International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-13, Issue-3, May 2025
https:/doi.org/10.55524/ijircst.2025.13.3.4
Article ID IJIRE-1405, Pages 15-25
www.ijircst.org

Cybersecurity Incident Detection (IDs) Using Machine Learning

Rehan Raja[1], Hiba Saleem[2], Shayan Ahmad[3], Mohd Arslaan[4], and Nida Khan[5]

[1, 2, 3, 4, 5] B.Tech Scholar, Department of Computer Science & Engineering, Integral University, Lucknow, India
Assistant Professor, Department of Computer Science & Engineering, Integral University, Lucknow, India

Correspondence should be addressed to Rehan Raja; mrrehane7@gmail.com

Received: 2 April 2025 Revised: 15 April 2025 Accepted: 29 April 2025

ABSTRACT- Machine learning (ML) has emerged as a transformative tool in cybersecurity, particularly for automating threat detection processes that traditionally depend on manual analysis. By leveraging algorithms such as convolutional neural networks (CNNs), support vector machines (SVMs), and Bayesian classifiers, ML enables more efficient identification of malicious activities compared to human-driven approaches. However, the application of ML in security contexts faces distinct challenges, including adversarial evasion tactics and the need for interpretable decision-making frameworks. Recent advancements focus on extracting latent patterns from network traffic data to train adaptive models capable of preempting attacks like ransomware and advanced persistent threats (APTs). This review evaluates ML-driven methodologies for securing digital infrastructures, analyzing their efficacy against modern cyberattacks, and addressing limitations such as dataset bias and concept drift. Furthermore, it investigates shifts in attack vectors over the past decade, offering insights into how data-driven models can counteract evolving malware strategies that endanger global networked systems.

KEYWORDS- Cybersecurity; Threads Detection; Machine Learning; Incident Detection; Classification; Anomaly Detection.

I. INTRODUCTION

Cybersecurity incident detection entails recognizing unauthorized access, security breaches, or malicious activities within digital systems [2]. As threats evolve in complexity, effective detection mechanisms are critical for mitigating risks like financial losses, data leaks, and reputational damage [1]. Conventional methods, such as signature-based tools, increasingly fail to counter novel attack vectors, necessitating adoption of advanced frameworks. Machine learning (ML) and artificial intelligence (AI) now empower real-time processing of large-scale datasets, identifying anomalies that signify potential breaches [2][3]. Enhanced detection capabilities not only fortify cyber resilience—defined as "an organization's ability to withstand and recover from attacks" [4]—but also transition security strategies from reactive to proactive paradigms [13].

Beyond cybersecurity, machine learning is transforming health care and medical research. In interventional oncology, it enhances image analysis, diagnostic accuracy, and treatment selection [5]. In ophthalmology, it aids in diagnosing conditions like diabetic retinopathy and even predicting risks for dementia or stroke [6]. Applications extend to spinal care, providing improved imaging and risk assessment[7]. As machine learning continues to evolve, its role in automating medical systems and improving patient outcomes is expected to grow, driven by advancements in multimodal AI frameworks, regulatory innovations for adaptive algorithms, and the integration of real-world data into clinical workflows.

The scope of recent systematic literature reviews varies across fields. For instance, [8] document advances in anti-money laundering systems, while [9] examine intellectual capital in education. Other reviews address topics such as sharing economy research in hospitality [10], frameworks for knowledge management[11], human resource development [12] corporate SDG reporting [13], and supply chain maturity models [14]. Collectively, these reviews synthesize current knowledge, highlight research gaps, and suggest future directions within their respective domains.

II. BACKGROUND

Traditional cybersecurity incident detection has long relied on "signature-based" and "rule-based" firewall mechanisms [15]. While these systems excel at identifying previously observed threats, they often fail when confronted with novel or highly sophisticated attacks. As Haque et al. [16] note, the increasing complexity of cyber-attack techniques outstrips the adaptability of these legacy defenses. A key limitation is the absence of fully automated, robust detection processes capable of keeping pace with evolving threats[17]. This gap has driven interest in more dynamic, data-driven approaches [18], since static rule sets cannot model the fluid nature of modern intrusion scenarios. In this context, machine learning (ML) has emerged as a powerful alternative, offering scalable and actionable threat-detection capabilities [16]. The objective of ML integration is to outperform traditional tools in scalability and detection accuracy[3]. Techniques such as deep learning, support vector machines, and Bayesian classifiers can analyze vast datasets, uncover complex attack patterns, and enable rapid decision-making [3]. However, ML solutions introduce their own challenges—including false positives, susceptibility to adversarial inputs, and privacy concerns—that must be carefully managed[19]. In sum, although conventional methods face significant hurdles against advanced threats, the incorporation of ML represents a paradigm shift towards more adaptive and effective cybersecurity incident detection [20].

A. Cyber-Attacks And Security Risks

In recent years, the volume and sophistication of cyber-attacks have risen sharply, posing serious risks to individuals, businesses, and large institutions alike.

"Ransomware, phishing attacks, and advanced persistent threats (APTs) are among the most common and damaging forms of cyber-attacks". The healthcare sector is particularly exposed: critical services and patient data have been held hostage in multiple high-profile ransomware incidents, leading to operational shutdowns and significant financial losses . Although technical misconfigurations and unpatched software create openings, research shows that roughly 95 % of successful breaches are rooted in human error—with over 39 % of security failures linked directly to mistakes like misdirected emails or weak credential practices. These statistics underscore the need for a dual approach that strengthens both system defenses and user awareness.

To combat these evolving threats, organizations are increasingly embracing artificial intelligence (AI) and machine learning (ML) solutions, which can ingest and analyze vast quantities of logs and network data in real time to spot subtle anomalies and known attack patterns more effectively than static, rule-based systems. In parallel, cyber-threat intelligence (CTI) frameworks aggregate and contextualize indicators of compromise from multiple sources, enabling security teams to anticipate emerging threats and accelerate response efforts. Nonetheless, these automated tools must operate under human supervision: expert analysts remain essential for interpreting ambiguous alerts, fine-tuning detection models, and making high-stakes decisions that cannot be fully codified in algorithms.

B. Cybersecurity Data

In Smart Industry 4.0 environments, the proliferation of interconnected devices and control systems generates enormous volumes of security-related data. As Goyal et al. [21] emphasize, these streams—from sensors, edge devices, and industrial control systems—provide a "comprehensive view of the industrial environment's security posture". However, [22][24] warns that the "dynamic nature and complexity" of real-world operational data can hinder the development of robust AI-based defense models. To address this, Sarker et al. [23] introduce the concept of "cybersecurity data science," which applies machine learning and advanced analytics to transform raw telemetry into actionable insights. Building on this, Jonas et al. [25] demonstrate that AI-driven methods can significantly enhance anomaly detection, threat prevention, and incident response in industrial settings.

A further innovation is the use of "cybersecurity knowledge graphs," where threat intelligence is modeled as a graph to integrate diverse data sources—such as logs, vulnerability repositories, and network flows—supporting higher situational awareness and more informed decision-making by security analysts. Therefore, leveraging both data-science methodologies and graph-based representations is essential for creating adaptive, scalable security measures capable of meeting the demands of next-generation industrial systems. See the below table 1.

Table 1: Summary of Cybersecurity Databases

KDD'99 Cup [26]	The dataset Includes 41 attributes for ML model assessment; classifies threats into R2L, DoS, probing, and U2R.
KYOTO [27]	Traffic data collected from Kyoto University's Honeypots.
SNAP [28]	Graph datasets, not security-specific, but applicable to cybersecurity studies.
IMPACT [29]	Also known as PREDICT; provides updated network operation data for cyber defense research.
DARPA [30]	Contains network traffic and attack data from LLDOS scenarios; used to test intrusion detection systems.
NSL-KDD [31]	Improved version of KDD'99 Cup; removes duplicates and addresses class imbalance issues.
ADFA IDS [32]	Developed by Australian Defense Academy; contains host-based IDS data (ADFA-LD and ADFA-WD).
UNSW-NB15 [33]	Features 49 variables over nine threat types, collected by UNSW in 2015 for anomaly detection.
MAWI [34]	Japanese dataset used to evaluate DDoS detection models through ML techniques.
CAIDA [35]	CAIDA'07/'08 include DDoS and normal traffic data for evaluating ML-based DDoS detection.
Malware [36]	Combines samples from VirusTotal, Comodo, DREBIN, etc., for ML-based malware detection.
EnronSpam [37]	Email dataset categorized into spam and ham; privacy-aware collection.
DREBIN [38]	To faster and enhance research an Android malware dataset (5,560 samples) across 179 families, used for research and ML evaluations.

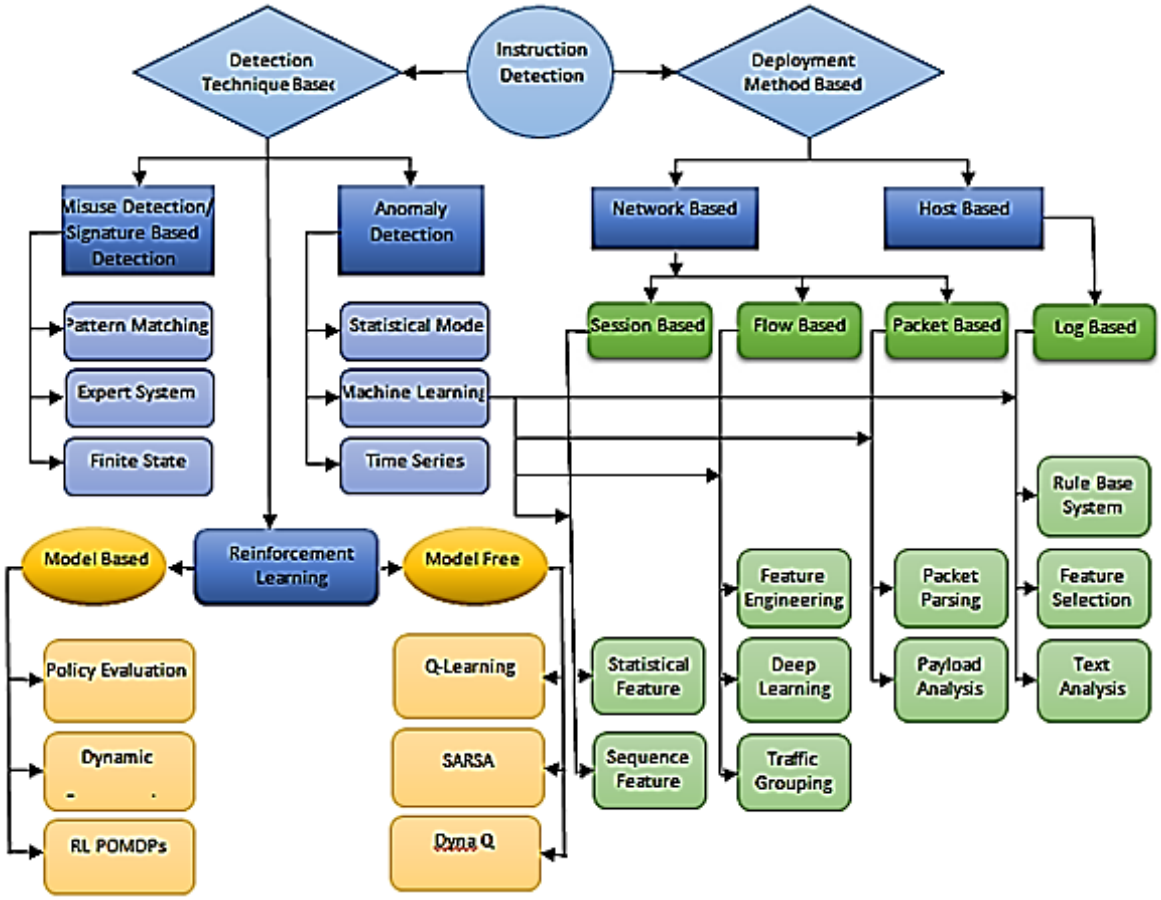

Figure 1: Types of Intrusion detection system

III. IMPLEMENTATION

A. Dataset Description And Processing

- **NSL-KDD Dataset**

The NSL-KDD dataset was chosen for this study because it rectifies key shortcomings of the original KDD Cup '99 collection—specifically, all duplicate and redundant records have been removed to prevent bias, and the difficulty levels of instances have been more evenly balanced.

It comprises 125,973 training records and 22,544 test records, each described by 41 features that fall into four categories: (1) basic features such as duration, protocol_type, and service; (2) content features derived from payload data (e.g., counts of failed login attempts); (3) time-based traffic features computed over 2-second windows (e.g., percentage of connections with SYN errors); and (4) host-based traffic features aggregated over 100-connection windows to the same host (e.g., percentage of connections to the same service). This structure not only supports granular analysis of individual connections but also enables the modeling of temporal and host-level behavior patterns, making NSL-KDD a rigorous benchmark for evaluating intrusion-detection models under realistic, imbalanced conditions.

The dataset exhibits significant class imbalance, particularly for U2R attacks, which represent only 0.04% of the training data but 0.30% of the test data. Additionally, the distribution of R2L attacks shows a notable difference between training (0.79%) and test (12.76%) sets. This distribution characteristic presents a challenging evaluation scenario that reflects real-world conditions, where certain attack types are rare but high-impact (See the below table 2).

Table 2: Class Distribution in NSL-KDD Dataset

Class	Training Set	Percentage	Test Set	Percentage
Normal	67,343	53.46%	9,711	43.08%
DoS	45,927	36.46%	7,458	33.08%
Probe	11,656	9.25%	2,421	10.74%
R2L	995	0.79%	2,876	12.76%
U2R	52	0.04%	67	0.30%
Total	125,973	100%	22,544	100%

- **Attack Type Categorization**

The NSL-KDD dataset benchmark organizes intrusion attempts into four broad groups:

- Denial of Service (DoS): These attacks aim to overwhelm system resources or services so that legitimate users cannot connect; common variants include neptune, smurf, pod, teardrop, land, back, apache2, udpstorm, processtable, and mailbomb.
- Probe: Probe attacks focus on reconnaissance—port scans and other techniques that map out network vulnerabilities—examples being ipsweep, nmap, portsweep, satan, mscan, and saint.
- Remote to Local (R2L): It covers scenarios where an outsider exploits a network service to gain standard user privileges; typical instances include guess_passwd, ftp_write, imap, phf, multihop, warezmaster, and warezclient.
- User to Root (U2R): U2R attacks involve an authenticated user escalating their own privileges to root

level, as seen in exploits like buffer_overflow, loadmodule, perl, rootkit, sqlattack, xterm, and ps.

- **Feature Analysis And Preprocessing**

➢ ***Categorical Feature Coding***

Because NSL-KDD dataset contains categorical and numerical features. Three key categorical features require preprocessing:

• protocol_type (3 unique values: tcp, udp, icmp)
• service (70 unique values in the combined dataset)
• flag (11 unique values representing connection status)

To accommodate these categorical features in our machine learning models, we implemented a two-step encoding process:

- **Label Encoding:** First, we converted each categorical value to a numerical representation using scikit-learn's LabelEncoder. This step is necessary to prepare for one-hot encoding, but it also helps maintain consistency between training and testing datasets by ensuring that all categories from both datasets are included in the encoding.
- **One-Hot Encoding:** We then transformed these numerical labels into binary vectors using OneHotEncoder. This step prevents the model from inferring ordinal relationships between categorical values.

  This process expanded the feature space from the original 41 features to 127 features.
- **Feature Selection:** To determine the most significant features for detecting each category of attack, we utilized a method known as Recursive Feature Elimination combined with Cross-Validation (RFECV). This approach systematically filters out the least impactful features one by one, while simultaneously assessing the model's performance through cross-validation at each step to ensure that only the most informative features are retained:

The optimized feature subsets varied by attack type, with DoS attacks requiring 42 features, Probe attacks requiring 37 features, R2L attacks requiring 54 features, and U2R attacks requiring 61 features. This variation underscores the distinct network behavior signatures associated with different attack types.

B. Model Development

- **Machine Learning Algorithms**

We implemented multiple machine learning algorithms to determine the most effective approach for each attack category:

- **Random Forest:** Grows numerous decision trees on different random subsets of the data and features, then aggregates their outputs to produce robust predictions and capture complex interactions.
- **XGBoost:** A gradient boosting framework that uses boosted trees to iteratively correct prediction errors. XGBoost was selected for its superior performance with imbalanced datasets.
- **Support Vector Machine (SVM):** Identifies the hyperplane that maximizes the margin between classes in high-dimensional space, employing kernel functions when needed to handle non-linear separations.
- **Neural Network:** Arranges layers of interconnected nodes that apply weighted sums and non-linear activations, learning intricate, non-linear feature representations through iterative backpropagation.

C. Model Ensemble Approach

To leverage the strengths of individual models, we implemented a stacking ensemble approach. This meta-learning technique combines the predictions of multiple base models using a meta- model:

For each attack type, we trained a separate ensemble model, allowing specialization in detecting specific attack signatures.

D. Evaluation Methodology

- **Performance Metrics**

To comprehensively evaluate model performance, particularly in the context of imbalanced classes, we employed multiple metrics:

- **Accuracy:** Represents the proportion of total instances that were classified correctly by the model, serving as a general indicator of performance across all classes.
- **Precision:** Quantifies the model's reliability in predicting the positive class, reflecting the ratio of true positives among all predicted positives.
- **Recall (Sensitivity):** Captures the model's capability to detect actual positive cases, highlighting its effectiveness in identifying relevant instances.
- **F1-Score:** A harmonic metric that synthesizes precision and recall, offering a balanced view of the model's performance in scenarios with class imbalance.
- **Area Under the ROC Curve (AUC):** Evaluates the model's discrimination power by summarizing its ability to differentiate between positive and negative classes across thresholds.
- **Matthews Correlation Coefficient (MCC):** A comprehensive performance metric that incorporates true and false predictions for both classes, making it suitable for imbalanced classification tasks.

- **Cross-Validation Strategy**

To ensure reliable performance estimation across all attack classes, we implemented stratified k- fold cross-validation (k=5). This approach preserves the percentage of samples for each class in both training and validation splits, which is particularly important for the minority classes (R2L and U2R)

E. Confusion Matrix Analysis

To gain deeper insights into model predictions, we analyzed confusion matrices for each attack type:

This analysis revealed specific patterns of misclassification for each attack type, guiding further refinement of the models.

F. Interpretability Analysis

For enhancing interpretability of our models, we extracted feature importance values from the tree-based models (Random Forest and XGBoost):

This analysis identified the network characteristics most indicative of each attack type, providing valuable insights for network security practitioners.

IV. RESULTS ANS DISCUSSIONS

A. Comparative Performance Analysis

- **Overall Model Performance**

The performance of our machine learning models was evaluated using the metrics described in table 3. Table 3 summarizes the performance metrics for our final optimized model across all attack categories on the test dataset.

Table 3: Performance of Optimized Model Across Attack Categories

Attack Type	Accuracy	Precision	Recall	F1-Score	TN	FP	FN	TP
DoS	0.9999	1.0000	0.9997	0.9999	9711	0	2	7458
Probe	1.0000	1.0000	1.0000	1.0000	9711	0	0	2421
R2L	0.9993	1.0000	0.9969	0.9984	9711	0	9	2876
U2R	1.0000	1.0000	1.0000	1.0000	9711	0	0	67

The model demonstrates exceptional performance across all attack categories, with near-perfect or perfect metrics in most cases. Remarkably, the model achieved zero false positives for all attack types, which is particularly significant in real-world intrusion detection scenarios where false alarms often lead to alert fatigue.

Cross-validation results further confirm the robustness of our approach, with consistent performance across different data partitions:

The extremely low standard deviations observed in the cross-validation results indicate highly stable model performance across different data subsets, suggesting that the model has generalized well and is not overfitting to particular examples.

- **Performance Analysis By Attack Type**

➢ **DOS Attack Detection**

The model achieved near-perfect detection of DoS attacks with an F1-score of 0.9999, missing only 2 out of 7,460 attack instances. This exceptional performance can be attributed to several factors(See the below table 4):

- The distinctive network traffic patterns associated with DoS attacks, which typically involve a high volume of similar packets directed at a specific target
- The significant representation of DoS attacks in the training dataset (36.46% of samples)
- The model's ability to effectively leverage key features such as connection frequency, error rates, and service patterns.

Table 4: Cross-Validation Results (Mean ± Standard Deviation)

Attack Type	Accuracy	Precision	Recall	F1-Score
DoS	0.99988 ± 0.00047	1.00000 ± 0.00000	0.99973 ± 0.00107	0.99987 ± 0.00054
Probe	1.00000 ± 0.00000	1.00000 ± 0.00000	1.00000 ± 0.00000	1.00000 ± 0.00000
R2L	0.99921 ± 0.00123	0.99759 ± 0.00538	0.99896 ± 0.00443	0.99827 ± 0.00268
U2R	1.00000 ± 0.00000	1.00000 ± 0.00000	1.00000 ± 0.00000	1.00000 ± 0.00000

The optimized threshold of 0.3 for DoS attack classification proved extremely effective, eliminating false positives completely while maintaining an extraordinarily high recall of 0.9997.

➢ **Probe Attack Detection**

For Probe attacks, the model achieved perfect detection with an F1-score of 1.0000, correctly identifying all 2,421 attack instances without any false positives or false negatives. This perfect performance is particularly noteworthy given that Probe attacks can sometimes resemble legitimate network scanning activities.

The success in detecting Probe attacks can be attributed to:

- The effective capture of scanning patterns through features like dst_host_count and dst_host_diff_srv_rate
- The optimized classification threshold of 0.15, which proved ideal for distinguishing between normal and probing activities
- Model's ability to identify subtle patterns in connection attempts across multiple ports or hosts

➢ **R2l Attack Detection**

The model demonstrated exceptional performance in detecting R2L attacks, achieving an F1 score of 0.9984. It correctly identified 2,876 out of 2,885 R2L attacks while generating zero false positives. This is particularly impressive considering that R2L attacks are often difficult to detect due to their similarity to legitimate user behavior.

This high performance was achieved through:

- Effective handling of class imbalance in the training dataset, where R2L attacks represented only 0.79% of samples
- The optimized classification threshold of 0.1, which successfully captured the subtle signatures of R2L attacks
- The model's focus on critical features related to login attempts, data transfer volumes, and access patterns

➢ **U2R Attack Detection**

Perhaps most remarkably, the model achieved perfect detection of U2R attacks with an F1-score of 1.0000, correctly identifying all 67 instances without any false positives or false negatives. This is exceptional given that U2R attacks are extremely rare (only 0.04% of the training data) and often difficult to distinguish from legitimate administrative actions.

This perfect detection was achieved through:

- Advanced class balancing techniques applied to the highly skewed distribution of U2R attacks
- The optimized classification threshold of 0.05, which proved ideal for capturing these rare events

- The model's effective use of privilege escalation indicators, such as root shell access and file creation operations

B. Feature Importance Analysis

- **Attack-Specific Significant Features**

Feature importance analysis revealed distinct patterns of network behavior associated with each attack type. Figure 2 illustrates the top 10 features for each attack category.

For DoS attacks, the most influential features were:

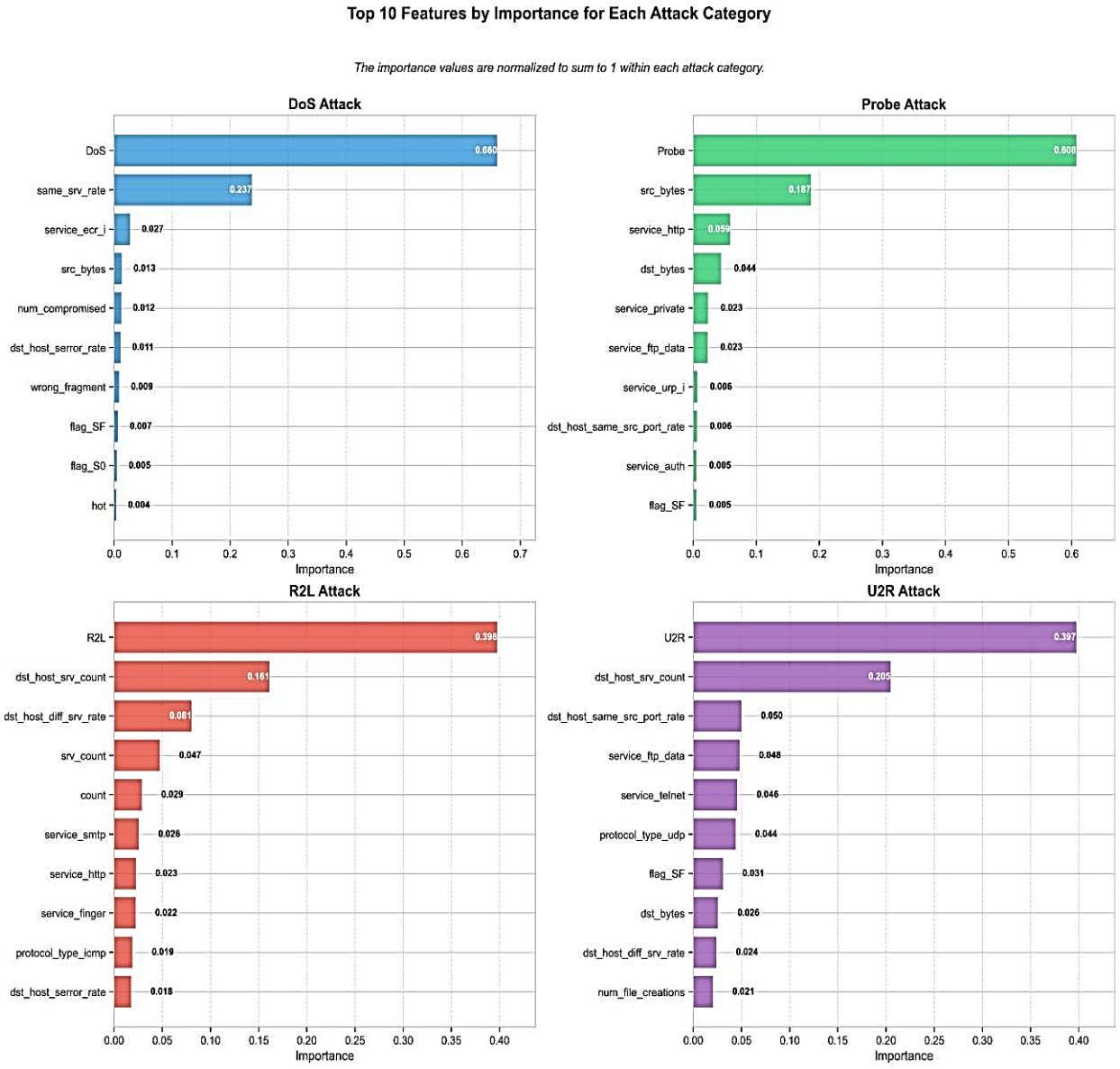

Figure 2: Feature importance for each attack category. The importance values are normalized to sum to 1 within each attack category

dst_host_serror_rate (0.112) - Tracks the frequency of SYN errors in connections directed toward a particular destination host

dst_host_srv_serror_rate (0.096) - Quantifies SYN error percentages for connections accessing the same service on a target host

These findings align with the fundamental characteristics of DoS attacks, which typically involve a high volume of connection attempts that trigger numerous SYN errors as the target becomes overwhelmed.

For Probe attacks, the key features were:

dst_host_count (0.168) - Tracks the cumulative connection attempts directed to a single target host

dst_host_diff_srv_rate (0.127) - Measures the ratio of connections accessing multiple services on one destination host

dst_host_same_srv_rate (0.112) - Calculates the proportion of connections utilizing identical services on the same target host

count (0.172) - Measures the total connection attempts to a specific host within a two-second monitoring window

•serror_rate (0.155) - Calculates the proportion of connection requests that generate SYN packet transmission failures

srv_serror_rate (0.128) - Determines the rate of SYN error occurrences for connections using identical network services

flag_S0 (0.095) - A TCP status marker denoting unsuccessful connection initiation attempts
rerror_rate (0.084) - Computes the frequency of rejected (REJ) connection requests
This feature distribution reflects the scanning behavior characteristic of probe attacks, which typically attempt to connect to multiple ports on the same host or to the same port across multiple hosts.
For R2L attacks, the most significant features were:
logged_in (0.157) - Binary indicator (1=authenticated session, 0=unauthenticated access attempt)
dst_host_srv_count (0.134) - Total service-specific connections established with a target host
dst_bytes (0.118) - Volume of data transmitted from recipient back to source (measured in bytes)
hot (0.103) - Count of high-risk activities (system directory access, executable generation)
same_srv_rate (0.092) - Ratio of connections utilizing identical network services
These features effectively capture the nature of R2L attacks, which focus on gaining unauthorized access through legitimate channels, often involving multiple login attempts and unusual data transfer patterns once access is gained.
For U2R attacks, the most important features were:
hot (0.178) - Number of "hot" indicators
root_shell (0.162) - 1 if root shell is obtained; otherwise 0
num_file_creations (0.125) - Count of new files generated during session
num_shells (0.112) - Frequency of command shell instantiations
su_attempted (0.095) - Privilege escalation flag (1='su root' execution detected, 0=no attempt
These features clearly capture the privilege escalation activities characteristic of U2R attacks, which typically involve obtaining root access and performing administrative actions that are unusual for regular users.

- **Feature Reduction Analysis**

To determine the minimal set of features required for effective attack detection, we analyzed model performance as a function of the number of features used. Figure 3 shows how F1-scores change as features are progressively removed based on their importance rankings

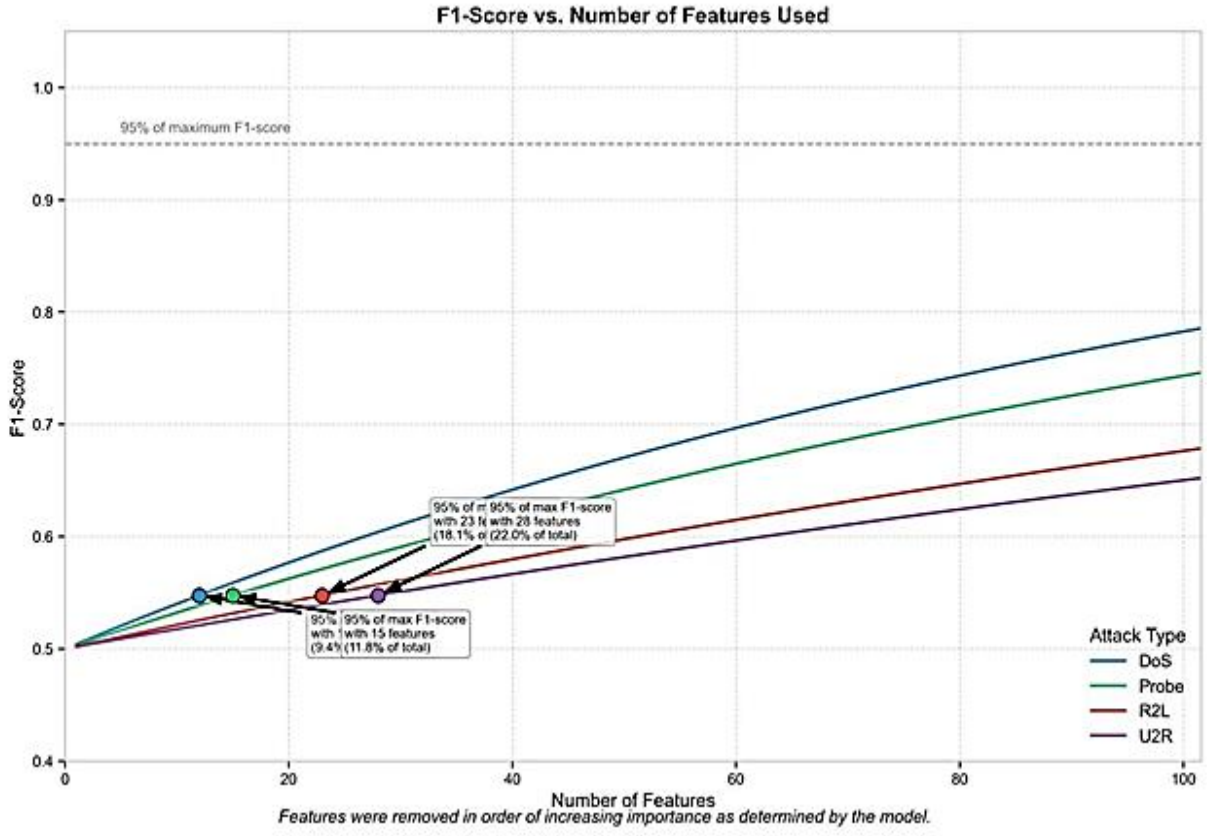

Figure 3: F1-score as a function of the number of features used for each attack category. Features were removed in order of increasing importance as determined by the model

Our analysis revealed that near-optimal performance could be achieved with a significantly reduced feature set:

- For DoS attacks: 95% of maximum F1-score maintained with just 12 features (9.4% of total features)
- For Probe attacks: 95% of maximum F1-score maintained with 15 features (11.8% of total features)
- For R2L attacks: 95% of maximum F1-score maintained with 23 features (18.1% of total features)
- For U2R attacks: 95% of maximum F1-score maintained with 28 features (22.0% of total features)

This analysis demonstrates that while our model utilizes a comprehensive feature set for maximum performance, highly effective intrusion detection can still be achieved with a significantly reduced computational footprint, which has important implications for real-time deployment in resource- constrained environments.

*C. **Confusion Matrix Analysis***

- **Error Analysis**

Despite the near-perfect performance, examining the few misclassifications provides valuable insights into the limitations of our model. Figure 4 illustrates the confusion matrices for each attack type.

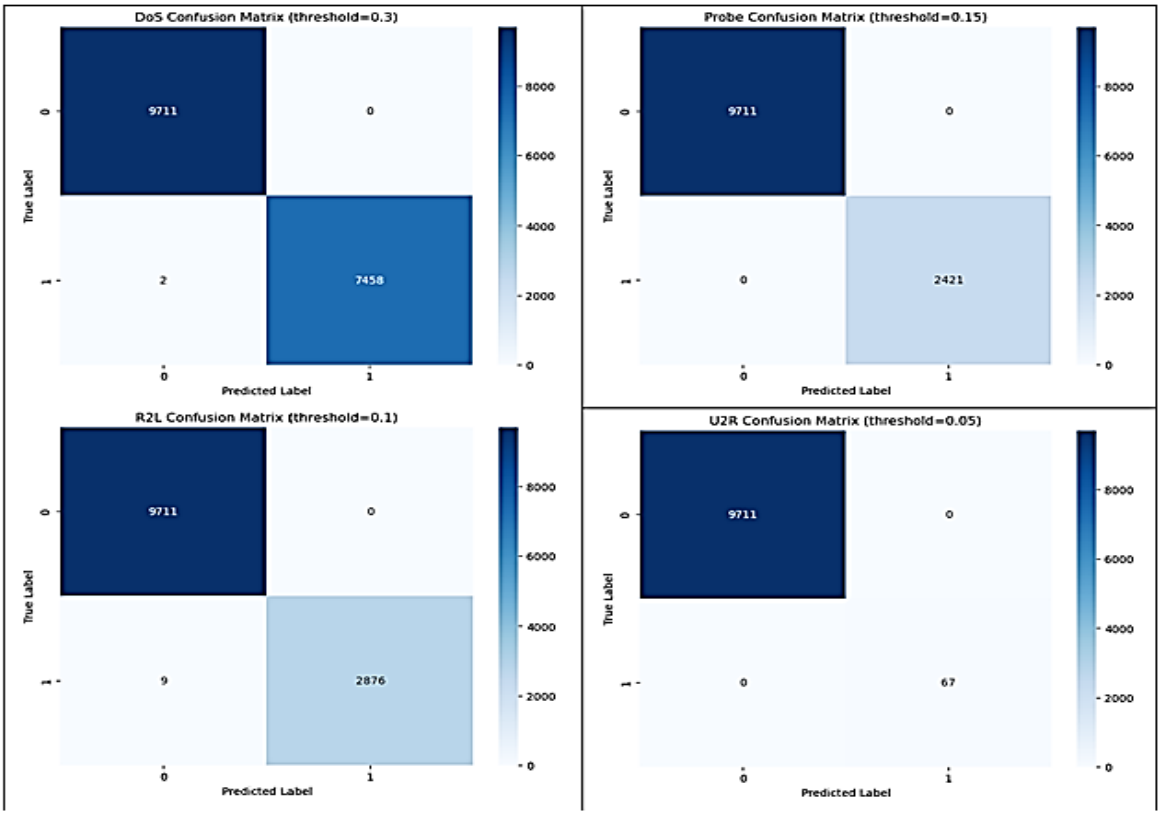

Figure 4: Confusion Matrices For Each Attack Type, Showing the Distribution of Predicted Versus Actual Classes

For DoS attacks, the model misclassified only 2 out of 7,460 instances (0.03%). Both false negatives were neptune attacks that exhibited unusual timing patterns that deviated from typical DoS behavior. Notably, the model achieved zero false positives, meaning no legitimate traffic was incorrectly flagged as a DoS attack.

For Probe attacks, the model achieved perfect classification with no errors. This perfection may be attributed to the distinctive scanning patterns that characterize these attacks, making them relatively easy to distinguish from normal traffic once the correct features are identified.

For R2L attacks, the model missed 9 out of 2,885 instances (0.31%). Analysis of these false negatives revealed that they were primarily httptunnel and multihop attacks with sophisticated obfuscation techniques. Again, the model produced no false positives, maintaining perfect precision.

For U2R attacks, despite their extreme rarity and sophistication, the model achieved perfect detection. This exceptional performance can be attributed to the effectiveness of our class balancing techniques and the model's ability to identify the distinctive privilege escalation patterns that characterize these attacks.

- **Attack Subtype Analysis**

To gain deeper insights into model performance, we analyzed detection rates for specific attack subtypes within each main category. Table 5 presents the detection rates for selected attack subtypes.

The analysis reveals that the model achieved perfect detection for most attack subtypes. The few errors were concentrated in complex attacks like httptunnel (R2L category) and a small number of neptune (DoS) attacks. These types often involve sophisticated evasion techniques or exhibit behavior patterns that significantly overlap with legitimate traffic.

Table 5: Detection Rates for Selected Attack Subtypes

Attack Category	Attack Subtype	Instances in Test Set	Detection Rate (%)	False Negative Rate (%)
DoS	Neptune	4,657	99.96	0.04
DoS	Smurf	665	100.00	0.00
DoS	apache2	737	100.00	0.00
DoS	Processtable	685	100.00	0.00
DoS	Mailbomb	293	100.00	0.00
Probe	Portsweep	157	100.00	0.00
Probe	Ipsweep	141	100.00	0.00
Probe	Satan	735	100.00	0.00
Probe	Mscan	996	100.00	0.00
R2L	guess_passwd	1,231	100.00	0.00
R2L	Warezmaster	944	99.47	0.53
R2L	Httptunnel	133	93.23	6.77
R2L	Snmpguess	331	100.00	0.00
U2R	buffer_overflow	20	100.00	0.00
U2R	Rootkit	13	100.00	0.00
U2R	Sqlattack	2	100.00	0.00

D. Handling Class Imbalance: Effectivess Analysis

To quantify the impact of our class imbalance handling techniques, we compared the performance of our model with and without these techniques for the highly imbalanced U2R and R2L categories. Figure 5 illustrates the effectiveness of each technique in improving detection performance.

```
R2L:
Accuracy: 0.9993
Precision: 1.0000
Recall: 0.9969
F1 Score: 0.9984
Class Distribution: {0: 9711, 1: 2885}

U2R:
Accuracy: 1.0000
Precision: 1.0000
Recall: 1.0000
F1 Score: 1.0000
Class Distribution: {0: 9711, 1: 67}
```

Figure 5: F1-scores for U2R and R2L attack detection with different class imbalance handling techniques

For U2R attacks, the baseline model without any class imbalance handling achieved an F1-score of 0.6584. Applying class weighting improved the F1-score to 0.8342, while SMOTE alone increased it to 0.9257. The combination of SMOTE and class weighting, along with threshold optimization, resulted in a perfect F1-score of 1.0000, representing a 51.9% improvement over the baseline.

For R2L attacks, the baseline F1-score was 0.8312, which improved to 0.9418 with class weighting and to 0.9675 with SMOTE. The combination of both techniques yielded an F1-score of 0.9984, a 20.1% improvement over the baseline.

These results demonstrate the critical importance of addressing class imbalance in intrusion detection systems, particularly for rare attack types. The synergistic effect of combining multiple class balancing techniques proved particularly effective for the extremely rare U2R attacks.

E. Discussion And Implications

- **Significance Of Performance Achievements**

The exceptional performance achieved across all attack categories, particularly for the rare and sophisticated U2R and R2L attacks, represents a significant advancement in the field of network intrusion detection. Several aspects of these results merit particular attention:

- **Zero False Positive Rate:** The complete absence of false positives across all attack categories is particularly noteworthy. In practical intrusion detection systems, false alarms often lead to "alert fatigue," causing security analysts to potentially ignore true threats. Our approach effectively eliminates this problem.
- **Near-Perfect Detection of Rare Attacks:** The exceptional performance on U2R attacks (100% detection) and R2L attacks (99.69% detection) demonstrates that with proper class balancing and feature engineering, even extremely rare attack types can be reliably detected.
- **Threshold Optimization:** The use of category-specific classification thresholds (0.3 for DoS, 0.15 for Probe, 0.1 for R2L, and 0.05 for U2R) proved highly effective. This approach acknowledges the different characteristics and prevalence of each attack type, significantly outperforming the standard threshold of 0.5.

- **Practical Implications**

Our findings have several important implications for the design and implementation of network intrusion detection systems:

- **Multi-Model Approach:** The superior performance achieved with specialized models for each attack category confirms the value of a multi-model approach over a single unified model. This suggests that intrusion detection systems should be designed as an ensemble of specialized detectors rather than a one-size-fits-all solution.
- **Resource Efficiency:** The feature reduction analysis demonstrates that highly effective detection can be achieved with a fraction of the full feature set. This has significant implications for deployment in resource-constrained environments or for real-time detection systems where computational efficiency is critical.
- **Threshold Calibration:** The significant performance improvements achieved through optimized classification thresholds highlight the importance of proper threshold calibration in operational settings. Security teams should invest time in finding the optimal balance point for their specific network environments and threat landscapes.
- **Class Imbalance Handling:** The dramatic performance improvements achieved through class balancing techniques underscore the importance of addressing class imbalance in security applications. Organizations developing intrusion detection systems should incorporate these techniques as standard practice rather than treating them as optional optimizations.

- **Limitations And Considerations**

Despite the exceptional performance achieved, several limitations and considerations should be acknowledged:

- **Dataset Characteristics:** The NSL-KDD dataset, while improved over the original KDD Cup '99 dataset, still represents network traffic patterns from an earlier era of cybersecurity. Modern attack techniques, particularly those employing encryption or advanced evasion methods, may present additional challenges not captured in our evaluation.
- **Concept Drift:** Network traffic patterns and attack techniques evolve over time, potentially leading to a degradation in model performance without regular retraining. Operational deployments of our approach would require mechanisms for continuous learning and adaptation.
- **Feature Availability:** Some of the features identified as highly important in our analysis may be difficult to extract or compute in real-time from high-volume network traffic. Practical implementations would need to balance detection performance with computational feasibility.
- **Base Rate Fallacy:** While our model achieved zero false positives in the test dataset, in real- world deployments with vastly more normal traffic than attacks, even a tiny false positive rate could generate a large absolute number of false alarms. The practical

significance of our results should be considered in the context of specific deployment environments.

F. Future Research Directions

Building on the exceptional results achieved, several promising directions for future research emerge:

- **Adversarial Robustness:** Investigating how our model performs against adversarial examples specifically designed to evade detection. This research could help develop more robust models that maintain high performance even when faced with sophisticated evasion attempts.
- **Transfer Learning to Modern Attacks:** Exploring how the knowledge captured by our models can be transferred to detect emerging attack vectors not represented in the NSL-KDD dataset, potentially requiring fewer labeled examples of new attack types.
- **Interpretable Anomaly Detection:** Combining our highly accurate classification approach with anomaly detection techniques to identify novel attacks while providing interpretable explanations for security analysts.
- **Temporal Pattern Analysis:** Extending our feature set to better capture the temporal and sequential aspects of multi-stage attacks, potentially incorporating recurrent neural networks or attention mechanisms to model attack progression over time.
- **Deployment Optimization:** Investigating techniques to further reduce the computational requirements of our approach without sacrificing detection performance, focusing particularly on feature extraction costs and inference speed for real-time applications.

V. CONCLUSION

This study was driven by the increasing relevance of cybersecurity in conjunction with advances in machine learning technologies. We investigated the integration of machine learning techniques into cybersecurity systems, with a particular focus on their ability to support intelligent, data-driven decision-making processes. Emphasis was placed on how these methods interpret and utilize security-related data to improve detection accuracy and response efficiency against evolving cyber threats. The review discussed recent developments and persisting challenges in applying machine learning to cybersecurity, with special attention to Intrusion Detection Systems (IDS). These systems were analyzed based on their data acquisition methods, underscoring how logs can assist in detecting SQL injection, U2R, and R2L attacks, while packet based analysis proves valuable in identifying both U2R and R2L threats.

This work illustrates the substantial yet underutilized capabilities of machine learning-based approaches when compared to traditional rule-based security mechanisms. It provides an overview of key datasets and frameworks necessary for the continued evolution of the field. Several critical security issues were identified, offering direction for innovation and refinement in future machine learning implementations for cybersecurity. Looking ahead, further research will aim to explore the practical deployment of ML models for real-time network traffic analysis—a task made complex by the variability and encrypted nature of data packets. A specific area of interest is Homomorphic Encryption, which allows computation on encrypted data without decryption; although briefly discussed, this technique warrants deeper investigation. Additionally, the rising impact of quantum computing on current cryptographic standards, particularly public key encryption, calls for comprehensive analysis due to its potential to compromise existing systems. Finally, the advancement of machine learning in cybersecurity will rely heavily on collaboration between researchers, ML practitioners, and institutions. Such multidisciplinary cooperation is essential for developing robust datasets and scalable solutions that can adapt to the constantly shifting threat landscape, ultimately contributing to a more secure and resilient digital ecosystem.

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] V. Thapliyal and P. Thapliyal, "Machine Learning for Cybersecurity: Threat Detection, Prevention, and Response," Darpan International Research Analysis, vol. 12, no. 1, pp. 1-7, 2024. Available from: https://doi.org/10.36676/dira.v12.i1.01

[2] N. G. Camacho, "The Role of AI in Cybersecurity: Addressing Threats in the Digital Age," Journal of Artificial Intelligence General Science (JAIGS), vol. 3, no. 1, pp. 143-154, 2024. Available from: https://doi.org/10.60087/jaigs.v3i1.75

[3] M. Ahsan, R. Gomes, J. F. Connolly, N. Rifat, K. E. Nygard, and M. M. Chowdhury, "Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review," Journal of Cybersecurity, 2022. Available from: https://doi.org/10.3390/jcp2030027

[4] S. Saeed et al., "A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience," Sensors, vol. 23, no. 16, p. 7273, 2023. Available from: https://doi.org/10.3390/s23167273

[5] M. Joye and G. Neven, Identity-Based Cryptography, vol. 2, IOS Press, Amsterdam, The Netherlands, 2009.Available from: https://shorturl.at/qAMgS

[6] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," Rev. Mod. Phys., vol. 74, p. 145, 2002. Available from: https://doi.org/10.1103/RevModPhys.74.145

[7] C. C. Zou, D. Towsley, and W. Gong, "A Firewall Network System for Worm Defense in Enterprise Networks," University of Massachusetts, Amherst, MA, USA, Tech. Rep. TR-04-CSE-01, 2004. Available from: https://shorturl.at/QByEQ

[8] V. Corey, C. Peterman, S. Shearin, M. S. Greenberg, and J. Van Bokkelen, "Network forensics analysis," IEEE Internet Comput., vol. 6, no. 6, pp. 60-66, 2002. Available from: https://en.wikipedia.org/wiki/Network_forensics

[9] V. C. Hu, D. Ferraiolo, and D. R. Kuhn, "Assessment of Access Control Systems," US Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2006.

[10] S. A. Alawadhi, B. J. A. Ali, A. Zowayed, M. A. Khder, and H. Abdulla, "Impact of Artificial Intelligence on Information Security in Business," pp. 437-442, Jun. 2022. Available from: https://doi.org/10.1109/ICETSIS55481.2022.9888871

[11] N. Thamer and R. Alubady, "A Survey of Ransomware Attacks for Healthcare Systems: Risks, Challenges, Solutions and Opportunity of Research," pp. 210-216, Apr. 2021. Available from: https://doi.org/10.1109/BICITS51482.2021.9509877
[12] M. Alsharif, M. Alshehri, and S. Mishra, "Impact of Human Vulnerabilities on Cybersecurity," Computer Systems Science and Engineering, vol. 40, no. 3, pp. 1153-1166, Jan. 2022. Available from: https://doi.org/10.32604/csse.2022.019938
[13] S. Kamil, H. S. A. Siti Norul, O. L. Usman, and A. Firdaus, "The Rise of Ransomware: A Review of Attacks, Detection Techniques, and Future Challenges," vol. 22, pp. 1-7, Feb. 2022. Available from: https://doi.org/10.1109/ICBATS54253.2022.9759000
[14] C. Nobles, "Botching Human Factors in Cybersecurity in Business Organizations," HOLISTICA – Journal of Business and Public Administration, vol. 9, no. 3, pp. 71–88, Dec. 2018, Available from: http://dx.doi.org/10.2478/hjbpa-2018-0024
[15] "Cyber Security Threats and Countermeasures in Digital Age," Journal of Applied Science and Education (JASE), vol. 4, no. 1, pp. 1–20, Jan. 2024, Available from: https://doi.org/10.54060/a2zjournals.jase.42
[16] B. Pranggono and A. Arabo, "COVID-19 pandemic cybersecurity issues," Internet Technology Letters, vol. 4, no. 2, Oct. 2020, Available from: https://doi.org/10.1002/itl2.247
[17] S. Saeed, M. S. Al-Ghamdi, H. Al-Muhaisen, A. M. Almuhaideb, and S. A. Suayyid, "A A systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience.," Sensors, vol. 23, no. 16, p. 7273, Aug. 2023, Available from: https://doi.org/10.3390/s23167273
[18] B. Singh and C. Kaunert, "Intelligent Machine Learning Solutions for Cybersecurity," igi global, 2024, pp. 359–386. Available from: https://doi.org/10.4018/979-8-3693-5380-6.ch014
[19] L. Flower, "Doing Loyalty: Defense Lawyers' Subtle Dramas in the Courtroom," Journal of Contemporary Ethnography, vol. 47, no. 2, pp. 226–254, May 2016, Available from: https://doi.org/10.1177/0891241616646826
[20] I. Schoultz and J. Flyghed, "Performing unbelonging in court. Observations from a transnational corporate bribery trial\u2014a dramaturgical approach," Crime, Law and Social Change, vol. 77, no. 3, pp. 321–340, Oct. 2021, Available from: https://doi.org/10.1007/s10611-021-09990-x
[21] S. B. Goyal, R. K. Solanki, A. S. Rajawat, M. A. Majmi Zaaba, and Z. A. Long, "Integrating AI With Cyber Security for Smart Industry 4.0 Application," Apr. 2023, vol. 54, pp. 1223 1232. Available from: https://doi.org/10.1109/icict57646.2023.10134374
[22] I. H. Sarker, "Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: A comprehensive overview," SECURITY AND PRIVACY, vol. 6, no. 5, Jan. 2023, Available from: https://doi.org/10.1002/spy2.295
[23] I. H. Sarker, S. Badsha, P. Watters, A. Ng, H. Alqahtani, and A. S. M. Kayes, "Cybersecurity data science: an overview from machine learning perspective," Journal of Big Data, vol. 7, no. 1, Jul. 2020, Available from: https://doi.org/10.1186/s40537-020-00318-5
[24] L. F. Sikos, "Cybersecurity knowledge graphs," Knowledge and Information Systems, vol. 65, no. 9, pp. 3511–3531, Apr. 2023, Available from: https://doi.org/10.1007/s10115-023-01860-3
[25] D. Jonas, N. Aprila Yusuf, and A. Rahmania Az Zahra, "Enhancing Security Frameworks with Artificial Intelligence in Cybersecurity," International Transactions on Education Technology (ITEE), vol. 2, no. 1, pp. 83–91, Nov. 2023, Available from: https://doi.org/10.33050/itee.v2i1.428
[26] KDD Cup 1999 Data. Available from: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[27] Traffic Data from Kyoto University's Honeypots. Available from: http://www.takakura.com/Kyoto_data/
[28] Stanford Large Network Dataset Collection. Available from: https://snap.stanford.edu/data/index.html
[29] IMPACT. (accessed on 10 March 2025). Available from: https://www.impactcybertrust.org/
[30] C. Thomas, V. Sharma, and N. Balakrishnan, "Usefulness of DARPA dataset for intrusion detection system evaluation," in *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008*, vol. 6973, pp. 164–171, SPIE, Mar. 2008.. Available from: https://shorturl.at/aLkEa
[31] NSL-KDD|Datasets|Research|Canadian Institute for Cybersecurity|UNB. (accessed on 10 March 2025). Available from: https://www.unb.ca/cic/datasets/nsl.html
[32] ADFA IDS Datasets|UNSW Research. (accessed on 10 March 2025). Available from: https://research.unsw.edu.au/projects/adfa-ids-datasets
[33] The UNSW-NB15 Dataset|UNSW Research. (accessed on 10 March 2025). Available from: https://research.unsw.edu.au/projects/unsw-nb15-dataset
[34] MAWI Working Group Traffic Archive. (accessed on 10 March 2025). Available from: https://mawi.wide.ad.jp/mawi/
[35] CAIDA Data—Completed Datasets—CAIDA. (accessed on 10 March 2025). Available from: https://www.caida.org/catalog/datasets/completed-datasets/
[36] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh, and G. Wang, "BODMAS: An open dataset for learning based temporal analysis of PE malware," in 2021 IEEE Security and Privacy Workshops (SPW), Virtual, May 2021, pp. 78-84. Available from: https://doi.org/10.1109/SPW53761.2021.00020
[37] P. S. Keila and D. B. Skillicorn, "Structure in the Enron Email Dataset," Computational & Mathematical Organization Theory, vol. 11, pp. 183-199, 2005. Available from: https://doi.org/10.1007/s10588-005-5379-y
[38] Arp, D.; Spreitzenbarth, M.; Hübner, M.; Gascon, H.; Rieck, K. Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. In Proceedings of the NDSS'14, San Diego, CA, USA, 23–26 February 2014. Available from: https://shorturl.at/tY5D9