Tree Classifier Algorithm on same dataset. International Journal for Research in Applied Science and Engineering Technology, 10(12), 1654–1665. https://doi.org/10.22214/ijraset.2022.48311
[16] Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majitar, Sikkim, India, Gurung, S., Kanti Ghose, M., & Subedi, A. (2019). Deep Learning Approach on Network Intrusion Detection System using NSL-KDD Dataset. International Journal of Computer Network and Information Security, 11(3), 8–14. https://doi.org/10.5815/ijcnis.2019.03.02
[17] Simmonds, A., Sandilands, P., & Van Ekert, L. (2004). An Ontology for Network Security Attacks. In S. Manandhar, J. Austin, U. Desai, Y. Oyanagi, & A. K. Talukder (Eds.), Applied Computing (Vol. 3285, pp. 317–323). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-30176-9_41
[18] Tama, B. A., Comuzzi, M., & Rhee, K.-H. (2019). TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System. IEEE Access, 7, 94497–94507. https://doi.org/10.1109/ACCESS.2019.2928048
[19] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 1–6. https://doi.org/10.1109/CISDA.2009.5356528
[20] Thomas, R., & Pavithran, D. (2018). A Survey of Intrusion Detection Models based on NSL-KDD Data Set. 2018 Fifth HCT Information Technology Trends (ITT), 286–291. https://doi.org/10.1109/CTIT.2018.8649498
[21] Wu, Y., Wei, D., & Feng, J. (2020). Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey. Security and Communication Networks, 2020, 1–17. https://doi.org/10.1155/2020/8872923



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)