International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I11P132

Volume 74 Issue 11, 500-520, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Integrating Machine Learning for Improved Threat Detection Accuracy in Extended Detection and Response (XDR) Systems

Erik Hadianto[1], Astari Retnowardhani[2]

[1,2]*Information Systems Management Department, BINUS Graduate Program - Master of Information Systems Management, Bina Nusantara University, Jakarta 11480, Indonesia.*

[1]*Corresponding Author: erik.hadianto@binus.ac.id*

Received: 15 May 2026 Revised: 21 September 2026 Accepted: 26 September 2026 Published: 25 November 2026

Abstract *- As digital transformation continues to accelerate, cybersecurity risks have also increased due to the widespread deployment of digital solutions and applications on the internet. Although the main goal of these technologies is to make access and communication easier, they also add a number of possible vulnerabilities that raise the possibility of data loss or manipulation. Therefore, specialized IT security teams are necessary to continuously monitor, analyze, and proactively detect vulnerabilities early, minimizing the risk of attacks by malicious actors. Endpoint-level monitoring utilizing Endpoint Detection and Response (EDR) solutions, network traffic analysis using Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), and other methods can be used to identify such vulnerabilities. The time and complexity required for a thorough threat analysis, however, are frequently exacerbated by the use of multiple security technologies. Extended Detection and Response (XDR) platforms, which collaborate data and capabilities from several security solutions into a centralize framework, have evolved as a solution to this problem. Furthermore, the accuracy with which threats are recognized and examined has significantly increased thanks to recent advancements in machine learning. Specific to, random forest models shown detection accuracies above 99%, indicating that threats are successfully identified. This integration implies that, with machine learning approaches, XDR technology can significantly raise an organization's cybersecurity profile.*

Keywords *- Information Technology, Cyber-Attack, Cyber security, Cross-Industry Standard Process for Data Mining (CRISP-DM), Extended Detection and Response (XDR), Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS), Endpoint Detection and Response (EDR).*

1. Introduction

Nowadays, the Internet has become a vast and virtually limitless resource that is deeply integrated into modern daily life. More than 73.2% of the world's population has adopted the Internet; 9.6% of the global population is in Southeast Asia; and in Indonesia itself, around 230.4 million people are internet users [1]. It functions as a medium for communication and information exchange for people worldwide. According to Forbes, the internet generates around 2.5 million terabytes of data every day [2].

In both business and everyday life, Information and Communication Technology (ICT) is important, and it is managed through Digital Transformation across multiple dimensions in the enterprise. In fact, the Compound Annual Growth Rate (CAGR) in the Indonesia report 2021-2026 shows a consistent trend of a 2-digit increase in IT investment. Cyberattacks on ICT systems have grown more complex and sophisticated due to emerging of digital transformation, this make cyber security is a major concern, Based on report from several cybersecurity provider and consultant, cost of cyberattack reach 10,500 bio USD and 9.1% of Global Gross Domestic Product (GDP) [3] and in and in Indonesia, 3300 cyberattack per week, over 660 million records and 1 terabyte of data breach and possible losses IDR 14,2 trillion during range 2020-2022 [4, 5]. International Data Corporation (IDC) research indicates that 44% of Indonesian enterprises are concerned about cybersecurity threats, and 88% plan to increase spending on security services range is 2012-2023. According to Deloitte Audit, cybersecurity remains in the top 5, followed by digital transformation [6].

The rising number of cybercrimes and cyberattacks highlights the need for better security procedures. Due to the increasing usage of digital technology globally, the development of secure and dependable computers, internet infrastructure, and other technology that can resist such assaults is critical. The cyber threat environment is constantly

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

evolving, and cybersecurity technologies are continually being developed to address it and make more difficult for IT security teams to keep up with this complexity.

On average, an organization hires a dozen or more employees for a security team to manage a collection of security software from several vendors, which together produce thousands of alerts daily. Surprisingly, the majority of security executives rank the complexity of their environment as one of their biggest security challenges. The threat environment is still evolving and expanding rapidly. As attack vectors multiply, it is critical to begin. Many businesses use the best-in-class solution for each vector, with the endpoints connecting to a cloud network to protect those specific vulnerabilities [7].

Consider one of the cyber events that occurred in one of Indonesia's banking institutions in 2023, namely Bank Syariah Indonesia (BSI). The cyber incident at BSI began with brute-force attacks on security devices whose default settings were still enabled, granting access to systems and enabling lateral movement. The user was compromised, some central controller was affected, and the security tools at BSI were successfully hacked. Through investigation and adequate security measures, this attack could have been detected early and prevented from causing complete system damage [8]. The sheer number of security alerts causes 70% of Security Analyst teams to experience emotional overload, while 55% of security teams report feeling this way, according to Anton's research [9]. A poll found that, on average, Security Operation Center (SOC) teams manage 3,832 alerts per day, but 62% are ignored, resulting in critical alerts being missed. According to 78% of cybersecurity professionals, large businesses (with over 10,000 employees) receive 1,000 or more alerts daily, and 56% of them are missed. They claim that it averages out at that. It takes more than ten minutes to check each alarm. This happens because security products are not integrated, and architectures are fragmented. Connect the dots throughout the entire stack. Teams have multiple management websites to review, and they have to link the data from each manually.

As a result, security data is collected and analyzed in isolation, which opens up vulnerabilities outside any security team's line of sight by not requiring any viewpoint or connection, as well as by identifying. They respond to notifications more slowly, have fewer opportunities for investigation, and are more likely to neglect to account for an ongoing cyberattack [8]. Integrated security tools are required to automatically support detection, response, and protection through Extended Detection and Response (XDR). Extended Detection and Response (XDR) platforms, which unify data from endpoints, networks, and cloud services, are shown in Figure 1 [7], Figure 2 [10]. The dataset used in this study represents consolidated XDR telemetry rather than a single-source security dataset. The telemetry contains security events and characteristics originating from multiple security sources within the EDR and XDR environment. These data are consolidated at the XDR layer before being processed by the proposed machine learning pipeline.

Consequently, the proposed approach focuses on classifying threats based on integrated security telemetry rather than relying on integrated security telemetry rather than relying on isolated characteristics from a single security device or log source. The cybersecurity system, as shown in Figure 3, comprises three phases. The initial phase records every traffic log on the network, firewalls, endpoints, servers, and other equipment. The second step is that all logs will be saved in the XDR data lake, which will subsequently be used to identify and analyze cyberattacks. The last step is to react to cyberattacks immediately and in accordance with approved protocols, such as instructing IPS, firewalls, and EDR to prevent them. This security system is designed to use XDR to monitor, detect, and respond to cyberattacks.

To ensure XDR technology works well and as expected, its detection performance can be analyzed using machine learning. The research analyzes and provides accuracy based on specific tools and purposes, like an Intrusion Detection System (IDS), by detection numbers, false alert notifications, and the identification of unknown attacks and prove accuracy more than 99% [12]. ML is a strong cybersecurity tool because it improves systems' ability to identify trends and predict potential data threats [13].

To develop models that can adequately safeguard systems from spyware and malicious behavior, it maximizes the efficacy of training and processing procedures. The technology is revolutionary because it enables systems to learn from data and make logical choices without explicit programming [14].

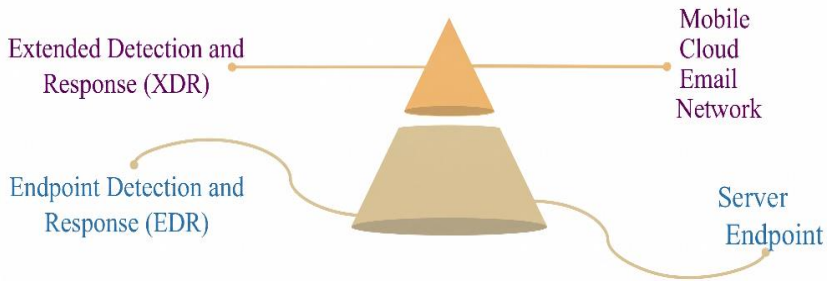


Fig. 1 XDR eliminate EDR limitation detail diagram

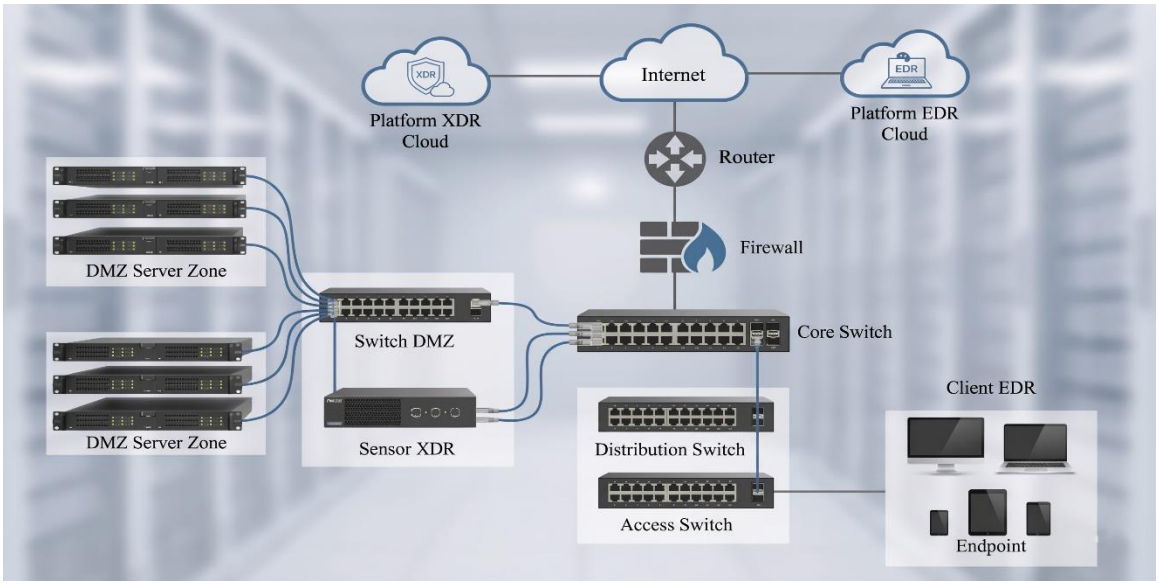


Fig 2. Topology security system using XDR and EDR

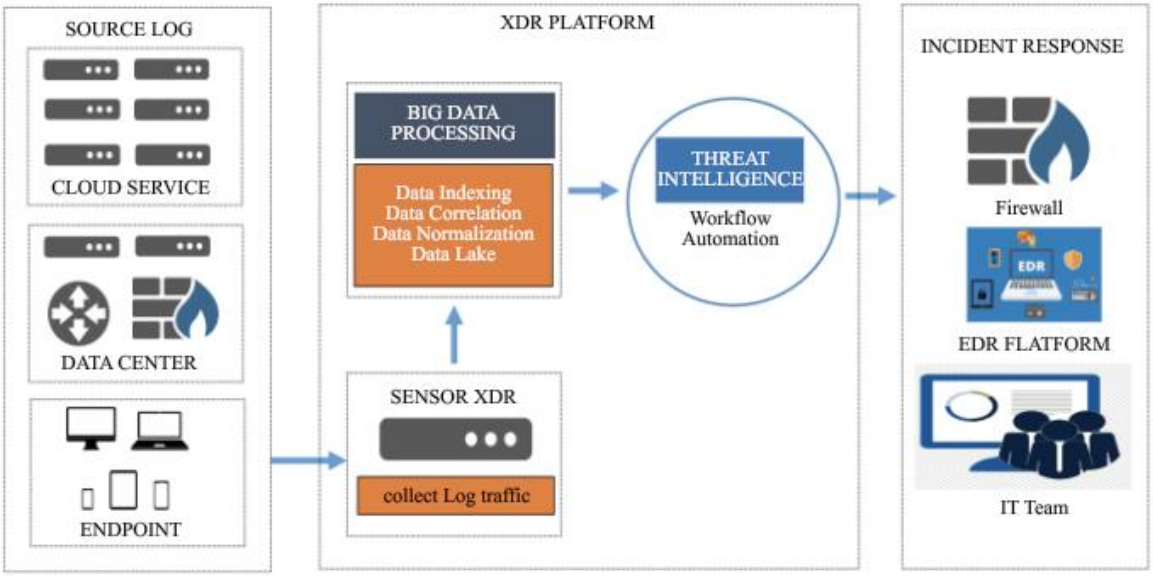


Fig. 3 Proposed cybersecurity system

In the context of IDS, ML algorithms may use historical and live data to identify anomalies and patterns of acceptable behavior, thereby exposing potential security threats. To recognize new and emerging attack vectors, these datasets are trained on a wide range of datasets and will process with test data, the result enables security systems to defend data and networks against unlawful access and malicious activities [15]. This study addresses these obstacles by using an XDR data lake and sensitive information detection, while keeping the template and essential features intact for analysis and enhances threat detection capabilities with Machine Learning (ML).

The study was conducted within an enterprise security system tool with multi source, such as IPS, IDS, and EDR, consolidate in XDR data lake. The XDR system centralizes the monitoring and analysis of security events across network and endpoint layers by integrating telemetry from network and endpoint. The dataset includes over 300,000 security event records collected from the security operations center, benchmarked dataset from the UNSW-NB15 structure and features, with actual data derived from XDR logs. These logs, representing operational security outputs over a monthly period, represent the high volume and rapid rate of alerts, over a million per month.

Due to high volume and challenges, the need to research, analyze and examine security event more essential and develop a detection model that reduces false positives and false negative while enhancing threat identification is the best strategy approach. It demonstrates how advanced analytical

techniques can improve incident detection and response using large security datasets. This research promotes a machine learning development workflow to improve identification and detection models and more effectively methods used in integrated security systems. After that, a comprehensive report detailing the proposed detection model, evaluation outcomes, and conclusions from the data analysis will be provided.

2. Literature Review

By integrating security telemetry from endpoints, networks, the cloud, servers, and workloads to automatically correlate security events and enable rapid response, Extended Detection and Response (XDR) overcomes the limitations of isolated tools like EDR, SIEM, and NTA, which depend on manual forensic analysis [7]. Conversely, several studies demonstrate that Machine Learning (ML) and Deep Learning (DL) methods consistently achieve excellent detection performance over 90% accuracy and with some models achieving 98–100% accuracy on popular intrusion detection datasets.

By using Random Forest and Decision Tree models, Stiawan [16] and Guezzaz [17] get accuracies over 98%, while Hammad [18] and Talita [19] obtain accuracies of up to 99–100% via ensemble learning and enhanced feature selection. These performance results demonstrate that ML-based models are successful at both identifying assaults and reducing false positives by learning patterns of anomalous traffic and minimizing the misclassification of harmless behaviors. However, DL-based methods prove to increase detection accuracy and lower false-positive rates in large-scale, sensor-enabled environments by optimizing DNN, CNN, LSTM, GRU, and semi-supervised and combined supervised capabilities for modeling complex, sequential network behaviors [20-23].

Previous research demonstrates that certain machine learning techniques achieve very high detection performance (>98%), demonstrating their efficacy in detecting cyberattacks and reducing false positives. Guezzaz [17] shows 98.80%–99.42% accuracy with benchmark datasets, Talita [19] achieves 99.12% accuracy with Naïve Bayes mixed with PSO feature selection, and Hammad [18] reports up to 100% accuracy with the T-SNERF model. Research above shows confidence level that collaborate XDR technology and machine learning algorithm will deliver a thorough and effective approach for improve detection accuracy.

Expansion to the earlier inquire about, information mining too plays a pivotal part in cybersecurity and assault discovery. It permits for the examination of huge volumes of arrange activity and security logs to spot peculiarities and noxious exercises. Information mining takes part in separating typical behavior from assaults, empowers early risk location, and brings down wrong positive and wrong negative rates [24-26] by applying classification, clustering, and prescient procedures. The work of administered, unsupervised, and semi-supervised learning moreover empowers security frameworks to adjust to changing assault designs and unidentified dangers within the future. Security frameworks, by nature, can adjust to changing assault designs and unidentified dangers through the utilize of a behavior-driven approach and unsupervised learning, but the genuine operations, it increment untrue positive and false negatives that's why machine learning-based interruption discovery and coordinates security frameworks like Extended Detection and Reaction (XDR), which use correlated information examination to extend discovery precision and reaction adequacy, depend intensely on information mining [7, 25, 26].

To optimized machine learning models, this explore clarifies on five machine learning calculations: Decision Tree, Random Forest, K-Nearest Neighbors (KNN), CatBoost, and Extreme Gradient Boosting (XGBoost) to study area execution on large-scale XDR datasets. These models are utilized for classification errands and reflect some learning benchmarks and judgments. Decision Tree models are especially known in data mining cause its interpretability and capacity to classify complex data without depending on complex, truthful assumptions, making them culminate for analyzing sweeping security event logs [27].

In comparison, the Random Forest approach, which aggregates various decision trees into an furnish, updates estimate exactness and diminishes overfitting by utilizing subjective highlight assurance and bootstrap amassing and outstandingly strong in twofold class [24].

Other models, whereas demonstrate viability depends on legitimate parameter determination and information dissemination, instance-based learning strategies like K-Nearest Neighbor give adaptable, non-parametric classification competent of recognizing neighborhood designs in expansive datasets which the closest design of dataset [27]. To improve predictive precision, this ponder too utilizes progressed outfit boosting calculations such as CatBoost in highlight boosting [28] and XGBoost for boosting in disposing of blunder [29].

CatBoost and XGBoost strategies progress demonstrate execution through continuous model refinement and gradient-boosting systems, which are viable over different classification and expectation errands. The inquire about offer to recognize the foremost compelling demonstrate for identifying malevolent occasions whereas diminishing untrue positives and negatives in cybersecurity checking frameworks by comparing different machine learning procedures on the XDR dataset. Continue with Random Forest algorithm models specific for complex datasets since it minimizes overfitting and combines various choice trees, ordinary it to realize tall accuracy and quality [24] though choice trees handle misplaced data well and offer a clear, different-level structure for classification [30].

KNN may be a coordinate, similarity-based approach sensible for small to medium-sized datasets, but it gets to be not optimal with high-dimensional or colossal data due to extended computational taken a toll [27]. Inside the paper entitled “Comparison of the CatBoost Classifier with other Machine Learning Methods” [29], CatBoost builds on Jerome H. Friedman's (2001) slant boosting framework. It presents how As Ask Boosting diminish estimate move and target spillage, capable categorical incorporate encoding through target-parameter estimations with alter structure, and symmetric (careless) trees to overhaul computational adequacy and regularization. These highlights make strides generalization, diminish overfitting, and boost dauntlessness, particularly with high-cardinality categorical data. Moreover, XGBoost progresses slant boosting by lessening botch and planning speed, making it suite machine learning illustrate for complex and large-scale classification and backslide issues [30].Though machine learning advances accuracy and efficiency, CRISP-DM bridges the gap between data and machine learning models and performance by emphasizing data quality, model selection, and critical evaluation before deployment [31].

The CRISP-DM framework offers a unmistakable perspective by prioritizing a organized, productive handle for actualizing data mining wanders over distinctive businesses [31]. Unlike approaches centered solely on calculations like Decision Tree, Random Forest, KNN, CatBoost, and XGBoost, CRISP-DM highlights six iterative stages: business understanding, data understanding, data preparation, data modeling, evaluation, and deployment to ensure that interpretive models align with business objectives and meet data constraints.

Optimizing machine learning models in overhauling execution area in cybersecurity, these interpretive moves clearly deliver organizational cyber more flexibility when balanced with the NIST Cybersecurity Framework. The NIST framework guides operational security operation in 5 capacities: Identify, Protect, Detect, Respond, and Recover [32], and the CRISP-DM data mining lifecycle [31] make suitable workflow framework. By Made strides asset perceivability and risk mindfulness from machine learning, it fortifies the detect function, while data confirmation sensor and cleverly get to control take work in protecting. Threat detection is extraordinarily advanced, faster, more accurate, and more precise in threat identification.

Predictive analytics and recovery orchestration provide advanced automated classification, action examination, and help endeavors, engaging more reasonable responses. Integrating machine learning models into the NIST and CRISP-DM framework within XDR systems collaboration improves expectations, detection, response, and recovery, coming almost in a more solid and energetic cybersecurity maturity posturing.

3. Research Methodology

This inquire about combining the Cross-Industry Standard Process for Data Mining (CRISP-DM) and the NIST Cybersecurity System (NIST CSF) to create a workflow system and optimizing machine learning calculation in cybersecurity exactness detection. CRISP-DM system gives analytics prepare with six iterative stages: business understanding, data understanding, preparation, modelling, evaluation and deployment. This system guarantees big security datasets are methodically changed into prescient models. In the interim, the NIST CSF adjusts the workflow with best cybersecurity hones through its five centre capacities: Identify, Protect, Detect, Respond, and Recover.

In this consider, the system basically bolsters Identify and detect as part of NIST CSF by cleansing and classification of odd or pernicious occasions inside security telemetry produced by security devices. By connecting CRISP-DM with the NIST CSF, this investigate guarantees that the machine learning models not as it were convey solid predictive capabilities but too encourage compelling threat detection and incident response inside advancing and developing cybersecurity roadmap.

3.1. Business Understanding

This research aims to improve Extended Detection and Response (XDR) systems by refining machine-learning-based threat detection to achieve high accuracy and reduce false positives. Using a CRISP-DM approach, it analyzes an XDR dataset featuring nine attack classes that depict different network traffic and endpoint behaviors. Oversampling techniques address severe class imbalance, followed by Random Forest-based feature selection to identify the most relevant features. Dimensionality reduction is carried out with Principal Component Analysis (PCA), and k-fold cross-validation is employed to prevent overfitting and ensure reliable model evaluation.

3.2. Data Understanding

Using a log system, which represents a variety of network traffic and endpoint behaviors. Specific to this research, the XDR dataset was used and followed the best practices of prior research. The following sections provide a more thorough analysis of each dataset. It contains a large amount of internet traffic and endpoint patterns and constructs 50 features, as shown in Table 1.

The Excel files contain 350.680, respectively. To set it up, the dataset is divided into two groups: a training group and a test group. There are 245.476 entries in the test set and 105.204 in the training set, for a total of 350.680 entries. Along with their corresponding class labels, the dataset includes both manufactured attack events and ordinary conduct from the modern world. From the perspective of today. The top 10 attacks monitored are Risk Tool Miner, Hacktool, Backdoor, Trojan, Downloader, Network Discovery, HackTool, and

worm attacks. These are examples of generic, exploitative, reconnaissance, and analytical tactics. The information in this dataset is terribly unbalanced. The distributions of binary and multi-label attack data, as well as benign data, before and after preprocessing are shown in Figures 4 and 5. A brief explanation of the attack classes is provided in Table 2, and oversampling is performed afterward. The frequency distribution of attack categories is shown in Table 3.

Table 1. Feature of XDR dataset

No.	Independent Variable/Features		
	Variable	**Data Type**	**Description**
1	Event Id	Object	Identification of Alert
2	Timestamp	Object	Time of Alert
3	Rule Message	Object	Rule Message
4	Source ip	Object	Alert IP
5	Destination ip	Object	Destination IP
6	udm.metadata.product_name	Object	Shows the name of the security product or system that generated the event/log.
7	udm.security_result.severity	Object	Shows the severity of the event/log.
8	Event Subtype	Object	Sub-category of the event.
9	Protokol	Object	Shows the network protocol used, for example, TCP, UDP, or ICMP.
10	Service	Object	Service-related examples: HTTP, FTP, SSH.
11	spkts	int64	Number of packets sent by the source.
12	dpkts	int64	The number of packets received at the destination.
13	sbytes	int64	Number of data bytes sent by the source
14	dbytes	int64	The number of data bytes received by the destination.
15	rate	float64	Speed of traffic during the communication session.
16	sttl	int64	Value Time-to-Live (TTL) of the source
17	dttl	int64	Value Time-to-Live (TTL) of the destination.
18	sload	float64	Load traffic of the source (bytes per second).
19	dload	float64	Load traffic of the destination (bytes per second).
20	sloss	int64	Number of packets lost by the source.
21	dloss	int64	Number of packets lost at the destinations.
22	sinpkt	float64	Time-average inter-packet intervals of the sources.
23	dinpkt	float64	Time-average inter-packet intervals for the destinations.
24	sjit	float64	Jitter value (delay variation) of the sources.
25	djit	float64	Jitter value (delay variation) of the destinations.
26	swin	int64	TCP window size of the sources.
27	stcpb	int64	TCP base sequence number of the source.
28	dtcpb	int64	TCP base sequence number of the destination.
29	dwin	int64	TCP window size of the destination.
30	tcprtt	float64	Round-Trip Time TCP (RTT).
31	synack	float64	Time between the packet SYN packet and the SYN-ACK.
32	ackdat	float64	Time between ACK and data send.
33	smean	int64	Average packet size of the source.
34	dmean	int64	Average packet size for the destinations.
35	trans_depth	int64	The number of pipelined HTTP requests in a single connection.
36	response_body_len	int64	HTTP response body size.
37	ct_srv_src	int64	The number of connections from the same source to the same service.
38	ct_state_ttl	int64	The number of connections with the same status and TTL
39	ct_dst_ltm	int64	The number of connections to the same destination in a given time period.
40	ct_src_dport_ltm	int64	The number of connections from the source to the same destination port.

41	ct_dst_sport_ltm	int64	The number of connections to the destination from the same source port.
42	ct_dst_src_ltm	int64	The number of connections between the same source and destination.
43	is_ftp_login	int64	Determines FTP access attempt.
44	ct_ftp_cmd	int64	Amount of FTP scripts used.
45	ct_flw_http_mthd	int64	The number of URL methods used in a single connection flow.
46	ct_src_ltm	int64	Number of connections from the same source in a given time period.
47	ct_srv_dst	int64	Number of connections to the same service and destination.
48	is_sm_ips_ports	int64	Shows whether the source and destination IP and port are the same.
49	attack_cat	object	Category of detected attack
50	Label	Int64	Attack alert or normal alert?

Table 2. Briefly describe all of the attack classes in the XDR dataset

Attack Categories	Brief Description
Fuzzers	The insider attempts to bring down a network, operating system, or piece of software by providing a large amount of random data to discover security flaws, trigger crashes, and cause anomalous behavior.
Botnet C&C, Malicious Host	Malicious software under a compromised server whose purpose is to control and manage the botnet.
HackTool.Cobalt.UDP.Server Request	A well-known tool used for red teaming and penetration testing. It used UDP to execute arbitrary commands on the infected machine, move laterally across the network, exfiltrate data, and drop additional malware.
HackTool.Nessus.SMB.C&C	A well-known tool used for red teaming and penetration testing. It used Server Message Block (SMB), which is commonly used for file sharing, authentication, and enumeration of network resources to scan, and attackers often use or misuse these tools.
CS2	A second-stage command-and-control communication is often stealthier and more encrypted. It is used in malware campaigns, Ransomware, and Advanced Persistent Threat (APT) post.
Botnet C&C	Sends commands to infected devices or bots to a specific target host/endpoint, and sometimes used for a fuzzer or DDoS attack.
RemoteAdmin.RustDesk.UDP .C&C	Remote control tools that attackers can abuse and that are commonly used for Command & Control (C&C).
Downloader.TorLoader.UDP. C&C	A malware event tried to download additional malicious payloads and hide its traffic via the Tor Network.
HackTool.Nessus.TFTP.C&C	Suspicious activity associated with Nessus security assessment tools using Trivial File Transfer Protocol (TFTP) communication for network discovery, file transfer, or interaction with remote systems.
HackTool.Nmap.UDP.C&C	Network reconnaissance activity using Nmap over UDP to identify active hosts, open ports, and available network services, potentially supporting unauthorized network discovery.
NetTool.AdvancedIPScanner. SNMP.C&C	Network scanning activity using Advanced IP Scanner through Simple Network Management Protocol (SNMP) to discover network devices and collect infrastructure information.
NetTool.DefaultCommunity.S NMP.ServerRequest	Suspicious SNMP requests using default community strings to query network devices, potentially exposing device information, configurations, and management interfaces to unauthorized parties.
NetTool.Equinox.UDP.Update r	Network utility updater activity associated with Equinox using UDP communication, potentially involving software update requests or interactions with remote infrastructure.
Trojan.Agentb.UDP.C&C	Trojan-related malicious activity associated with Agentb, involving UDP-based communication with remote command-and-control infrastructure to receive instructions or facilitate unauthorized operations.
Trojan.Miner.UDP.C&C	Cryptocurrency mining malware activity involving communication with remote infrastructure to coordinate mining operations, receive configurations, or utilize compromised system resources.

HackTool.GlobalSocket.UDP.C&C	Suspicious network activity associated with GlobalSocket using UDP communication to establish remote connections or facilitate command-and-control interactions.
NetTool.TunnelVisualStudio.UDP.C&C	Network tunneling activity associated with Visual Studio-related utilities, potentially establishing UDP communication channels for remote access, traffic forwarding, or connectivity to external infrastructure.
HackTool.Nessus.UDP.C&C	Suspicious Nessus-related network scanning activity over UDP to identify exposed hosts, services, and potential vulnerabilities, potentially supporting unauthorized reconnaissance.
NetTool.Ngrok.UDP.C&C	Tunneling activity associated with Ngrok that can expose internal services through externally accessible communication channels, potentially enabling unauthorized remote access and command-and-control communication.
Trojan.Regsup.UDP.ServerRequest	Trojan-related activity associated with Regsup involving UDP-based server requests, potentially indicating communication with remote infrastructure or attempts to retrieve malicious instructions.
HackTool.Nessus.TCP.C&C	Suspicious Nessus-related network scanning activity over TCP to identify open ports, exposed services, and potential vulnerabilities that may be exploited by attackers.
RiskTool.Miner.UDP.C&C	Potentially unwanted cryptocurrency mining activity involving UDP-based communication with remote mining infrastructure, which may consume system resources without the system owner's authorization.
NetTool.AdvancedIPScanner.UDP.C&C	Network discovery activity using Advanced IP Scanner over UDP to identify reachable devices, discover network services, and gather information about the target network.
Trojan.Denes.UDP.C&C	Trojan-related malicious communication associated with Denes, using UDP-based connections to interact with remote command-and-control infrastructure and potentially execute unauthorized activities.
Trojan.Snovir.UDP.ServerRequest	Trojan-related activity associated with Snovir involving UDP-based server requests, potentially indicating communication with remote malicious infrastructure or attempts to retrieve instructions.
Worms	Self-replicating malicious software designed to spread across systems and networks, potentially exploiting vulnerabilities or using network connections to infect additional devices without requiring direct user interaction.
HackTool.Nmap.TCP.C&C	Network reconnaissance activity using Nmap over TCP to identify active hosts, open ports, running services, and potential entry points for further attacks.
Downloader.DriverPack.UDP.C&C	Downloader activity associated with DriverPack involving UDP-based communication to retrieve software components or additional payloads from remote infrastructure, potentially enabling further malicious execution.
Shellcode	Small sequences of executable code delivered or injected during exploitation to execute arbitrary commands, establish unauthorized access, escalate privileges, or initiate additional malicious activities.
NetTool.SoftEtherVPN.UDP.C&C	Network tunneling or VPN-related activity associated with SoftEther VPN using UDP communication, potentially enabling encrypted remote connectivity, traffic redirection, or unauthorized access to internal networks.
HackTool.Nmap.HTTP.C&C	Network reconnaissance activity associated with Nmap involving HTTP-based communication to identify exposed services and gather information about network infrastructure, potentially supporting further attack activities.
Backdoor	Malicious functionality that provides an unauthorized access mechanism to a compromised system, allowing attackers to execute commands, maintain persistence, bypass normal authentication, or remotely control the affected environment.
Analysis	Security-related analysis or diagnostic activity involving the examination of network behavior, system events, or potentially suspicious artifacts. This classification does not independently establish malicious intent and requires additional contextual investigation.
HackTool.Nmap.TCP.ServerRequest	Nmap-related TCP server request activity potentially involving port scanning, service discovery, and network enumeration to identify accessible systems and services within a target environment.
Reconnaissance	Information-gathering activities conducted to identify target systems, network topology, exposed services, vulnerabilities, and potential attack paths before subsequent exploitation or intrusion attempts.

Exploits	Malicious techniques or code that take advantage of software vulnerabilities or system weaknesses to execute unauthorized operations, escalate privileges, bypass security controls, or compromise target systems.
Generic	A generalized security detection classification used to identify suspicious or potentially malicious activities that cannot be assigned to a specific malware family, attack technique, or threat category. Further investigation of event metadata and detection rules may be required to determine the exact threat behavior.
Normal	Legitimate network or endpoint activities that do not exhibit known malicious behavior or match identified threat indicators. This category represents benign system operations and serves as the baseline class for distinguishing normal activities from potentially malicious events.
DoS	Denial-of-Service activity aimed at disrupting system or network availability by overwhelming resources, exhausting bandwidth, consuming processing capacity, or exploiting service limitations.

Table 3. The frequency distribution of attack categories of the XDR dataset

Attack Category	Count	%
Downloader.TorLoader.UDP.C&C	2	0,00%
HackTool.Nessus.TFTP.C&C	2	0,00%
HackTool.Nmap.UDP.C&C	2	0,00%
NetTool.AdvancedIPScanner.SNMP.C&C	2	0,00%
NetTool.DefaultCommunity.SNMP.ServerRequest	2	0,00%
NetTool.Equinox.UDP.Updater	2	0,00%
Trojan.Agentb.UDP.C&C	2	0,00%
Trojan.Miner.UDP.C&C	2	0,00%
HackTool.GlobalSocket.UDP.C&C	6	0,00%
NetTool.TunnelVisualStudio.UDP.C&C	6	0,00%
HackTool.Nessus.UDP.C&C	12	0,00%
NetTool.Ngrok.UDP.C&C	12	0,00%
Trojan.Regsup.UDP.ServerRequest	18	0,01%
HackTool.Nessus.TCP.C&C	26	0,01%
RiskTool.Miner.UDP.C&C	40	0,01%
NetTool.AdvancedIPScanner.UDP.C&C	42	0,01%
Trojan.Denes.UDP.C&C	72	0,02%
HackTool.Cobalt.UDP.ServerRequest	74	0,02%
Trojan.Snovir.UDP.ServerRequest	80	0,02%
HackTool.Nessus.SMB.C&C	106	0,03%
Worms	206	0,06%
HackTool.Nmap.TCP.C&C	298	0,08%
RemoteAdmin.RustDesk.UDP.C&C	334	0,10%
Downloader.DriverPack.UDP.C&C	358	0,10%
Botnet C&C, Malicious host	1206	0,34%
Shellcode	1774	0,51%
NetTool.SoftEtherVPN.UDP.C&C	1922	0,55%
HackTool.Nmap.HTTP.C&C	2168	0,62%
Backdoor	2772	0,79%
Analysis	2906	0,83%
Botnet C&C	8700	2,48%
cs2	8876	2,53%
HackTool.Nmap.TCP.ServerRequest	10630	3,03%
Reconnaissance	16844	4,80%
DoS	20378	5,81%
Fuzzers	26714	7,62%
Exploits	52618	15,00%

Generic	80000	22,81%
Normal	111466	31,79%
Grand Total	350.680	100,00%

3.3. Data Preprocessing

Data gathering from the dataset during the first stage is the initial process. The data is processed using techniques such as outlier identification, removal of null and infinite values, variable correlation analysis, conversion of categorical variables to binary or numerical form (label encoding), and standardization to handle extremely high or low values. The next step is feature selection, which selects the most relevant and informative collection of attributes. Although feature scaling is not strictly required for tree-based algorithms, it was retained as part of the standardized preprocessing pipeline to ensure that all evaluated models received the same transformed feature representation, particularly because KNN is sensitive to feature scale.

PCA was applied to the selected numerical feature space rather than directly to raw categorical variables. Its primary purpose was to reduce feature dimensionality and computational complexity after feature selection. When the data is unbalanced, oversampling is used to resample it. Random forest feature selection is then used to extract the most important or optimal features from the dataset.

Principal component analysis, or PCA, was then used to reduce the most significant data label. Additionally, average accuracy is assessed and trained using k-fold Cross-Validation (CV). Once the data is ready, the learning model is trained using algorithms that identify patterns and make predictions. A variety of datasets are then used to evaluate the performance of models trained with Decision Tree, CatBoost, KNN, Random Forest, and XGBoost. Deep performance processing analysis is performed to assign which model performs best. Finally, optimization improves the model performance result. This process may involve adopting model parameters, selecting different algorithms, or adding further data. This illustrates the complex, comprehensive data processing involved in building an effective learning model, as shown in Figures 4 and 5. Methods used in data processing include outlier detection, removal of null and infinite values, determination of variable correlations, conversion of categorical variables into binary or numerical representations (label encoding), and standardization to handle very high or low values. Use Google Colab Pro+ with a TPU v6e-1 to accelerate model training and evaluation; all tests passed successfully. The TPU v6e-1 with big throughput parallel processing, making it suited for large machine learning processing.

This setup of system grants execute genuinely errands while moving forward the speed and efficiency of running handle. Component utilized in programming are distinctive Python libraries, checking pandas for dataset processing, Optuna for hyperparameter optimization, NumPy for numerical calculations, Scikit-learn for model building and examination, Matplotlib for data descriptive, and Imbalanced-learn for managing course lopsidedness. The system was fully energized tests in a cloud environment.

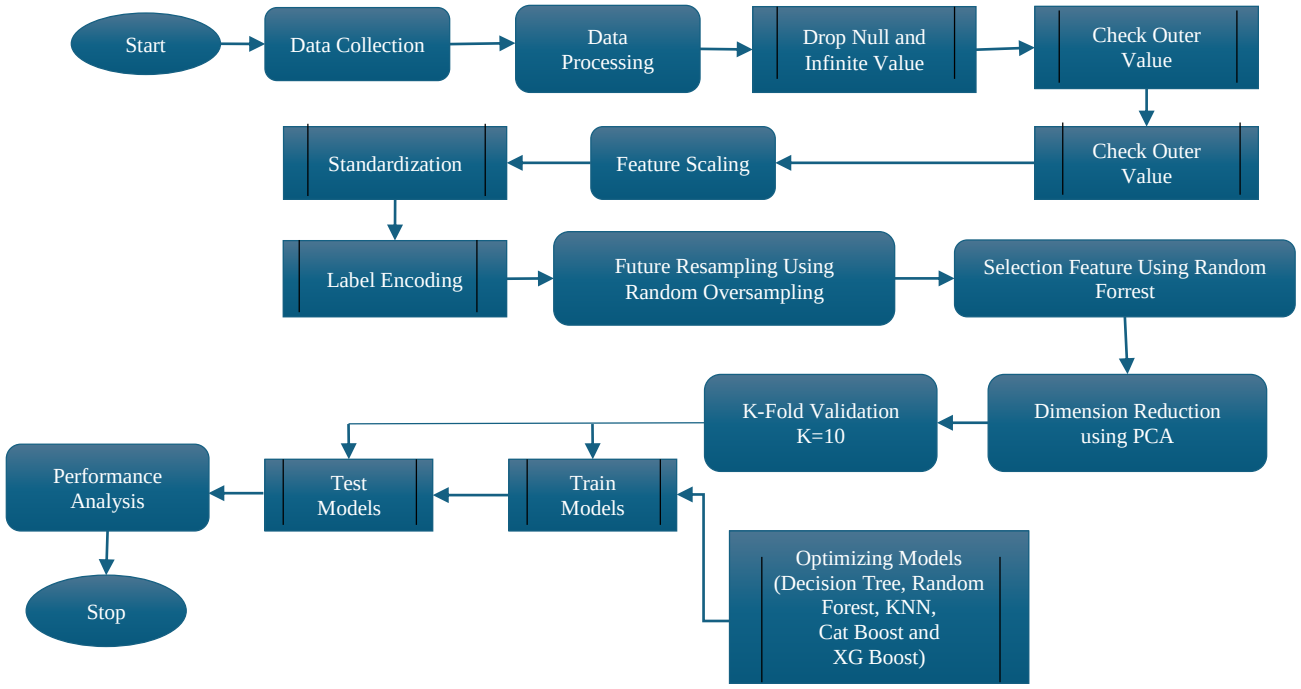


Fig. 4 System flowchart

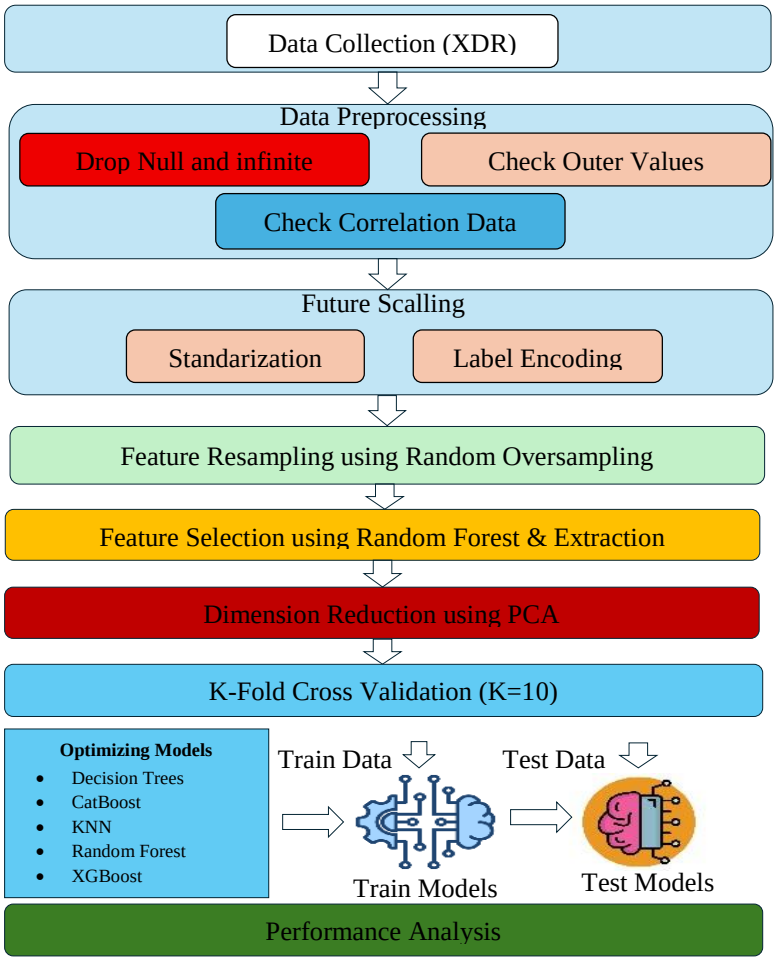


Fig. 5 Design

Result execution estimations such as F1-score, precision, accuracy, recall, ROC curve, and confusion matrix were utilized to assess the model's execution. These estimations offer a comprehensive evaluation of the model's prescient precision, comprehensive, and classification capacity over differing test conditions [33].

3.4. Data Modeling

Machine learning models such as Decision Tree, CatBoost, KNN, Random Forest, and XGBoost are utilized for prescient examination since of their illustrated execution and affect in prior research, that outlined strong capabilities in classification errands.

To help evaluating machine learning classification execution is the confusion matrix, which comprises of four regard combinations:

True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP talks to precisely expected positives, TN appears precisely expected negatives, FP implies to incorrect desires which legal alert and FN implies to off base estimate which illegal alert.

4. Result and discussion

The test results appear in this fragment and illustrative bits of information of the proposed machine learning workflow framework for Extended Detect and Response (XDR). The foremost objective is to review the informative pipeline, which consolidates feature selection, dimensionality diminishment, managing class imbalance, preprocessing, and model optimization, improving cybersecurity alert detection accuracy.

A layout of the workflow is given in Figures 4 and 5, taken after by a detailed clarification of each illustrative step. The review essentially on the workflow's capacity to advance detection accuracy while minimizing false positives and false negatives, which are key performance metrics in XDR circumstances. The dataset comprises XDR event logs with network and endpoint dataset.

4.1. Preprocessing Results

Preprocessing is basic for ensuring data quality, consistent quality, and unflinching quality to model training. It incorporates data cleansing, invalid data checking and data lopsidedness checking.

4.1.1. Data Cleansing

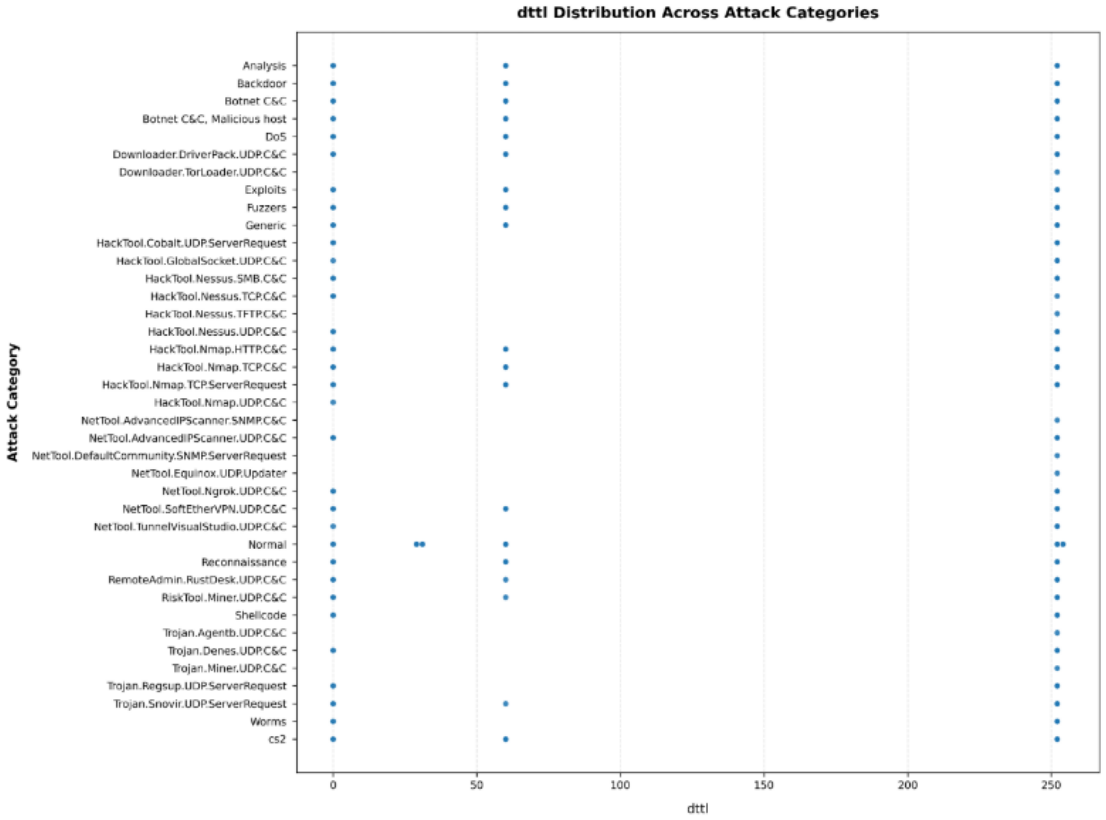
The dataset contains of 50 features decided from XDR logs and included 350,680 records. Data cleaning code was made to apportion with misplaced values and ensure consistency in data sorts over all highlights. Dismember Boundless values and duplicate records were deleted to protect numerical strength. Exemption examination was performed to hold values characteristic of attack patterns while filtering out commotion and anomalous. These steps to create past any question the unflinching quality of dataset a few times as of late proceeding to preprocessing.

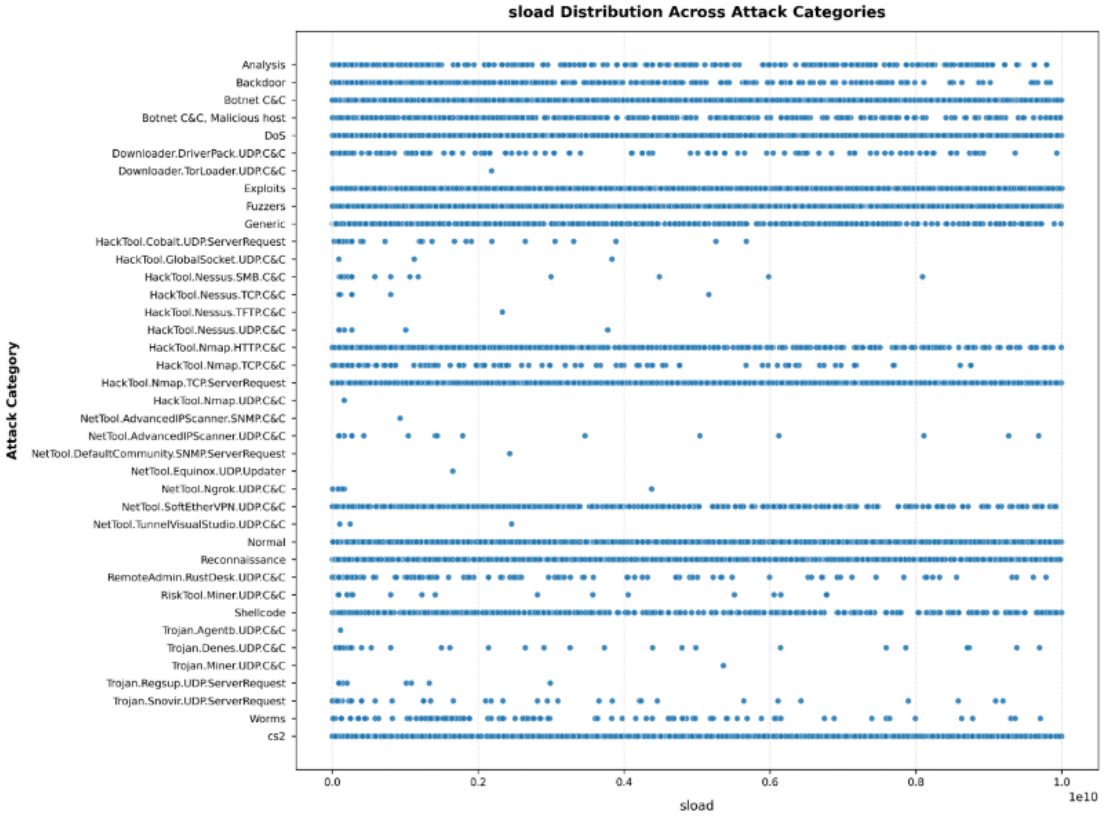
Figure 6 shows up that duplicate data were emptied in the midst of data preprocessing through redress duplicate checks. A thrust was stamped as a duplicate on the off chance that all its highlight values facilitated those of a past thrust. The starting dataset included 175,340 duplicate lines, making up nearly 50% of the beginning records. This cleansing must be executed a few time as of late portion into 70% training and 30% testing sets. This approach is utilized to expect the same discernments from appearing up in both datasets, which may boost execution estimations and engage exact evaluation of the model's generalization capabilities.

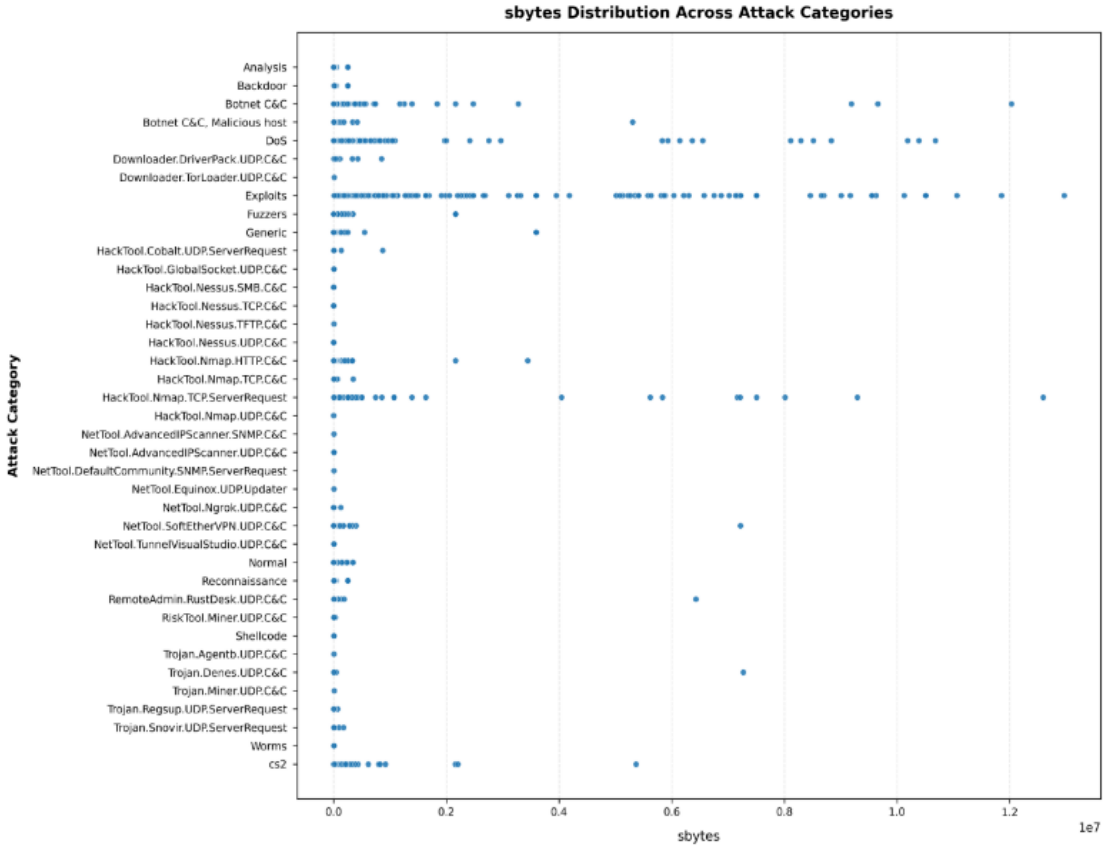
```
#Clean Data sets
total = dataset.shape[0]
missing_columns = [col for col in dataset.columns if dataset[col].isnull().sum() > 0]
for col in missing_columns:
    null_count = dataset[col].isnull().sum()
    per = (null_count/total) * 100
    print(f"{col}: {null_count} ({round(per, 3)}%)")

print(f"Number of duplicate rows: {dataset.duplicated().sum()}")

Number of duplicate rows: 175340
```

Fig. 6 Data cleansing

4.1.2. Correlation Analysis




sload Distribution Across Attack Categories
Attack Category
Analysis
Backdoor
Botnet C&C
Botnet C&C, Malicious host
DoS
Downloader.DriverPack.UDP.C&C
Downloader.TorLoader.UDP.C&C
Exploits
Fuzzers
Generic
HackTool.Cobalt.UDP.ServerRequest
HackTool.GlobalSocket.UDP.C&C
HackTool.Nessus.SMB.C&C
HackTool.Nessus.TCP.C&C
HackTool.Nessus.TFTP.C&C
HackTool.Nessus.UDP.C&C
HackTool.Nmap.HTTP.C&C
HackTool.Nmap.TCP.C&C
HackTool.Nmap.TCP.ServerRequest
HackTool.Nmap.UDP.C&C
NetTool.AdvancedIPScanner.SNMP.C&C
NetTool.AdvancedIPScanner.UDP.C&C
NetTool.DefaultCommunity.SNMP.ServerRequest
NetTool.Equinox.UDP.Updater
NetTool.Ngrok.UDP.C&C
NetTool.SoftEtherVPN.UDP.C&C
NetTool.TunnelVisualStudio.UDP.C&C
Normal
Reconnaissance
RemoteAdmin.RustDesk.UDP.C&C
RiskTool.Miner.UDP.C&C
Shellcode
Trojan.Agentb.UDP.C&C
Trojan.Denes.UDP.C&C
Trojan.Miner.UDP.C&C
Trojan.Regsup.UDP.ServerRequest
Trojan.Snovir.UDP.ServerRequest
Worms
cs2
0.0
0.2
0.4
0.6
0.8
1.0
1e10
sload


sbytes Distribution Across Attack Categories
Attack Category
Analysis
Backdoor
Botnet C&C
Botnet C&C, Malicious host
DoS
Downloader.DriverPack.UDP.C&C
Downloader.TorLoader.UDP.C&C
Exploits
Fuzzers
Generic
HackTool.Cobalt.UDP.ServerRequest
HackTool.GlobalSocket.UDP.C&C
HackTool.Nessus.SMB.C&C
HackTool.Nessus.TCP.C&C
HackTool.Nessus.TFTP.C&C
HackTool.Nessus.UDP.C&C
HackTool.Nmap.HTTP.C&C
HackTool.Nmap.TCP.C&C
HackTool.Nmap.TCP.ServerRequest
HackTool.Nmap.UDP.C&C
NetTool.AdvancedIPScanner.SNMP.C&C
NetTool.AdvancedIPScanner.UDP.C&C
NetTool.DefaultCommunity.SNMP.ServerRequest
NetTool.Equinox.UDP.Updater
NetTool.Ngrok.UDP.C&C
NetTool.SoftEtherVPN.UDP.C&C
NetTool.TunnelVisualStudio.UDP.C&C
Normal
Reconnaissance
RemoteAdmin.RustDesk.UDP.C&C
RiskTool.Miner.UDP.C&C
Shellcode
Trojan.Agentb.UDP.C&C
Trojan.Denes.UDP.C&C
Trojan.Miner.UDP.C&C
Trojan.Regsup.UDP.ServerRequest
Trojan.Snovir.UDP.ServerRequest
Worms
cs2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e7
sbytes

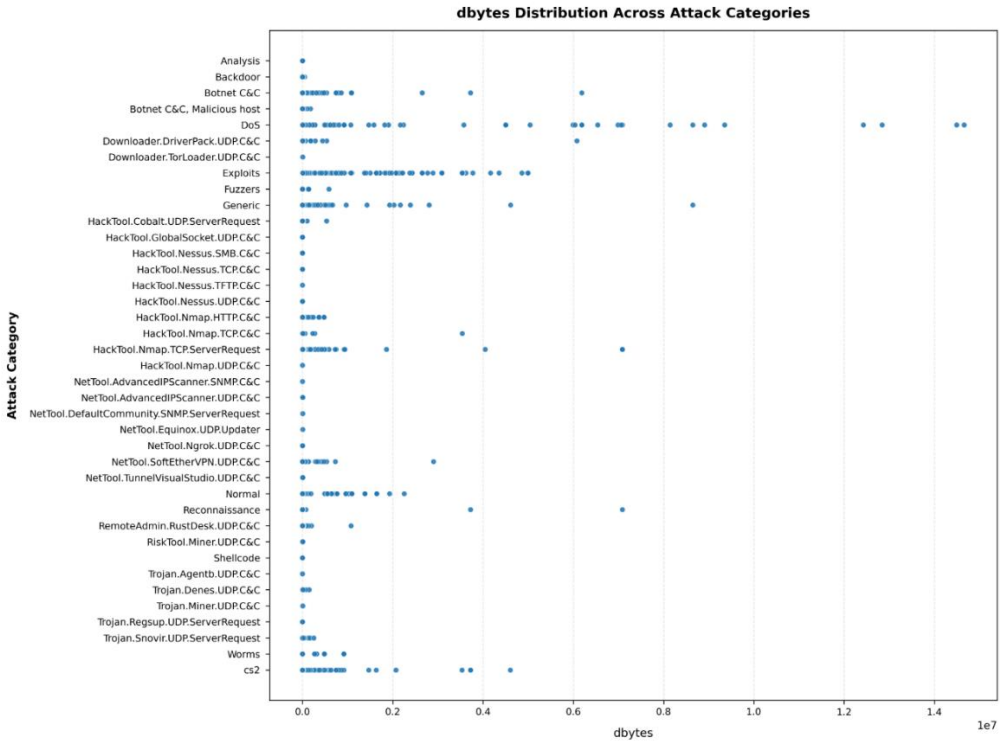


Fig. 7 Correlation analysis

A relationship arrange was printed to study associations among highlights and recognize duplicate properties that might impact the model's consistent quality and ampleness. The examination included fifty parameters, such as affiliation, traffic intensity, timing characteristics, protocol, and application-layer interactions, covering various behavioral viewpoints.

Visualization comes almost of some highlights, dttl, sload, dbytes, and dbytes (Figure 7) illustrate that behavioral varieties from the standard, traffic concentrated patterns, and affiliation go over components, rather than not dynamic protocol characteristics, are the key differentiators of intrusion behavior.

Based on correlation strength and predictive relevance, 39 out of 50 features were retained as useful predictors. The feature distribution was highly right-skewed with several extreme outliers; malicious traffic shows greater dispersion and farther-out dots, while normal traffic clusters more stably. This feature is crucial for intrusion detection, as elevated values may indicate irregular payload transfer, data exfiltration, or malware indicator.

4.1.3. Future Selection Result

Features were chosen through Recursive Feature Elimination (RFE), utilizing a Random Forest classifier as the estimator. This approach was used because Random Forest can reliably estimate feature-importance in high-dimensional, coupled network traffic data and accurately identify nonlinear interactions. Before choosing features, the training data was balanced using resampling. $\chi_{resampled}, y_{resampled}$ to mitigate bias toward the majority class. The RFE procedure retained 50 features, including:

- Network traffic metrics: spkts, dpkts, sbytes, dbytes, rate, sload, dload
- Network stability & timing: sloss, dloss, sinpkt, dinpkt, sjit, djit
- Indicator of Transport layer: sttl, dttl, swin, stcpb, dtcpb, dwin, tcprtt, synack, ackdat
- Indicator of Connection metrics: ct_srv_src, ct_srv_dst, ct_src_ltm, ct_dst_ltm, ct_dst_src_ltm
- Application layer indicators: response_body_len, trans_depth, ct_flw_http_mthd
- Authentication & behavior signals: is_ftp_login, ct_ftp_cmd
- Contextual security metadata: severity, rule message, product name, service, protocol

4.2. Dimension Reduction and Optimization

4.2.1. Dimensions Reduction

Principal Component Analysis (PCA) was used as a dimensionality reduction technique to increase computing performance and reduce feature redundancy. In order to maintain the maximum variance in the data, PCA converts the original feature space into a new set of orthogonal components [12]. The balanced dataset $(X_{resampled})$ was utilized as PCA input following feature selection and resampling.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=20)
pca.fit(X_resampled)
X_pca = pca.transform(X_resampled)
```

The number of main components was set at 20 because more features were analyzed, accuracy decreased, and more resources were needed. PCA improved processing performance, reduced feature redundancy, and improved model generalization.

4.2.2. Dimension Optimization

Optuna, an automated hyperparameter tuning framework that uses effective search techniques to find ideal parameter combinations, was used to optimize hyperparameters in order to increase predictive performance.

```
rf_study = optuna.create_study(direction='maximize')
rf_study.optimize(objective, n_trials=30)
print(rf_study.best_trial)
```

Using the specified objective function, Optuna conducted 30 trials of optimization in which it investigated various combinations of hyperparameters. Every feature was assessed using performance indicators such ROC-AUC, F1-score, and accuracy.

4.2.3. K-Fold Cross Validation

To validate performance evaluation of machine learning models and strong generalization, 10-fold cross-validation was used during model validation which value equal 10 means it will running test data 10 times [12]. Optuna used as tools to optimized before perform K-Fold validation and after dimension reduction using PCA execute.

```
rf.fit(x_train, y_train)
scores_rf = cross_validate(rf, x_train, y_train, cv=10, scoring='accuracy')
scores_avg_rf = np.average(scores_rf['test_score'])
print(f"Accuracy: {scores_avg_rf}")
```

The dataset is divided into ten equal subsets by K-Fold Cross-Validation, each subset is used once as a validation set, and the remaining folds are utilized for training. To create a trustworthy final score, performance data from each fold are averaged.

4.3. Performance Result and Analysis

This section determines the performance evaluation of five machine learning models for intrusion detection: Decision Tree, Random Forest, K-Nearest Neighbors (KNN), CatBoost, and XGBoost. To achieve reliable and objective performance, the dataset was divided into 70% training data and 30% testing data. Ten-fold cross-validation was performed on the training dataset to estimate the model's average accuracy during model evaluation.After model training, the final model was evaluated using the independent 30% testing dataset. Precision, recall, F1-score, and accuracy were calculated using the classification report.

Table 4 displays the performance comparison. All models performed well in classification, according to the data, however ensemble-based methods showed better accuracy and stability.

4.3.1. Comparative Evaluation Result of Accuracy Performance:

- Random Forest - 99.64%
- Decision Tree - 95.77%
- XGBoost - 95.24%
- KNN - 91.36%
- CatBoost - 87.37%

4.3.2. Key Observations

- Random Forest demonstrated its appropriateness for complex behavioral cybersecurity datasets by outperforming all other models.
- Ensemble models (Random Forest and XGBoost) provided more stable and robust performance than single learners.
- CatBoost showed lower accuracy, suggesting that PCA transformation of numerical features may reduce its advantage in handling categorical variables.
- KNN performance may be affected by high dimensionality and distance-based sensitivity.

Table 4. Performance result of 5 models machine learning algorithm

No.	Machine Learning Model	PCA	TRAIN DATA (%)	K-Fold Cross Validation	Accuracy (%)	Precision (%)	F1 (%)	Recall (%)	ROC
1	Decision Tree	20	70	10	95.77	97.00	97.00	97.00	99.90
2	Random Forest	20	70	10	99.64	99.67	99.67	99.67	100
3	KNN	20	70	10	91.36	93.00	92.00	92.00	97.40
4	CatBoost	20	70	10	87.37	87.00	87.00	87.00	99.76
5	XGBoost	20	70	10	95.24	95.00	95.00	95.00	100

4.3.3. ROC Performance Interpretation

Figure 8 shows the ROC curves and delivers confidence AUC values for Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), CatBoost (CB), and XGBoost (XGB) are all close to the ideal 99%, indicating strong class separation ability. Among these, RF and XGBoost achieve perfect scores of 100%, while DT reaches 99.90%, Catboost gets 99.76%, and KNN scores 97.40%. AUC values demonstrate good predictive performance on the XDR dataset and confirm their effectiveness in accurately differencing normal from malicious traffic [12].

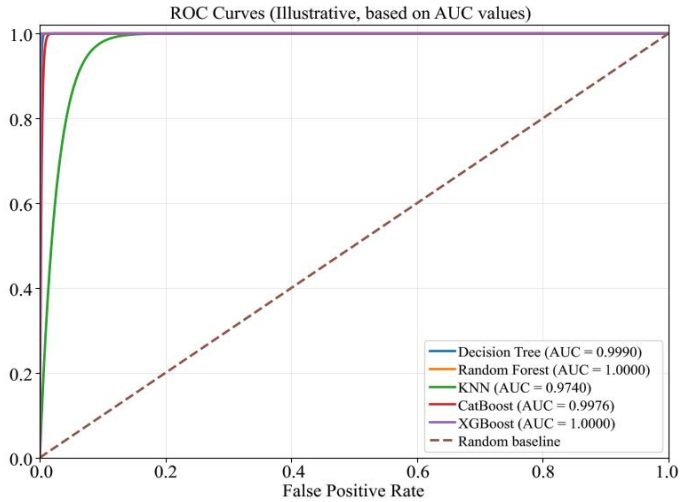


Fig. 8 ROC result

4.3.4. Fundamental Highlights Distinguished by Significance Investigation

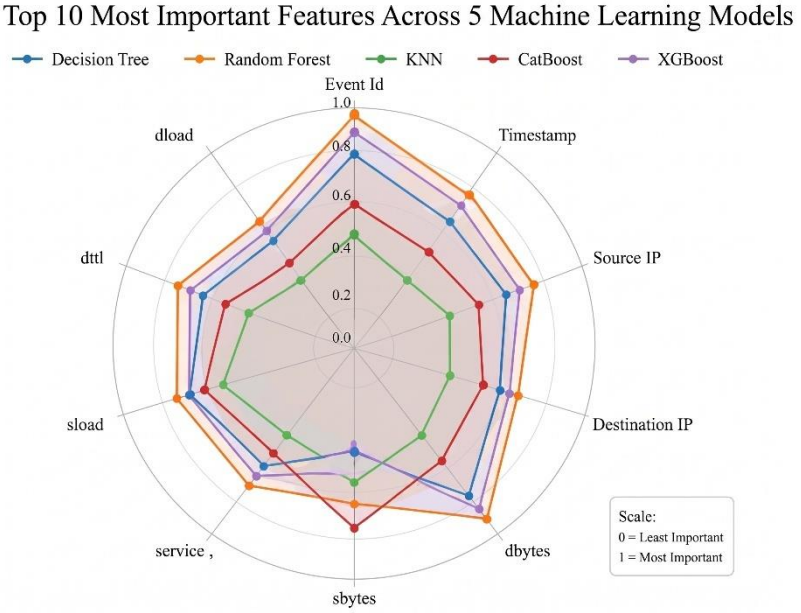


Fig. 9 Top 10 Selection feature all models

Figure 9 shows up that beat 10 highlights and important metadata play basic parts inside the machine learning models in the XDR framework. Behavioral pointers such as dttl, dbytes, sload, sbytes effectively recognize irregularity action, instability network, and attack pattern. Additional highlights like Event ID, timestamp, Source IP, Destination IP provide further discrimination. Strikingly, XGBoost highlights the noteworthiness of application-layer shrewdly, distributing high priority to protocol-related behavior. Overall, the ensemble models, especially Random Forest and XGBoost, reasonably combine these varied features, updating classification precision. In general, robust identification of intricate attack patterns, decreased misclassification, and more precise and scalable cybersecurity analytics are made possible by the integration of high-impact feature or behavioral metrics and contextual factors.

4.3.5. Confusion Matrix

The confusion matrix shown in Figure 10, resulting from the five machine learning models-Decision Tree, Random Forest, KNN, CatBoost, and XGBoost-demonstrates strong classification performance, as indicated by the dominant concentration of values along the diagonal, which represents correctly classified instances for each attack category.

Near-perfect classification with low off-diagonal values is demonstrated by Random Forest and XGBoost, demonstrating their better capacity to reliably differentiate between various attack types and typical traffic. Despite having a few more misclassifications than ensemble models, decision trees likewise exhibit good accuracy. Other models like KNN exhibit sensitivity to the feature space and distance metric; some values are off the diagonal, suggesting subpar performance.

Although CatBoost is generally effective, it exhibits somewhat increased misclassification in some classes, indicating that its effectiveness might be affected by altering feature representations following PCA. Generally, the confusion Matrices support the appropriateness of ensemble-based methods for multi-class intrusion detection in XDR contexts by confirming that they offer more accurate and stable classification, successfully lowering false positives and false negatives.

4.4. Discussion

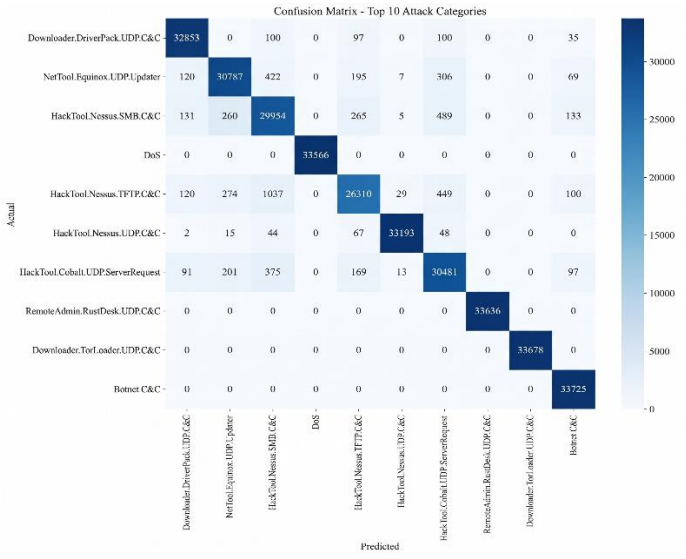
By addressing three key issues-improving detection workflow efficiency, controlling class imbalance, and determining the most pertinent features for dimensionality reduction-this study aims to create an efficient machine learning-driven workflow for Extended Detection and Response (XDR) datasets.

4.4.1. Effectiveness of the Proposed XDR Workflow Framework

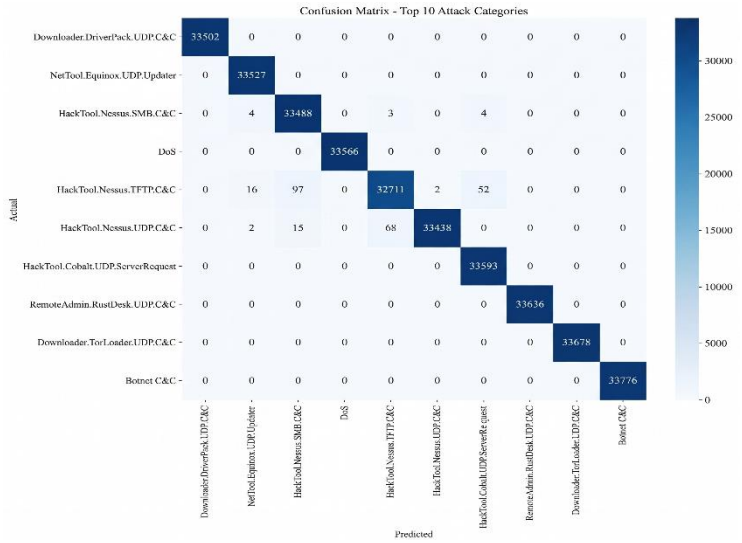
The results of the experiment demonstrate that an effective XDR technique may be produced by integrating feature engineering, dimensionality reduction, data preparation, and ensemble machine learning models. Using contextual event metadata and behavioral feature properties in the dataset, the proposed method effectively ranks alerts and classifies security occurrences. The procedure improves security operations by removing alert fatigue and speeding up incident response by reducing false positives and false negatives. Strong detection capabilities and dependable distinction between innocuous and harmful events are suggested by high ROC scores close to 100% and near-perfect recall values, especially from the Random Forest model. This performance suggests that the framework can significantly reduce the time required for security analysts to investigate alerts and respond to threats [9]. Combining with cross-validation and hyperparameter adjustment increases model reliability and ensures consistent performance in a variety of traffic circumstances. Thus, the workflow increases operational efficiency by enabling automatic, accurate, and scalable threat identification.

4.4.2. Managing Class Imbalance in XDR Datasets

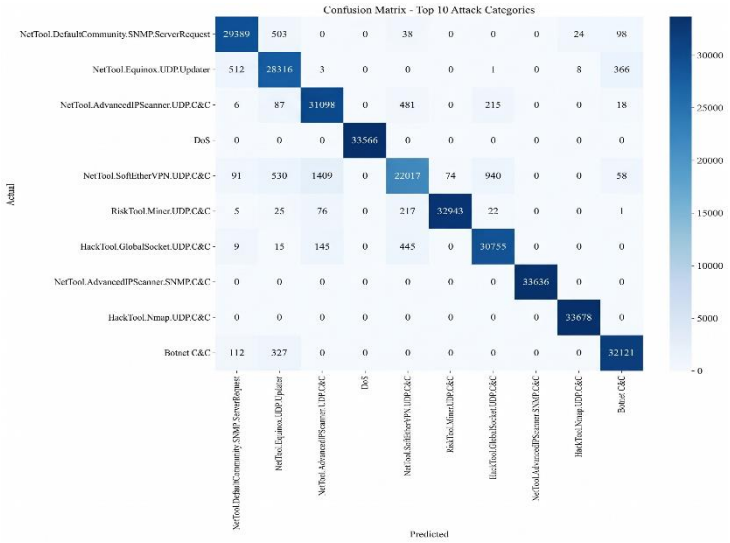
Class imbalance during the preprocessing stage represents the main challenge in intrusion detection because anomaly events are significantly less frequent than benign activities. Without balancing, machine learning models tend to favor the majority class, resulting in high false negative rates and missed attacks. By using resampling techniques on the training dataset, this study addresses the imbalance problem. The balanced dataset improved recall performance and increased the model's sensitivity to minority attack classes. The findings show that ensemble learning models-Random Forest and XGBoost in particular-effectively use balanced data to identify uncommon attack patterns.



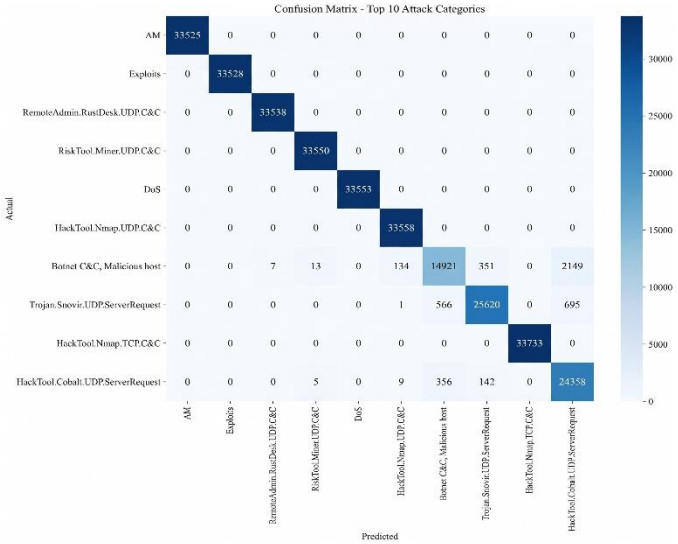
(a) Decision Tree



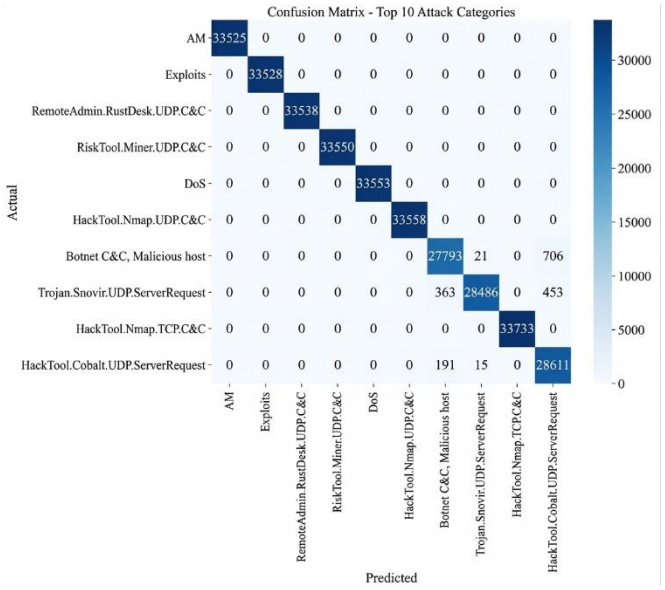
(b) Random Forest



(c) KNN



(d) CatBoost



(e) XGBoost

Fig. 10 Confusion matrix all model

Class balance is important in machine learning to ensure accurate identification of minority attack categories, as shown by improvements in recall and F1-score. The suggested approach reduces the risk of undetected threats and increases the false alert detection system's overall robustness by improving the representation of rare attack samples.

4.4.3. Importance of Feature Extraction and Dimensionality Reduction

Improving model performance and computing efficiency was mostly dependent on feature selection and dimensionality reduction. The most important behavioral and environmental characteristics that greatly influence categorization accuracy were found using Recursive Feature Elimination (RFE). By converting correlated variables into a compact set of orthogonal components while maintaining important variance, Principal Component Analysis (PCA) further decreased redundancy. The most significant features of malicious activity, according to feature importance analysis, are traffic intensity measurements (such as byte volume, load, and rate), timing anomalies (such as jitter and inter-packet timing), TTL behavior, and session characteristics. Additionally, other characteristics, traffic spikes, programmed patterns, and unstable connections are behavioral markers of modern cyberattacks. Because PCA may reduce dimensionality without sacrificing predictive performance, it is a useful tool for managing high-dimensional XDR datasets. Dimensionality decrease improved processing efficiency and model generalization without sacrificing detection accuracy.

4.4.4. Behavioral Characteristics of Malicious Traffic

The study verifies that harmful network activity is best described by behavioral patterns rather than static characteristics. Repetitive connection attempts, high traffic bursts, erratic timing, anomalous packet loss, and unusual session interactions are all characteristics of attack traffic. Malicious alerts differ from normal alerts through coordinated anomalies across various behavioral features. This examination supports behavior-driven in detection procedures, as conventional signature-based or rule-based approaches may miss new threat techniques [10].

5. Conclusion

This experiment about concludes fundamental highlights:

- XDR workflow was made by coordination behavioral incorporate illustrating, dimensionality diminish, and validation. This approach minimizes false positives and false negatives and enliven threat examination.
- Research indicates that managing class imbalance is crucial for effective intrusion detection. Resampling techniques help prevent missing significant threats bias dataset.
- Key feature extraction on traffic intensity, timing irregularities, TTL behavior, and session traits offers the most effective way to reduce dimensions and enhance intrusion detection models.

The proposed architecture at the end improves cybersecurity functions in evolving network settings, reduces analyst alert fatigue, and boosts detection accuracy.

5.1. Research Contributions

This research advances the cybersecurity and intrusion detection fields by:

- Proposing a behavior-driven XDR workflow framework to be implemented by the provider to strengthen cybersecurity,
- Demonstrating effective class imbalance mitigation strategies due to the large amount of data processing,

- Improving detection performance by identifying critical behavioral features for alert detection through ensemble learning and optimization.

5.2. Future Research Directions

There are multiple areas for future research, recommendations, and improvement.

- Real-Time Detection and Streaming Data Processing: Future studies should explore real-time deployment using streaming frameworks to enable continuous threat detection in live network environments.
- Exploring ensemble learning with deep learning methods like Neural Networks could enhance the detection of complex attack patterns.
- Future research may evaluate advanced techniques in generating oversampling to improve minority class detection further.
- Combination with AI for Security Operations: Integrating explainable AI techniques can enhance trust and transparency by helping analysts understand why alerts are triggered.
- Multi-Source XDR Correlation: Next machine learning research should take into research more endpoint telemetry to enhance the capability of XDR detection.

5.3. Final Remark

As computerized alter development exponentially makes cyber threats rise more advanced and complex, behavior-driven machine learning calculation offers a promising course of action for making brilliantly flexible and capable security systems. The framework showed how combining data altering, dimensionality diminishment, and gathering learning can make strides area execution and operational viability, making a contrast cutting edge security operations investigate dynamically complex, progressed scenes procedure.

Acknowledgements

Erik Hadianto take portion in data collecting, examination and documentation. He also created and drafted the first version of the manuscript and rectifications after getting co-author comments, recommendations, and feedback. Astari Retnowardhani, as a co-author, contributed productive admonishment and fast feedback in the midst of the article change. She guided the manuscript and presentation of the disclosures and overviewed the drafts to ensure academic meticulousness and coherence.

Data Availability Statement

XDR dataset utilized in this examination is sensitive; it'll remain confidential and will not be shared.

References

[1] Meltwater, Datareportal, Digital 2026 Global Overview Report, Datareportal, 2026. [Online]. Available: https://datareportal.com/reports/digital-2026-global-overview-report

[2] Marr Bernard, "How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read," *Forbes*, vol. 21, pp. 1-5, 2018. [Google Scholar]

[3] Estefania Vergara Cobos et al., "A Review of the Economic Costs of Cyber Incidents," *World Bank*, Washington, DC, USA 2026. [Google Scholar]

[4] IndoSec, The Escalating Cyber Threat in Indonesia: A Wake-Up Call for Digital Security, Tradepass, 2026. [Online]. Available: https://indosecsummit.com/the-escalating-cyber-threat-in-indonesia-a-wake-up-call-for-digital-security/

[5] Kompas, Potensi Risiko Kejahatan Siber 2024 Makin Kompleks, Kompas, 2026. [Online]. Available: https://www.kompas.id/artikel/potensi-risiko-kejahatan-siber-masih-berlanjut-pada-2024

[6] Deloitte, 2023 Hot Topics for IT Internal Audit, 2015. [Online]. Available: https://www.deloitte.com/uk/en/Industries/financial-services/perspectives/hot-topics-it-internal-audit.html

[7] A. Shaji George et al., "XDR: The Evolution of Endpoint Security Solutions-Superior Extensibility and Analytics to Satisfy the Organizational Needs of the Future," *Article in International Journal of Advanced Research in Science Communication and Technology*, vol. 8, no. 1, pp. 493-501, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Arlina Laras, Financial, Begini Serangan Ransomware BSI Tahun Lalu, Mirip dengan Penyebab PDN Down?, Financial, 2024. [Online]. Available: https://finansial.bisnis.com/read/20240627/90/1777564/begini-serangan-ransomware-bsi-tahun-lalu-mirip-dengan-penyebab-pdn-down

[9] Anton Chuvakin, Medium, Anton's Alert Fatigue: The Study, Medium, 2024. [Online]. Available: https://medium.com/anton-on-security/antons-alert-fatigue-the-study-0ac0e6f5621c

[10] Dedi Soleman, and Benfano Soewito, "Information Security System Design using XDR and EDR," *Inform: Scientific Journal in the Field of Information and Communication Technology*, vol. 9, no. 1, pp. 51-57, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] paloalto, What is XDR (Extended Detection and Response)?, paloalto, 2026. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-extended-detection-response-XDR

[12] Md. Alamin Talukder et al., "Machine Learning-based Network Intrusion Detection for Big and Imbalanced Data using Oversampling, Stacking Feature Embedding and Feature Extraction," *Journal of Big Data*, vol. 11, no. 1, pp. 1-44, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Iqbal H. Sarker et al., "Cybersecurity Data Science: An Overview from Machine Learning Perspective," *Journal of Big Data*, vol. 7, no. 1, pp. 1-29, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Fatima Hussain et al., "Machine Learning in IoT Security: Current Solutions and Future Challenges," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1686-1721, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Md Abdullahel Kafi, and Nazma Akter, "Securing Financial Information in the Digital Realm: Case Studies in Cybersecurity for Accounting Data Protection," *American Journal of Trade and Policy*, vol. 10, no. 1, pp. 15-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Kurniabudi et al., "CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection," *IEEE Access*, vol. 8, pp. 132911-132921, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Hanna Attou et al., "Cloud-based Intrusion Detection Approach using Machine Learning Techniques," *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 311-320, 2023. [Google Scholar]

[18] Mohamed Hammad, Nabil Hewahi, and Wael Elmedany, "T-SNERF: A Novel High Accuracy Machine Learning Approach for Intrusion Detection Systems," *IET Information Security*, vol. 15, no. 2, pp. 178-190, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] A.S. Talita, O.S. Nataza, and Z. Rustam, "Naïve Bayes Classifier and Particle Swarm Optimization Feature Selection Method for Classifying Intrusion Detection System Dataset," *Journal of Physics: Conference Series*, vol. 1752, no. 1, pp. 1-9, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Meenakshi Mittal, Krishan Kumar, and Sunny Behal, "Deep Learning Approaches for Detecting DDoS Attacks: A Systematic Review," *Soft Computing*, vol. 27, no. 18, pp. 13039-13075, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Suad A. Alasadi, and Wesam S. Bhaya, "Review of Data Preprosessing Techniques in Data Mining," *Journal of Engineering and Applied Sciences*, vol. 12, no. 16, pp. 4102-4107, 2017. [Google Scholar]

[22] Mohamed Amine Ferrag et al., "Deep Learning-based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0," *Electronics (Switzerland)*, vol. 10, no. 11, pp. 1-26, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Ömer KASIM, "An Efficient and Robust Deep Learning based Network Anomaly Detection against Distributed Denial of Service Attacks," *Computer Networks*, vol. 180, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Jaime Lynn Speiser et al., "A Comparison of Random Forest Variable Selection Methods for Classification Prediction Modeling," *Expert Systems with Applications*, vol. 134, pp. 93-101, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[25] Salma Khan, and Muhammad Shaheen, "From Data Mining to Wisdom Mining," *Journal of Information Science*, vol. 49, no. 4, pp. 952-975, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] Dimitrios Papakyriakou, and Ioannis S. Barbounakis, "Data Mining Methods: A Review," *International Journal of Computer Applications*, vol. 183, no. 48, pp. 5-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Shichao Zhang, "Challenges in KNN Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4663-4675, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] L.U. Ying, "Decision Tree Methods: Applications for Classification and Prediction," *Shanghai Archives of Psychiatry*, vol. 27, no. 2, pp. 130-135, 2015. [Google Scholar]

[29] Abdullahi A. Ibrahim et al., "Comparison of the CatBoost Classifier with other Machine Learning Methods," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 11, pp. 738-748, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[30] Santhanam Ramraj et al., "Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets," *International Journal of Control Theory and Applications*, vol. 9, no. 40, pp. 651-662, 2016. [Google Scholar]

[31] Christoph Schröer, Felix Kruse, and Jorge Marx Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," *Procedia Computer Science*, vol. 181, pp. 526-534, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Osama Aljumaiah et al., "Analyzing Cybersecurity Risks and Threats in IT Infrastructure based on NIST Framework," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 2, pp. 12-26, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[33] Barus, Ertina Sabarita et al., "Development of Machine Learning for Forecasting Optimization Implemented in Morphology Plant Growth," *Eastern-European Journal of Enterprise Technologies*, vol. 3, no. 2, pp. 42-53, 2025. [CrossRef] [Google Scholar] [Publisher Link]