iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68882

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 13 Issue IV Apr 2025- Available at www.ijraset.com

Enhanced Intrusion Detection System Using Machine Learning

Anbarasan R[1], Aditya T[2], Ayub Ahmed A[3], Mr. T. Umamahesh[4], Dr.R.Shobarani[5]
[1, 2, 3]*Student,* [4]*Asst Professor,* [5]*Professor, Department of Computer Science, Dr.M.G.R. Educational and Research Institute, India*

Abstract: ***The evolving cybersecurity landscape demands intrusion detection systems capable of identifying diverse attack patterns across network and application layers. This study addresses limitations in current benchmark datasets by enhancing the CICIDS-2017 dataset through systematic incorporation of multiple attack variants, including web-based threats like Cross-Site Scripting alongside its existing network attack profiles. Our methodology combines realistic attack simulation with rigorous feature engineering to maintain dataset integrity while expanding its threat coverage.We train and evaluate multiple algorithms, selecting the most effective approach based on comprehensive evaluation metrics. The resulting model demonstrates strong capabilities in identifying both traditional network intrusions and contemporary attack patterns. Particular attention is given to maintaining low false positive rates while ensuring broad threat coverage.***
Index Terms: Computer & Network Security, Intrusion Detection, Intrusion Detection System, Cybersecurity, Threat Detection, machine learning

I. INTRODUCTION

The CICIDS2017 dataset has become a cornerstone in the domain of network intrusion detection research, serving as a benchmark for evaluating the efficacy of various machine learning (ML) techniques in identifying and classifying malicious network traffic. Its significance stems from its comprehensive inclusion of both contemporary attack vectors and realistic benign background traffic, offering a more representative scenario compared to earlier datasets. Consequently, a substantial body of research has emerged, leveraging the labeled network flow data within CICIDS2017 to develop and assess a wide spectrum of ML-based Intrusion Detection Systems (IDS). These studies typically focus on training supervised learning models to categorize network connections as either normal or indicative of specific attack types, encompassing prevalent threats such as Brute Force attacks targeting sensitive protocols like FTP and SSH, a range of Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks aimed at disrupting service availability, various Web Attacks exploiting vulnerabilities in web applications, stealthy Infiltration attempts designed to compromise system integrity and confidentiality, the coordinated malicious activities orchestrated by Botnets, and the exploitation of specific software vulnerabilities like Heartbleed [1, 2]

II. RELATED WORK

Within the realm of traditional machine learning, a diverse array of algorithms has been applied and evaluated on the CICIDS2017 dataset. Tree-based methods, including Random Forests and Gradient Boosting Machines, have consistently demonstrated strong performance due to their inherent ability to handle high-dimensional datasets with complex feature interactions and their capacity to identify salient features indicative of malicious behavior [3, 4]. Support Vector Machines (SVMs), known for their effectiveness in high-dimensional spaces and their ability to find optimal separating hyperplanes between different classes of data, have also been extensively investigated for their potential in accurately classifying network flows within CICIDS2017 [5]. Furthermore, simpler yet computationally efficient algorithms such as K-Nearest Neighbors (KNN) and Naive Bayes have often been utilized as baseline models, providing a point of comparison for more sophisticated techniques and sometimes being integrated into hybrid or ensemble architectures to capitalize on their specific strengths in certain aspects of the classification task [6]. The rigorous evaluation of these diverse machine learning models on the CICIDS2017 dataset typically involves employing a suite of performance metrics to provide a comprehensive assessment of their detection capabilities. These metrics commonly include accuracy (the overall proportion of correctly classified instances), precision (the ratio of correctly identified malicious instances to the total number of instances classified as malicious, indicating the model's ability to avoid false positives), recall (the ratio of correctly identified malicious instances to the total number of actual malicious instances, indicating the model's ability to detect a high proportion of threats), and the F1-score (the harmonic mean of precision and recall, offering a balanced measure of performance, particularly crucial in datasets with imbalanced class distributions like CICIDS2017) [7, 8].

©IJRASET: All Rights are Reserved

Recognizing the intricate and often non-linear nature of network traffic patterns, a significant and growing body of research has also focused on the application of deep learning (DL) methodologies to the CICIDS2017 dataset. Deep learning models, including Multi-Layer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), possess the inherent capability to automatically learn complex hierarchical features directly from the raw or pre-processed network flow data, potentially overcoming the limitations associated with manual feature engineering required by traditional machine learning approaches [9, 10]. CNNs, originally designed for image processing tasks, have been adapted to the analysis of network traffic data by treating sequences of flow features as one-dimensional or two-dimensional "images," enabling them to capture local patterns and dependencies that might be indicative of malicious activity. RNNs, particularly the Long Short-Term Memory (LSTM) variant, are exceptionally well-suited for processing sequential data and can effectively model the temporal dynamics within network flows, which is particularly relevant for detecting attacks that unfold over time, such as sophisticated infiltration attempts or the sustained communication patterns associated with botnet command and control [11, 12]. Autoencoders, another class of deep learning models, have been explored for their utility in anomaly detection within the context of CICIDS2017. These models learn a compressed representation of normal network traffic and can subsequently identify deviations from this learned representation as potential anomalies or attacks [13]. The successful application of deep learning to the CICIDS2017 dataset often necessitates careful consideration of network architecture design, meticulous hyperparameter tuning, and the implementation of strategies to mitigate the risk of overfitting on the large and complex dataset.

A persistent challenge encountered in the analysis of the CICIDS2017 dataset, which mirrors the inherent characteristics of real-world network traffic, is the significant issue of class imbalance, where the volume of benign network flows vastly exceeds the number of malicious flows. This skewed distribution can lead to the development of biased machine learning models that exhibit high performance on the majority class (benign traffic) but demonstrate poor detection rates for the minority attack classes, which are, by definition, the primary focus of intrusion detection efforts. To effectively address this challenge, researchers have investigated and implemented a variety of techniques tailored to handle imbalanced datasets. These techniques include oversampling methods, such as the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic instances of the minority classes to balance the class distribution; undersampling methods, which aim to reduce the number of instances in the majority class; and cost-sensitive learning approaches, which assign higher misclassification costs to instances of the minority classes during the model training process, thereby encouraging the model to pay greater attention to their correct classification [14, 15]. Ensemble learning techniques, which combine the predictions of multiple individual classifiers to make a final prediction, have also proven to be effective in improving the robustness and generalization ability of IDS models trained on the imbalanced CICIDS2017 dataset, often leading to enhanced detection performance and a reduction in the impact of class imbalance [16, 17]. Common ensemble methods explored in this context include Bagging, Boosting, and Stacking.

Furthermore, the crucial role of feature selection and engineering in the development of effective machine learning-based IDS for the CICIDS2017 dataset cannot be overstated. Researchers have employed a range of feature selection techniques, including Information Gain, Chi-squared test, and correlation analysis, to identify the most informative features within the network flow data that contribute significantly to the accurate discrimination between benign and malicious traffic [18, 19]. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), have also been applied to reduce the number of features while preserving the majority of the variance in the data, potentially leading to improved model efficiency, reduced computational complexity, and a mitigation of the risk of overfitting. In addition to selecting existing features, the process of feature engineering, which involves creating new features based on domain knowledge and statistical analysis of the existing attributes, has been explored to capture more subtle and potentially more indicative characteristics of network traffic that might signal malicious activity [20]. The impact of different feature subsets and various feature engineering strategies on the performance of diverse machine learning models when applied to the CICIDS2017 dataset has been a recurring and important theme in the research literature.

A significant portion of the research involving the CICIDS2017 dataset has also been dedicated to comparative studies that rigorously evaluate the performance of different machine learning algorithms and deep learning architectures in the context of intrusion detection. These comparative analyses aim to identify the most effective approaches for detecting various attack types present in the dataset and to provide valuable insights into the relative strengths and weaknesses of different techniques when applied to this specific benchmark [21, 22]. Such comparisons often consider a multitude of factors beyond just detection accuracy, including the false positive rate, computational efficiency of the models, and their ability to generalize effectively to unseen network traffic patterns. The findings from these comparative studies serve as crucial guidance for researchers and practitioners in the field, informing the selection of appropriate machine learning techniques for the development of robust and effective intrusion detection systems.

In conclusion, the CICIDS2017 dataset has undeniably played a pivotal role in driving advancements in machine learning-based intrusion detection research. The extensive body of work conducted on this dataset has explored a wide array of machine learning and deep learning algorithms, addressed critical challenges such as class imbalance and the importance of feature selection and engineering, and provided valuable comparative insights into the effectiveness of different approaches for identifying the diverse range of network attacks it encompasses. The ongoing research utilizing the CICIDS2017 dataset continues to be instrumental in the pursuit of more intelligent, adaptive, and ultimately more effective security systems capable of detecting and mitigating the ever-evolving landscape of cyber threats.

III. METHDOLOGY

The methodology section outlines the systematic approach to simulate a web-based XSS attack, capture network traffic, generate a dataset, integrate it with CICIDS-2017, and train a machine learning model for intrusion detection. This project combines virtualized environments, network analysis, and predictive modeling. The process is divided into distinct phases: environment setup, attack simulation, traffic capture, dataset generation, dataset integration, and model training.

A. Environment Setup Methodology

The initial stage involves creating a controlled environment for attack simulation and data collection. VirtualBox is used to configure two virtual machines: Kali Linux and Linux Mint. Kali Linux, a penetration testing platform, is allocated 2 GB RAM and 20 GB storage, serving as the attacker. Linux Mint, hosting a vulnerable web server, is assigned identical resources. Both machines are connected via a host-only network to ensure isolation and enable communication. IP addresses are assigned, such as 192.168.56.101 for Kali and 192.168.56.102 for Mint. Connectivity is confirmed using ping commands to validate the setup.

B. Attack Simulation

Attack simulation focuses on generating malicious XSS traffic. On Linux Mint, Apache2 and PHP are installed to host a webpage with a vulnerable input field that echoes user data without sanitization. From Kali Linux, a browser submits an XSS payload, such as script alert XSS script, to exploit the vulnerability. Multiple submissions are performed to produce sufficient network activity. This phase ensures the system mimics real-world attack scenarios for analysis.

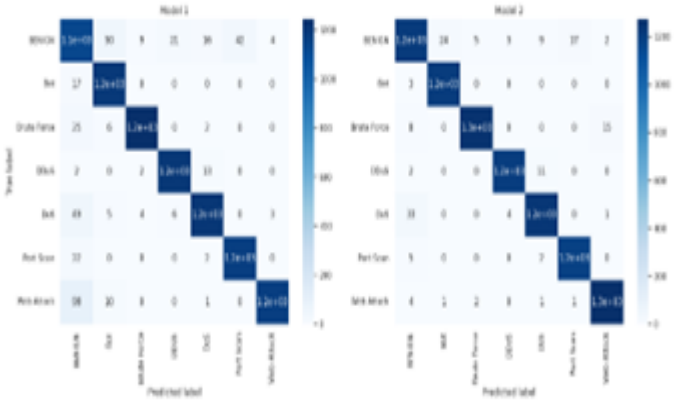


Fig. 1. Predicted Label

C. Network Traffic Capture

Network traffic capture involves recording packets during the XSS attack. Wireshark is installed on Kali Linux and configured to monitor the host-only interface. Capture begins before the attack and ends after adequate data is collected, typically 5 to 10 minutes. The resulting packets are saved as a pcap file, named xss_attack.pcap, providing raw data for subsequent processing.

D. Dataset Generation

Dataset generation transforms captured traffic into a structured format. CICFlowMeter is installed on Kali Linux and used to convert the pcap file into a CSV, named xss_flows.csv, containing over 80 flow features like packet length and flow duration. A label, XSS, is added to identify the attack type. This phase ensures compatibility with machine learning algorithms by producing clean, relevant data.

E. Dataset Integration

Dataset integration merges the XSS data with CICIDS-2017, an established intrusion detection dataset. Using Python and pan-das, the xss_flows.csv file is appended to a CICIDS-2017 CSV, such as Wednesday-workingHours.pcap_ISCX.csv. The combined data-set, saved as updated_cicids_2017.csv, aligns columns and includes the new XSS class. This step enhances the dataset for broader attack detection.

F. Model Training and Evaluation

Model training employs scikit-learn to build a Random Forest Classifier. The updated dataset is preprocessed by removing miss-ing values and replacing infinite values with zeros. Features are separated from the label column, and the data is split into 70 percent training and 30 percent testing sets. The model, configured with 100 trees, is trained on the training set. Performance is assessed using accuracy and classification metrics like precision and recall, providing insights into detection capabilities.

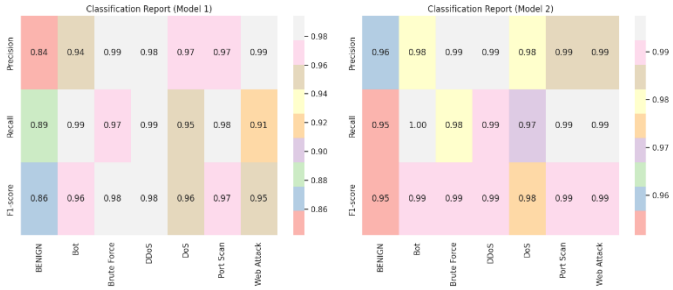


Fig. 2. Classification report

IV. RESULTS

After training the RANDOM FOREST model, the following results were obtained: Model Accuracy: The model with a test accura-cy of 0.98, the model demonstrated a comparatively high capacity to forecast the risk of Intrusion detection. The accuracy for the minor-ityclass (indicating intrusion detection) was lower, highlighting the challenge of class imbalance. Report on Classification: For both the non-intrusion and intrusion classifications, precision, recall, and F1 score were calculated, providing insights into the accuracy of each class's classification by the model. Cross-Validation Results: Cross-validation was used to assess the model's per-formance, and the mean recal score was 0.982, reflecting a moderately good ability to distinguish between the classes.

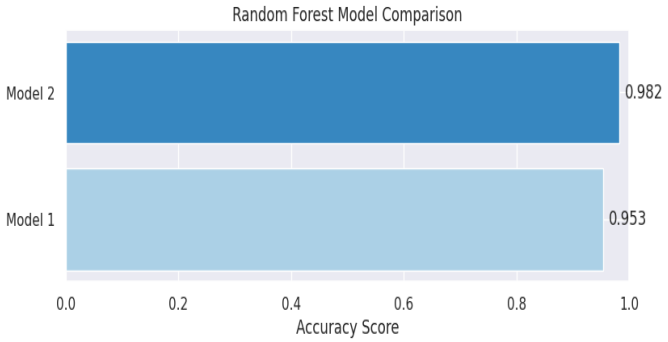


Fig. 3. Accuracy Score

The findings imply that RANDOM FOREST is a useful model for estimating the risk of intrusion. The good classification per-formance and comparatively high accuracy demonstrate that machine learning can be an effective tool in the cyber security industry for forecasting intrusion detections. However, some limitations were observed, particularly with the class imbalance issue, where the model performed better at predicting the majority class. Further steps to address this imbalance, such as using oversampling techniques or adjusting class weights, could potentially improve performance for the minority class.

The results section presents the outcomes of the project, detailing the effectiveness of the XSS attack simulation, the quality of the generated dataset, the integration with CICIDS-2017, and the performance of the Random Forest model in detecting network intru-sions. Each phase of the methodology produced measurable outputs, analyzed to evaluate the system's success

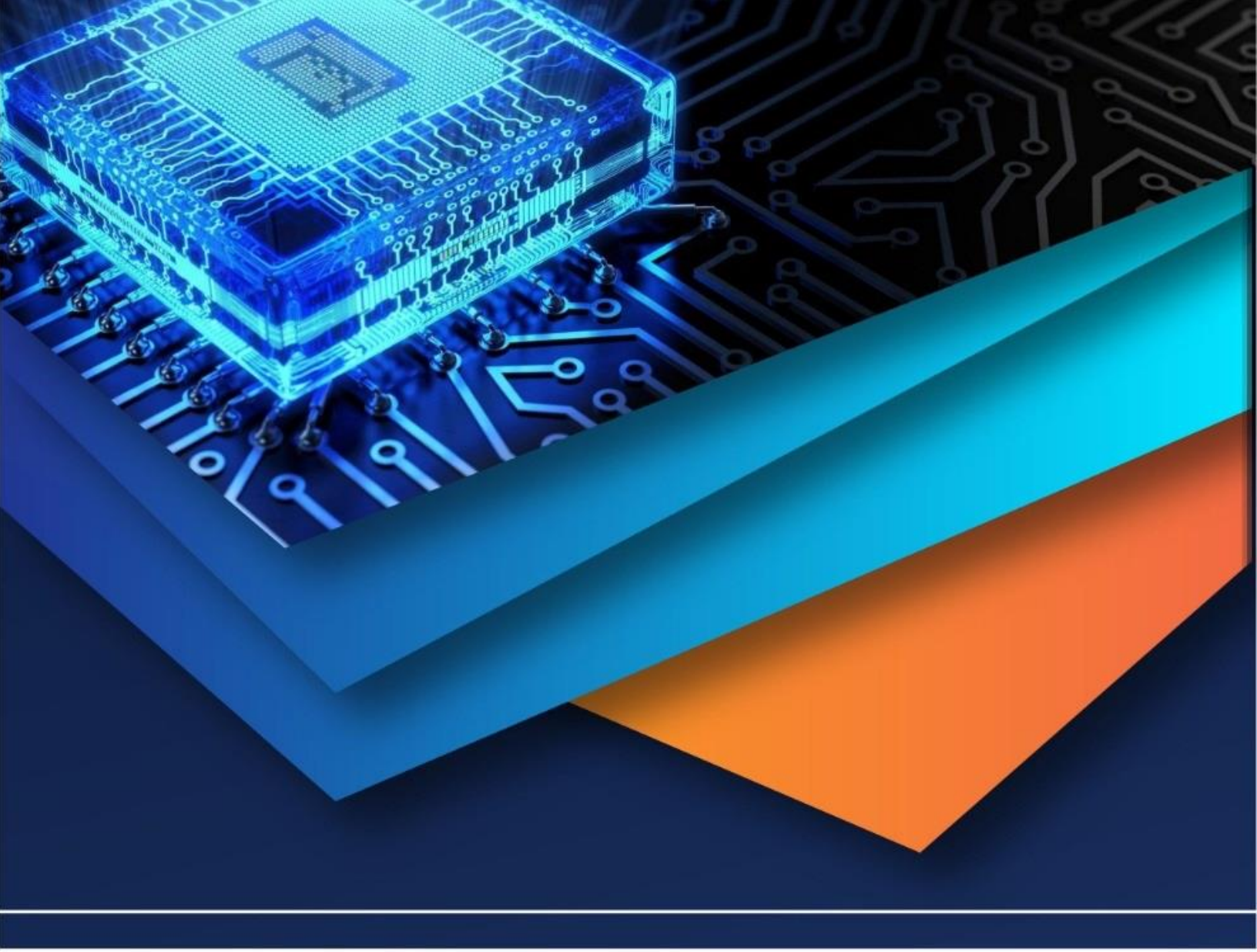
V. CONCLUSION

This study shows how machine learning methods, specifically RANDOM FOREST, can be used to forecast the risk of intrusion detec-tion. The results are promising, but future work should focus on improving model performance through more sophisticated feature engineering and handling class imbalance . intrusion decision support systems may benefit greatly from the use of machine learn-ing models like RANDOM FOREST, which could help medical practitioners identify patients more quickly and accurately.

REFERENCES

[1] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. 4th International Conference on Information Systems Security and Privacy (ICISSP), 108-116.
[2] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. Second IEEE International Conference on Secure Software Integration and Reliability Engineering (Companion), 128-131. (Highlights the need for comprehensive datasets, motivating the creation of CICIDS2017).
[3] Ho, T. K. (1995). Random Decision Forests. Proceedings of 3rd International Conference on Document Analysis and Recognition, 278-282. (Foundational work on Random Forests, a key algorithm used with CICIDS2017).
[4] Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189-1232. (Seminal paper on Gradi-ent Boosting, widely applied in CICIDS2017 research).
[5] Cortes, C., &Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273-297. (The foundational paper on Support Vector Machines, fre-quently used with CICIDS2017).
[6] Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-Based Learning Algorithms. Machine Learning, 6(1), 37-66. (Discusses K-Nearest Neighbors, often a baseline in CICIDS2017 studies).
[7] Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2(1), 38-53. (Explains key evaluation metrics used in IDS research).
[8] Sokolova, M., & Lapalme, G. (2009). A Systematic Analysis of Performance Measures for Classification Tasks. Information Processing & Management, 45(4), 427-437. (Another important paper on classification performance metrics).
[9] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444. (A seminal review of deep learning).
[10] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. (A comprehensive textbook on deep learning).
[11] Hochreiter, S., &Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. (Foundational paper on LSTM networks).
[12] Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. 2013 IEEE International Conference on Acous-tics, Speech and Signal Processing, 6645-6649. (Demonstrates RNN capabilities for sequence modeling).
[13] Hinton, G. E., &Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504-507. (Introduces autoen-coders).
[14] Chawla, N. V., Bowyer, K. W., Hall, L. O., &Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelli-gence Research, 16, 321-357. (Key paper on SMOTE for class imbalance).
[15] Drummond, C., & Holte, R. C. (2003). C4. 5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling Beats Over-Sampling. Workshop on Learning from Imbalanced Datasets II, 1-8. (Discusses class imbalance strategies).
[16] Breiman, L. (1996). Bagging Predictors. Machine Learning, 24(2), 123-140. (Introduces Bagging).
[17] Freund, Y., &Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 119-139. (Foundational paper on Boosting).
[18] Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning, 1(1), 81-106. (Discusses information gain for feature selection).
[19] Liu, H., & Yu, L. (2005). Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Transactions on Knowledge and Data Engi-neering, 17(4), 491-502. (Discusses feature selection).
[20] Ringberg, T., Soule, A., & Williamson, C. (2007). Traffic Characterization of a Campus Network. ACM SIGCOMM Computer Communication Review, 37(5), 17-30. (While not specific to CICIDS2017, it discusses traffic characterization, relevant to feature engineering).
[21] Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176. (A broader survey that includes discussion of datasets and methods relevant to CICIDS2017).
[22] Axelsson, S. (2000). Intrusion Detection Systems: A Survey and Taxonomy. Chalmers University of Technology. (A foundational survey of IDS concepts).

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



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)