A. Lara-Gutierrez, C. Fernandez-Gago and J. A. Onieva, "Towards maintainable AI-driven network anomaly and threat detection: a comparative analysis of datasets, preprocessing techniques, and model trade-offs", *Artificial Intelligence Review*, 2026.
http://doi.org/10.1007/s10462-026-11667-7
NICS Lab. Publications: https://www.nics.uma.es/publications

Artificial Intelligence Review manuscript No.
(will be inserted by the editor)

Towards Maintainable AI-Driven Network Anomaly and Threat Detection: A Comparative Analysis of Datasets, Preprocessing Techniques, and Model Trade-offs

Antonio Lara-Gutierrez [a,1], **Carmen Fernandez-Gago** [b,1], **Jose A. Onieva** [c,1]

[1]Network, Information and Computer Security (NICS) Lab, University of Málaga, 29010 Málaga, Spain

Received: date / Accepted: date

Abstract As the integration of Artificial Intelligence into network intrusion detection systems matures, a critical gap remains in the rigorous empirical benchmarking of datasets, preprocessing techniques, and model effectiveness. This article presents a comparative experimental study of anomaly and threat detection techniques used in network analysis through a multistep pipeline. First, we perform a structured comparison and Exploratory Data Analysis of the most commonly used network security datasets to quantitatively assess their balance, diversity, and real-world representativeness. Subsequently, we experimentally evaluate the performance of distinct Machine Learning, Deep Learning, and Hybrid Models derived under standardized preprocessing techniques to determine the most effective combinations for specific attack vectors. Our results demonstrate that hybrid architectures achieve superior generalisation, yet face challenges regarding computational overhead and cross-dataset adaptability. Addressing these limitations, we propose a proof-of-concept adaptive architecture designed to handle concept drift and adversarial threats. Finally, we outline a roadmap for future research, emphasizing the necessity of dynamic, verifiable AI-driven systems that operate reliably in evolving cybersecurity environments.

Keywords Artificial Intelligence · Anomaly Detection · Cybersecurity · Feature Engineering · Adversarial Attacks · Concept Drift

1 Introduction

The integration of Artificial Intelligence (AI) into cybersecurity has become one of the most impactful technological shifts in recent years, particularly in enhancing the capabilities of network analysis, intrusion detection, and automated threat response. This intersection of technologies simplifies the development of solutions across multiple security domains, such as detection, prevention, response, and resilience. This paper presents a critical analysis of anomaly and threat detection datasets and AI models.

Both anomalies and threats are closely related concepts; however, they represent different behaviours and imply different responses. Threats refer to intentional malicious activity, whereas anomalies are deviations from normal system behaviour that may be caused by a threat. While threat detection focuses on identifying attack patterns or signatures, anomaly detection seeks to uncover unusual behaviours that may indicate a potential threat.

The early history of anomaly and threat detection dates back to the 1970s, when James P. Anderson published a paper entitled *USAF Computer Security Requirements*. This paper was part of a security requirement planning study for USAF systems and described the computer security issues that were emerging at the time.

The study included an analysis of penetration threats and the effectiveness of current technologies in countering such threats. The study identified trends and issues in the use of computers by the US Air Force that would have a significant impact on computer security. These early efforts laid the groundwork for developing new anomaly and threat detection techniques that are fundamental to modern cybersecurity.

It was not until the early 1980s that James P. Anderson published another study entitled *Computer Security Threat Monitoring and Surveillance*, one of the first to systematically address computer security monitoring and auditing. This paper provides a foundation for modern Intrusion Detection Systems (IDS). By defining threats and vulnerabilities and providing a monitoring system that continuously scans systems for suspicious activity and alerts system administra-

[a]Corresponding author, alara@uma.es
[b]mcgago@uma.es
[c]onieva@uma.es

tors, James P. Anderson's work on this aspect of computer security is considered a milestone.

Later on, Dorothy E. Denning published *An Intrusion Detection Model* in 1983. Dorothy, an award-winning American researcher, developed the first real-time IDS model called the Intrusion Detection Expert System (IDES), a simple rule-based expert system to detect suspicious activity. Throughout the 1990s and early 2000s, the field shifted toward data-driven approaches, leveraging traditional algorithms to reduce reliance on handcrafted rules [1]. The 2010s further accelerated this evolution with the advent of Deep Learning (DL), which enabled automatic feature extraction from high-dimensional traffic through neural networks [2].

In the last decade, IDS systems have been combined with AI components, which has significantly increased their effectiveness. The importance of this integration lies in AI's ability to improve the accuracy and speed of threat detection, reduce the number of false positives, and adapt to new and evolving threats through machine learning [3]. However, current challenges and gaps include ensuring the robustness and security of AI models, efficiently handling large amounts of data, and addressing potential issues [4].

To contextualise the technological evolution that underpins contemporary intrusion detection research, we summarise the major historical milestones that have shaped anomaly and threat detection over the past five decades. Figure 1 provides am overview of this progression, highlighting the transition from early monitoring concepts to modern AI-driven systems capable of addressing high-dimensional data, adversarial threats, and evolving network behaviours.

In response to these challenges and the fragmented state of the literature, the main objective of this study is to bridge the gap by providing a deep analytical exploration of the ecosystem of AI-driven anomaly and threat detection in network analysis. Rather than presenting a broad survey, we focused on generating actionable insights from experiments conducted on widely used public cybersecurity datasets. Our methodology involves standardising pre-processing pipelines and applying a range of Machine Learning (ML), Deep Learning (DL), and hybrid approaches to understand their performance under different types of attacks and data conditions.

In addition to benchmarking detection capabilities, this work addresses persistent challenges in the field, such as handling unbalanced datasets, managing high-dimensional data, adapting to concept and feature drift, and defending against adversarial machine learning threats. Beyond performance comparisons, our analysis is explicitly oriented towards maintainable AI-driven detection, in the sense of systems that can be sustained, verified, and adapted over time under real-world operational constraints.

The paper is organised as follows: Section 2 presents foundational and recent related work, identifies gaps in previous surveys, and contextualises our analytical approach.

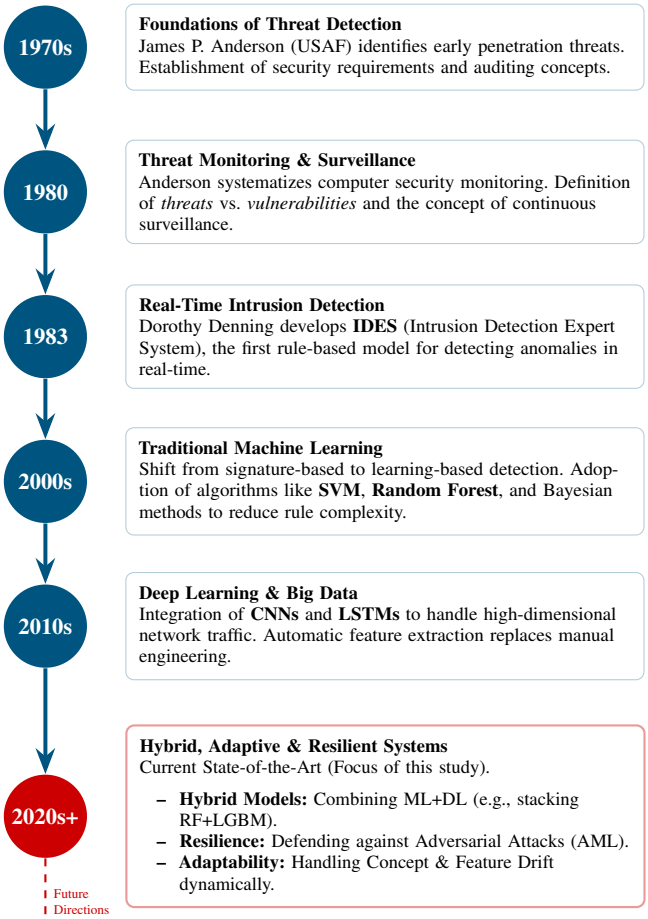

Fig. 1 Evolution of anomaly and threat detection from early rule-based systems to modern AI-driven adaptive approaches

Section 3 introduces the unified AI-based anomaly detection pipeline and experimental framework that guides the study. Section 4 presents the most used public datasets and evaluates their characteristics, attack diversity, and realism. Section 5 details the AI models, while Section 6 describes the preprocessing techniques, categorising them and reporting comparative results regarding their suitability for different attack scenarios. Section 7 summarises the key experimental findings, covering the effectiveness of hybrid approaches, cross-dataset generalisation, and adaptive methods for handling drift. Section 8 critically discusses the intrinsic vulnerabilities of these systems, the adversarial threat landscape, and the trade-offs between different modelling approaches. Finally, Section 9 provides a conclusion, highlighting the implications of our analysis and outlining future directions.

To improve the clarity and consistency of terminology used throughout this study, we provide a consolidated list of all abbreviations and acronyms referenced in the manuscript. This facilitates readability given the breadth of technical domains involved, including artificial intelligence, data science, adversarial machine learning, and network security. Table 1

Table 1 Summary of abbreviations referenced in this study

Abbreviation	Definition	Abbreviation	Definition
AD	Adversarial Detection	IG	Information Gain
AI	Artificial Intelligence	IoT	Internet of Things
AML	Adversarial Machine Learning	ISP	Internet Service Provider
AOA	African Buffalo Algorithm	k-NN	k-Nearest Neighbors
APT	Advanced Persistent Threat	LDA	Linear Discriminant Analysis
ASmoT	Adversarial Synthetic Minority Oversampling	LDAP	Lightweight Directory Access Protocol
BAE	Bayesian Autoencoder	LGBM	Light Gradient Boosting Machine
BER	Backward Elimination Ranking	LIME	Local Interpretable Model-agnostic Explanations
BGWO	Binary Grey Wolf Optimisation	LSTM	Long Short-Term Memory
BPNN	Backpropagation Neural Network	M-SvD	Modified Singular Value Decomposition
CCSA	Chaotic Crowd Search Algorithm	MIC	Mutual Information Classification
CIC	Canadian Institute for Cybersecurity	ML	Machine Learning
CNN	Convolutional Neural Network	MLHA	Multi-Layered Hybrid Architecture
D-SEM	Diverse-Self Ensemble Model	MLP	Multi-Layer Perceptron
DANN	Domain-Adversarial Neural Network	MSSQL	Microsoft SQL Server
DDoS	Distributed Denial-of-Service	MTL	Multi-Task Learning
DER	Deep Embedded Regularisation	NB	Naive Bayes
DFA-GPE	Dynamic Feature Aware Genetic Propagation Ensemble	NetBIOS	Network Basic Input/Output System
DL	Deep Learning	NTP	Network Time Protocol
DNS	Domain Name System	PCA	Principal Component Analysis
DoS	Denial-of-Service	PSO	Particle Swarm Optimisation
DR-DBMS	Dim. Reduction in Database Management Systems	RF	Random Forest
DT	Decision Tree	RFE	Recursive Feature Elimination
DyANN	Dynamic Artificial Neural Network	RL	Reinforcement Learning
EDA	Exploratory Data Analysis	RONI	Reject On Negative Impact
ELM	Extreme Learning Machine	S-DATE	Synthetic Data Augmentation Technique
FE	Feature Engineering	SBS	Sequential Backward Selection
FSR	Feature Selection Ranking	SDAE	Stacked Denoising Autoencoder
GA	Genetic Algorithm	SFS	Sequential Feature Selection
GAN	Generative Adversarial Network	SHAP	SHapley Additive exPlanations
GBM	Gradient Boosting Machine	SMOTE	Synthetic Minority Oversampling Technique
GOA	Grasshopper Optimization Algorithm	SNMP	Simple Network Management Protocol
GWO	Grey Wolf Optimisation	SSDP	Simple Service Discovery Protocol
HBA	Harris Hawk Optimisation	SVD	Singular Value Decomposition
HGS	Hunger Games Search	SVM	Support Vector Machine
HOE	Hierarchical Optimisation Evolutionary	TFTP	Trivial File Transfer Protocol
IAT	Inter-Arrival Time	UDP	User Datagram Protocol
IDS	Intrusion Detection System	XAI	Explainable Artificial Intelligence

summarises all abbreviations employed in the text together with their corresponding definitions.

2 Related Work

This section restructures prior surveys and review articles according to three principal dimensions: (i) AI approaches (Machine Learning, Deep Learning, Hybrid methods); (ii) Data science methodologies (preprocessing, datasets, evaluation metrics, feature engineering); and (iii) complementary dimensions (maintainability and adversarial attack taxonomies).

This organisation enables a systematic comparison of existing surveys beyond algorithm-centric perspectives, highlighting how each work contributes to the broader AI-based anomaly detection pipeline. In particular, the categorisation emphasises the specific methodological focus of each referenced study and clarifies the extent to which key aspects are addressed. Table 2 summarises the scope and coverage of the surveyed reviews across these dimensions.

2.1 AI approaches

Across the surveyed literature, the evolution of intrusion detection techniques reflects a progressive transition from rule-based and statistical systems to machine learning and, more recently, deep learning and hybrid architectures. Early surveys such as Mitchell and Chen [15] and Bhuyan et al. [8] provide a comprehensive characterisation of classical approaches based on statistical profiling, knowledge-driven rules, clustering and tree-based classifiers. These works emphasise the interpretability and computational efficiency of traditional models, particularly in environments with constrained resources or strict explainability requirements. However, they also highlight that shallow models struggle with

Table 2 Comparative categorisation of representative survey studies in intrusion and anomaly detection, organised into three groups: (A) general IDS and anomaly detection surveys, (B) AI-focused IDS surveys, and (C) specialised methodological or domain-specific surveys. The degree of coverage for each dimension is denoted as explicit (✓), partial or non-systematic (∼), or not addressed (✗).

	AI Approach			Data Science				
Study	ML	DL	Hybrid	Pre-Processing	Datasets	Evaluation Metrics	Maintainability	Adv. Taxonomy
General IDS & Anomaly Detection Surveys								
Bridges et al. (2019) [5]	✓	∼	∼	✓	✓	✓	∼	✗
Ahmed et al. (2016) [6]	✓	✗	✗	∼	✓	✓	✗	✗
Khraisat et al. (2019) [7]	✓	∼	∼	✓	✓	✓	∼	∼
Sabahi & Movaghar (2008) [1]	✓	✗	✗	✗	✓	✓	✗	✗
Bhuyan et al. (2014) [8]	✓	✗	∼	✓	✓	✓	✗	✗
Vani & Krishnamurthy (2017) [9]	✓	✗	✗	✓	✓	✓	✗	✗
AI-Focused IDS Surveys (Machine Learning, Deep Learning, Hybrid Methods)								
Drewek-Ossowicka et al. (2020) [10]	✓	✓	∼	✓	✓	✓	✗	✗
Resende & Drummond (2018) [11]	✓	✗	∼	✓	✓	✓	∼	✗
Abdulganiyu et al. (2023) [3]	✓	✓	✓	✓	✓	✓	∼	∼
Parkar & Bilimoria (2021) [12]	✓	✓	✗	✓	✓	✓	✗	✗
Liu & Lang (2019) [2]	✓	✓	∼	✓	✓	✓	∼	✗
Habeeb & Babu (2022) [13]	✓	✓	✗	✓	✓	✓	✗	✗
Badawy (2025) [4]	✓	✓	✓	✓	✓	✓	∼	✓
Lokhandwala (2025) [14]	✓	✓	✓	✗	✓	✓	✗	✗
Specialised Methodological and Domain-Specific Surveys								
Mitchell & Chen (2014) [15]	✓	✗	✗	✓	✓	✓	✗	✗
Nisioti et al. (2018) [16]	✓	∼	✓	✓	✓	✓	✓	✓
Thakkar & Lohiya (2021) [17]	✓	✓	✓	✓	✓	✓	∼	∼
Our study (2025)	✓	✓	✓	✓	✓	✓	✓	✓

increasingly sophisticated and rapidly evolving attacks, especially when attack behaviours diverge from the assumptions embedded in the model design. Ahmed et al. [6] reinforce this limitation by analysing the dependency of classical anomaly detection on accurate modelling of normal traffic, which may result in instability under fluctuating network conditions.

The subsequent generation of surveys reflects the impact of deep learning and high-dimensional representation learning. Drewek-Ossowicka et al. [10] conduct an extensive review of neural networks for IDS and illustrate how advanced architectures and recurrent models can extract hierarchical and temporal features from traffic data without manual feature engineering. Liu and Lang [2] expand on this by providing a taxonomy centred on data sources and demonstrating how deep learning techniques address limitations in false alarm rates, generalisation capabilities and robustness to novel attacks. Habeeb and Babu [13] further show that deep learning-based IDS have achieved significant gains in accuracy but continue to face challenges related to training cost, model complexity and the risk of overfitting when applied to limited or outdated datasets.

Several studies highlight the maturity and ongoing relevance of ensemble and tree-based methods. Resende and Drummond [11] analyse Random Forest variants and report competitive performance for feature ranking, dimensionality reduction and classification, particularly in high-dimensional scenarios. Similarly, Abdulganiyu et al. [3] and Thakkar and Lohiya [17] document the emergence of hybrid AI approaches where ML and DL components are integrated. These hybrid methods often combine the interpretability and efficiency of ML (such as Random Forest or SVM) with the representational power of neural networks. Parkar and Bilimoria [12] and Vani and Krishnamurthy [9] also show how ML-based IDSs remain essential in contexts involving big data analytics or real-time constraints where deep models may be computationally prohibitive.

More recent surveys, such as Lokhandwala [14], confirm the consolidation of deep learning and hybrid AI approaches as dominant paradigms in contemporary IDS research. While this work provides a comprehensive overview of modern ML and DL architectures and their reported performance across benchmark datasets, it largely abstracts away data preprocessing pipelines and omits any discussion on model maintainability, concept drift or adversarial resilience.

Taken collectively, these studies reveal that while classical ML approaches remain competitive for structured or resource-limited environments, deep learning and hybrid models now dominate the state-of-the-art due to their ability to automatically extract complex behavioural patterns and adapt to dynamic attack landscapes. The literature, however, consistently identifies model transparency, computational cost and dataset dependency as unresolved challenges.

2.2 Data science methodologies

Across all reviewed studies, data handling remains a central determinant of IDS performance and generalisability. Resende and Drummond [11], Bhuyan et al. [8], Ahmed et al. [6] and Abdulganiyu et al. [3] converge in emphasising that preprocessing pipelines are not peripheral tasks but foundational elements that influence every stage of IDS evaluation. These studies identify key operations such as cleaning, normalisation, encoding of categorical variables, sampling strategies to address imbalance and feature selection or reduction as essential components of any effective intrusion detection workflow.

Thakkar and Lohiya [17] provide one of the most extensive mappings of preprocessing strategies and show that the lack of standardised pipelines results in poor comparability across studies. Liu and Lang [2] reinforce this issue by noting that data source characteristics strongly influence model selection; packet-level, flow-level, session-level and log-based data require different representations and learning strategies. Studies such as Sabahi and Movaghar [1] and Mitchell and Chen [15] describe how shifts in traffic distribution, protocol structures or host behaviours can substantially degrade detection accuracy when preprocessing is not aligned with the data source.

Despite the attention given to individual preprocessing components, no previous survey establishes a harmonised cross-dataset evaluation framework with standardised pipelines and a controlled experimental setup. The absence of such a framework hinders direct comparison between ML, DL and hybrid models and limits the reproducibility of performance claims. This gap is consistently acknowledged in the literature and directly motivates the design of the present study.

2.3 Maintainability and adversarial taxonomies

A consistent weakness across related surveys is the limited consideration of model lifecycle management, long-term maintainability and resilience to adversarial manipulation. While most reviews focus on categorising algorithms, datasets and evaluation techniques, only a minority of the surveyed works engage with the practical issue that IDS models face evolving data distributions, emerging attack patterns and adversarial perturbations.

Khraisat et al. [7] recognise maintainability as a challenge but treat it descriptively rather than analytically, observing that existing IDS solutions often deteriorate over time when exposed to concept drift or outdated training data. Ahmed et al. [6] highlight the sensitivity of statistical and ML-based anomaly detection systems to distributional changes but do not integrate lifecycle management into their evaluation criteria. Nisioti et al. [16] address unsupervised learning for attribution yet do not discuss how attackers can exploit model assumptions or how model drift affects long-term reliability.

Very few surveys address adversarial machine learning. Khraisat et al. [7] mention adversarial threats but do not classify them nor analyse their implications for IDS robustness. Abdulganiyu et al. [3] acknowledge the risk of DL models being susceptible to adversarial perturbations and the need for resilient training strategies, but the discussion remains high-level. No surveyed work provides a taxonomy of adversarial threats, nor evaluates how such attacks compromise IDS performance or sustainability.

A notable exception is the recent survey by Badawy [4], which explicitly addresses adversarial machine learning in the context of intrusion detection. This work provides a structured discussion of evasion strategies, poisoning attacks and robustness-oriented training mechanisms, highlighting how adversarial manipulation can systematically undermine IDS performance. Nevertheless, although adversarial threats are categorised, the study does not integrate these considerations into a broader model lifecycle or drift-aware evaluation framework, leaving maintainability and long-term adaptation largely unaddressed.

2.4 Positioning of this work

Previous surveys provide extensive accounts of ML, DL and hybrid intrusion detection methods, as well as detailed taxonomies of datasets, attack types and evaluation metrics. However, the coverage is fragmented. Existing studies typically address isolated aspects such as feature selection, algorithmic taxonomies or dataset analysis without combining these components into a unified analytical framework. Furthermore, the lack of standardised preprocessing pipelines, the limited treatment of maintainability and the absence of adversarial threat modelling constrain the applicability of prior work to contemporary cybersecurity contexts. Even recent surveys continue to emphasise algorithmic advancements while overlooking harmonised preprocessing, lifecycle management and adversarial resilience, reinforcing the need for the unified framework proposed in this study.

The present study addresses these gaps by integrating AI approaches, data science methodologies, model lifecycle considerations and adversarial perspectives into a single comparative framework. By applying harmonised preprocessing pipelines and evaluating ML, DL and hybrid models across several benchmark datasets, this work provides a consistent cross-dataset analysis that is missing in previous surveys. Additionally, by incorporating maintainability criteria and addressing adversarial vulnerability and drift-related degradation, the study extends beyond traditional performance benchmarking and contributes new insights into the construction of robust, adaptable and verifiable IDS solutions.

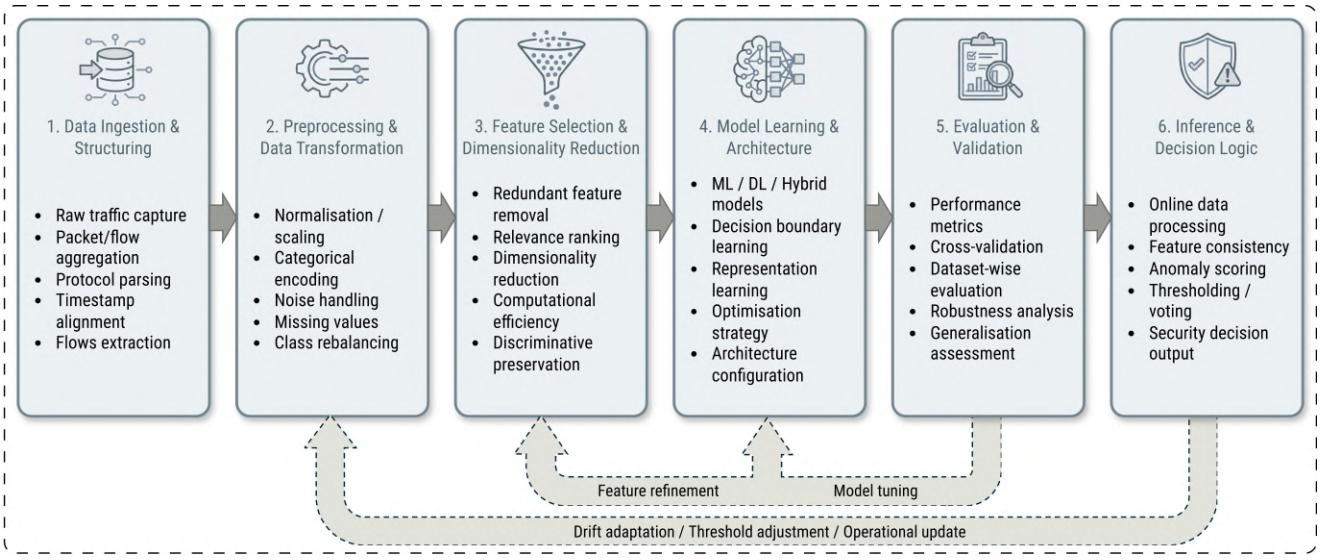

Fig. 2 Conceptual architecture of the AI-based anomaly detection pipeline adopted in this study. The upper layer presents the main processing stages, while the lower layer summarises the functional objectives associated with each stage, providing a structured reference for the datasets, preprocessing strategies, model learning process, and experimental evaluation discussed throughout the paper.

3 AI-Based Anomaly Detection Pipeline and Experimental Framework

AI-based Anomaly Detection (AD) systems are not defined solely by the learning algorithm they employ, but by the complete operational pipeline that governs how data are collected, transformed, analysed, and ultimately translated into security decisions. While existing survey literature often focuses on individual algorithms in isolation, practical IDS rely on a multi-stage and coupled workflow. To address this gap, this study explicitly adopts a standardised AD pipeline that serves as a unifying framework for datasets, preprocessing techniques and models analysed throughout the article.

The pipeline, illustrated in Figure 2 and through the organisation of Sections 4 to 7, consists of the following stages.

Data ingestion and structuring. Raw network traffic, packet traces, or host-level logs are collected and converted into structured representations suitable for ML algorithms. In the evaluated datasets, this stage includes flow aggregation, protocol parsing, timestamp alignment, and feature extraction using tools like CICFlowMeter. The resulting flow-level feature vectors form the common input space across datasets, enabling cross-dataset comparability, as detailed in Section 4.

Preprocessing and data transformation. Prior to model training, raw features undergo preprocessing operations tailored to dataset characteristics. These include normalisation, categorical encoding, noise handling, and class rebalancing. As shown later in Section 6, these steps play a critical role in mitigating class imbalance, stabilising model convergence, and improving generalisation, particularly for deep learning and hybrid architectures.

Feature selection and dimensionality reduction. Given the high dimensionality of modern intrusion detection datasets, redundant or weakly informative features can degrade both performance and efficiency. Filter-based, wrapper-based, embedded, and hybrid feature engineering techniques are applied within this stage to reduce dimensionality while preserving discriminative power. The impact of these strategies is systematically analysed in Section 6 and reflected in the experimental results of Section 7.

Model learning and architecture. At the core of the pipeline, AI models learn decision boundaries or anomaly representations from preprocessed data. This study considers traditional machine learning models, deep learning architectures, and hybrid compositions that combine both paradigms. Each model family follows its corresponding optimisation strategy, such as impurity minimisation, margin maximisation, or gradient-based learning. A taxonomy and architectural analysis of these models is presented in Section 5.

Evaluation and validation. Model performance is assessed using a consistent evaluation protocol across all datasets and configurations. Metrics such as accuracy, F1-score, recall, and class-specific performance are used to quantify detection capability, while cross-dataset and merged-dataset experiments are employed to evaluate robustness and generalisation. These results form the basis of the comparative analysis presented in Section 7.

Inference and decision logic. In deployment scenarios, incoming traffic is processed through the same preprocessing and feature engineering stages as during training, ensuring consistency between learning and inference. Model outputs are translated into anomaly scores or class predictions, which

may be combined through thresholding, voting mechanisms, or ensemble fusion. This stage reflects the operational constraints of real-world IDS deployments and motivates the analysis of adaptability, drift handling, and adversarial robustness discussed in Section 7. In recent studies, post-hoc Explainable AI (XAI) techniques such as SHAP and LIME have been applied to anomaly detection models to provide feature-level explanations of alerts. However, these methods are typically decoupled from the learning process and are not yet systematically integrated into IDS decision-making workflows [18].

By explicitly framing all experiments within this unified pipeline, the study establishes a clear architectural link between datasets, preprocessing techniques, AI models, and evaluation results. This process-oriented perspective enables a more coherent interpretation of experimental findings and addresses the need for transparent and reproducible AI-based anomaly detection systems.

All experiments conducted in this study adhere to a unified experimental design derived from the anomaly detection pipeline described above. For each dataset, identical preprocessing strategies and feature engineering configurations are applied prior to model training, ensuring fair comparison across AI techniques.

Models are trained and evaluated under both individual-dataset and multi-dataset settings to assess robustness, scalability, and generalisation. Cross-validation and stratified sampling are employed where applicable, and performance metrics are reported consistently across experiments. This controlled experimental framework allows the impact of datasets, preprocessing techniques, and model architectures to be analysed independently and jointly.

4 Benchmark Datasets for AI-Driven Intrusion Detection

AI-based intrusion detection models require extensive and representative network datasets to evaluate their performance and generalisation. Over the last decade, several benchmark datasets have been developed in this domain, each differing in terms of attack scenarios, data volume, and collection methodology. While some prioritise variation in the types of malicious traffic, others focus on the depth of specific attacks or reflect more realistic network conditions. This section reviews the major publicly available datasets, discussing not only their content and typical use cases, but also the key strengths and limitations that researchers and practitioners must consider when selecting a dataset for experimentation.

The most frequently used network datasets in detection research are those developed by the Canadian Institute for Cybersecurity (CIC), commonly referred to as CICIDS datasets. CICIDS2017 [19] captures a university-based network over five days, featuring attacks such as Brute Force, Distributed Denial-of-Service (DDoS), and web-based threats; its comprehensive set of labelled flows is excellent for benchmarking supervised models that must handle real-world traffic patterns in enterprise or academic contexts.

Building upon this, CICIDS2018 [19] extends the collection to 17 days within an AWS-based environment based on an attacker infrastructure versus a victim organisation, offering a broader array of attacks and proving valuable for evaluating robustness across longer time windows and cloud-oriented scenarios.

Meanwhile, CICDoS2019 [20] zeroes in on DDoS vectors in a controlled laboratory environment, facilitating DDoS-specific detection and mitigation studies, especially for data centres or Internet Service Provider (ISP) networks.

Another widely adopted dataset, UNSW-NB15 [21], stands out by mixing real network captures with synthetic traffic to reflect modern behaviours more accurately. In particular, this paper employs the CIC UNSW-NB15 variant [22], where features are re-extracted via CICFlowMeter for consistency with the other CIC-based datasets, resulting in a more standard flow-based dataset suited for cross-dataset experiments and mixed-traffic research.

In addition to the commonly used datasets, other datasets have been developed to address specific research needs or to incorporate new types of attacks not covered in standard datasets. Furthermore, some authors have created entirely new datasets derived from existing datasets or collected from real network environments to provide fresh perspectives and challenges for intrusion detection models. These datasets include novel attack patterns, advanced persistent threats, and data from emerging technologies like IoT.

In Table 3, we present a comparison among these datasets. This comparison highlights key features like the number of records, types of attacks included, time span of data collection, and the variety of features provided. By examining these characteristics, researchers can better align their dataset choice with their specific research objectives, ensuring that their models are trained and tested in environments that resemble the conditions under which they will be deployed.

After introducing the core datasets under study, we proceeded to extract multiple descriptive and comparative metrics to better understand their characteristics. In particular, we computed the Mutual Information Classification (MIC) values for each feature shared across these datasets, as can be seen in Figure 3.

MIC measures the degree of dependency between each feature X_i and the target class Y, providing insight into which attributes are most discriminative when detecting attacks. This is computed as:

$$I(X_i;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\left(\frac{p(x,y)}{p(x)\,p(y)}\right) \tag{1}$$

Table 3 Overview of key publicly available security datasets employed in intrusion detection research

Dataset	CICIDS2017	CICIDS2018	CIC UNSW-NB15	CICDDoS2019
Year	2017	2018	2023	2019
Developer	CIC	CIC	CIC + UNSW	CIC
Use Case	University network with different subnets and mixed real traffic	AWS-based environment (attacking infrastructure + victim organisation)	Augmented version of UNSW-NB15 with new flow-based features from CICFlowMeter	Dataset focusing on a new taxonomy of DDoS attacks
Location	Canada	Canada	Australia/Canada	Canada
Extraction Method	CICFlowMeter	CICFlowMeter	CICFlowMeter + re-labelling	CICFlowMeter
Format	CSV, PCAP	CSV, PCAP	CSV	CSV, PCAP
# Features	79	80	80	80
Data Volume	51.1 GB	220 GB	1.8 GB	6.3 GB
Duration	5 days	17 days	2 days	2 days
# Attacks	8	7	9	12
Attacks	Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, DDoS	Brute Force, Heartbleed, Botnet, DoS, DDoS, Web attacks, Infiltration	Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, Worms	DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP, SYN, TFTP, UDP, UDP-Lag, WebDDoS
Comments	Includes labelled data; is widely used for benchmarking IDS	Provides a broader range of attack scenarios than CICIDS2017	Created by matching CICFlowMeter flows with the original UNSW-NB15	In particular, it focuses on DDoS attack patterns

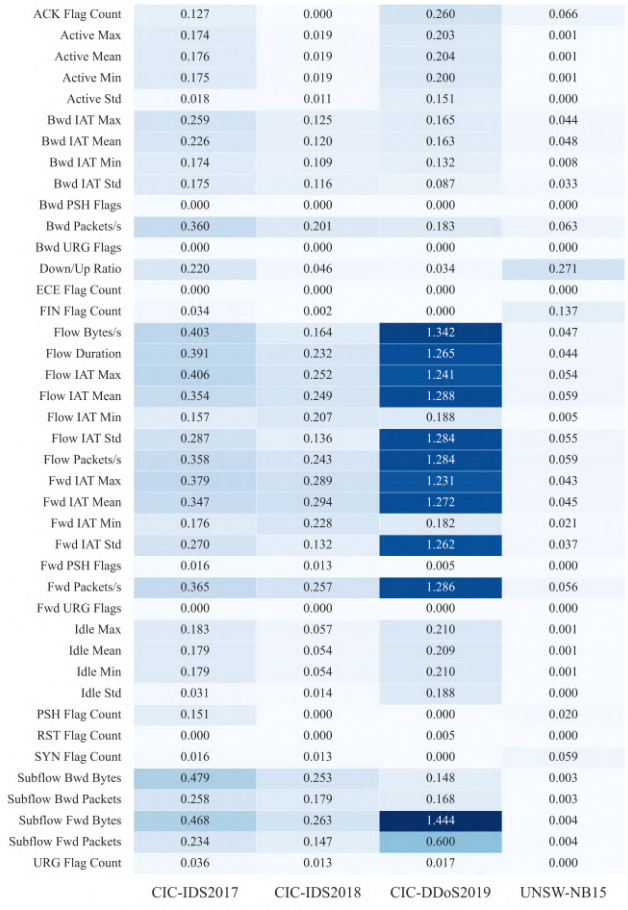

Fig. 3 MIC-based heatmap showing feature relationships across the studied datasets

However, instead of computing these distributions directly, it uses kNN-based entropy estimation from the Kraskov et al. (2004) [23] method. For every feature X_i the score $I(X_i;Y)$ measures the average reduction in class uncertainty once that feature is observed. A value of zero implies statistical independence, while larger values show that the feature carries more information about the label.

A clear concentration of high MIC values was observed in the CIC-DDoS2019 dataset, where temporal and traffic intensity features, such as Flow Bytes/s, Flow Duration, Fwd Inter-Arrival Time (IAT) Mean, and Subflow Fwd Bytes, exhibit particularly strong class relevance (values above 1.2), highlighting their role in identifying DDoS-specific behaviours. In contrast, CIC-IDS2017 demonstrated a more balanced distribution, with moderate MIC values (0.3-0.4) spread across multiple features like Flow Duration, Flow IAT Max, and Subflow Bwd Bytes, supporting the dataset diverse attack representation. UNSW-NB15, however, demonstrated low MIC scores across nearly all features, proving a weaker feature-label association and highlighting the need for stronger feature engineering or dimensionality reduction in this case. Note that Subflow Fwd Bytes consistently ranked high across all datasets, which demonstrates its potential as a universally informative feature. Thus, this heatmap guides feature selection, indicating that feature relevance is not only dataset-dependent but also highly biased in datasets dominated by specific attack types such as CIC-DDoS2019.

Alongside this feature-level analysis, we also performed an exploratory data analysis to assess the attack distribution in each dataset by capturing the frequency with which each threat type appeared and whether a significant class imbalance existed. Table 4 consolidates these findings by

Table 4 Overview of attack types on the four evaluated datasets. Each cell displays the count of occurrences and the corresponding percentage. A dash (-) indicates that the attack type is not present in that particular dataset

Attack Type	CICIDS2017	CICIDS2018	CICDDoS2019	UNSW-NB15	Total
BENIGN	2.272.895 (11.83%)	13.445.498 (70.01%)	96.546 (0.50%)	3.390.752 (17.65%)	19.205.691
TFTP	–	–	4.578.741 (100.00%)	–	4.578.741
DDoS	128.027 (9.32%)	1.246.366 (90.68%)	–	–	1.374.393
UDP	–	–	1.343.167 (100.00%)	–	1.343.167
DrDoS_UDP	–	–	1.127.468 (100.00%)	–	1.127.468
DrDoS_NTP	–	–	1.116.430 (100.00%)	–	1.116.430
DrDoS_SSDP	–	–	925.675 (100.00%)	–	925.675
DoS	252.660 (33.11%)	506.075 (66.31%)	–	4.467 (0.59%)	763.202
Syn	–	–	689.387 (100.00%)	–	689.387
MSSQL	–	–	290.470 (100.00%)	–	290.470
Bot	1.966 (0.69%)	282.310 (99.31%)	–	–	284.276
DrDoS_MSSQL	–	–	241.763 (100.00%)	–	241.763
Brute Force	13.835 (8.07%)	157.515 (91.93%)	–	–	171.350
Infiltration	36 (0.02%)	161.897 (99.98%)	–	–	161.933
Port Scan	158.930 (100.00%)	–	–	–	158.930
DrDoS_SNMP	–	–	129.225 (100.00%)	–	129.225
DrDoS_DNS	–	–	127.986 (100.00%)	–	127.986
UDP-lag	–	–	90.542 (100.00%)	–	90.542
DrDoS_LDAP	–	–	35.263 (100.00%)	–	35.263
Exploits	–	–	–	30.951 (100.00%)	30.951
Fuzzers	–	–	–	29.613 (100.00%)	29.613
DrDoS_NetBIOS	–	–	23.732 (100.00%)	–	23.732
LDAP	–	–	19.657 (100.00%)	–	19.657
Reconnaissance	–	–	–	16.735 (100.00%)	16.735
NetBIOS	–	–	11.428 (100.00%)	–	11.428
Generic	–	–	–	4.632 (100.00%)	4.632
Web Attack	2.180 (100.00%)	–	–	–	2.180
Shellcode	–	–	–	2.102 (100.00%)	2.102
Portmap	–	–	1.743 (100.00%)	–	1.743
UDPLag	–	–	455 (100.00%)	–	455
Backdoor	–	–	–	452 (100.00%)	452
WebDDoS	–	–	417 (100.00%)	–	417
Analysis	–	–	–	385 (100.00%)	385
Worms	–	–	–	246 (100.00%)	246
Injection	–	87 (100.00%)	–	–	87
Heartbleed	11 (100.00%)	–	–	–	11
Total	**2.830.540 (8.58%)**	**15.799.748 (47.93%)**	**10.850.005 (32.91%)**	**3.480.335 (10.58%)**	**32.960.628 (100.00%)**

listing all attacks present across the four datasets, providing a clear overview of their respective coverage and enabling researchers to select the most suitable dataset for a given experimental context.

5 AI Models for Anomaly Detection

In this section, we first examine a typical anomaly detection pipeline, and how models are integrated within the process. Next, we review the AI-based detection algorithms and preprocessing techniques selected for our experiments, situating them within the broader literature on ML and DL approaches for intrusion detection. Then, we apply these methods to each benchmark dataset introduced in the previous section.

ML algorithms analyse patterns in network data to classify traffic, and DL techniques excel at identifying complex patterns in large datasets. By combining these approaches, hybrid methods can enhance both accuracy and adaptability, thereby enabling more effective detection of both known and unknown threats.

Among hybrid approaches, Multi-Layered Hybrid Architectures (MLHA) have gained prominence. These models integrate ML techniques for structured and tabular data with DL capabilities for sequential and spatial pattern recognition. By leveraging the strengths of both, MLHAs can effectively process high-dimensional data in modern cybersecurity scenarios, ensuring adaptability and resilience.

5.1 Categorisation of AI Detection Models

Understanding the architecture and design principles of each model is essential for evaluating their suitability to diverse cybersecurity contexts. Although traditional models are typically known to be lightweight and interpretable, they may fail when addressing complex or novel threats. DL models excel at identifying sophisticated patterns; however, they require significant computational resources.

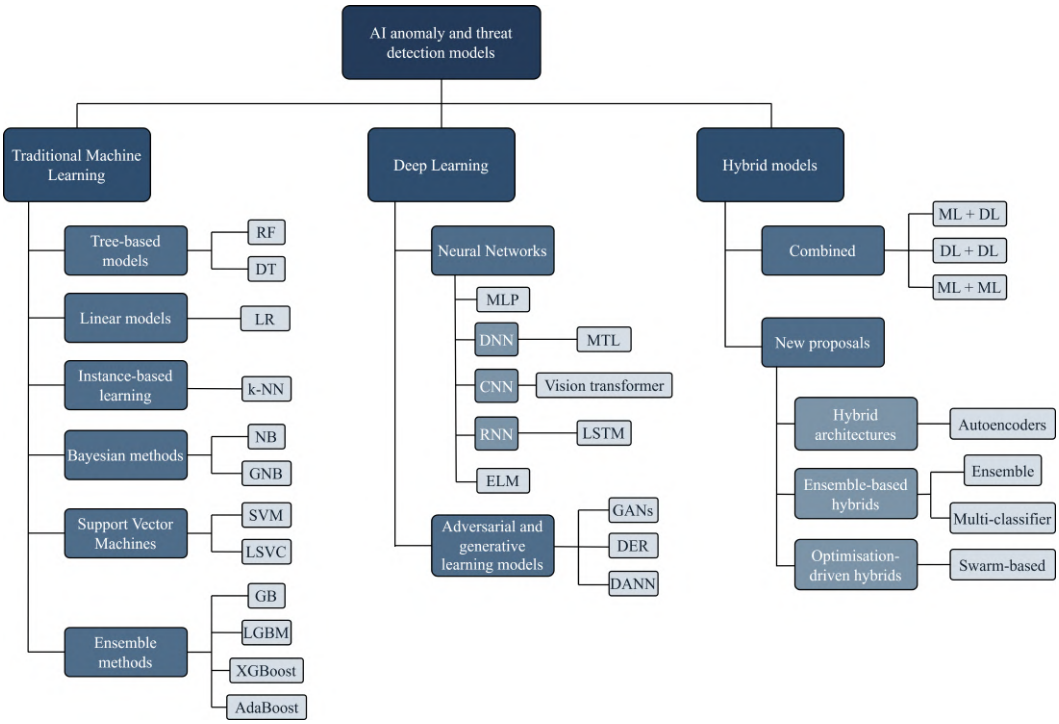

Fig. 4 Categorisation of the AI detection models examined in this study

The categorisation depicted in Figure 4 provides a structured overview of AI detection models, which are classified into three main categories, namely Traditional Machine Learning, Deep Learning, and Hybrid Models.

Traditional ML approaches remain a cornerstone of anomaly detection because of their relative simplicity and interpretability. Among these, tree-based methods, including Random Forest (RF) [24, 25] and Decision Trees (DT) [26–31] stand out for their robustness, feature-importance insights, and strong performance.

Beyond tree-based approaches, linear models (e.g., Logistic Regression [25, 30, 31]) rely on simple decision boundaries that are easy to interpret and quick to train, although they can struggle with nonlinear or multistage attacks. Instance-based methods like k-Nearest Neighbors (k-NN) [26, 28–30, 32, 33] outperform in controlled scenarios; however, they are costly for large-scale applications due to high inference-time complexity. Bayesian classifiers (Naive Bayes, Gaussian NB [26, 29–31]) offer relatively quick predictions through probabilistic assumptions; however, they can misclassify when features are highly correlated. Support Vector Machines (SVM) [25, 26, 28–30, 32, 34] excel at handling high-dimensional data; however, they require substantial computational resources and should not be considered for real-time intrusion detection. Finally, ensemble techniques like Gradient Boosting (GB), Light Gradient Boosting Machine (LGBM) [29, 30, 35] and eXtreme Gradient Boosting (XGBoost)/AdaBoost [29, 35] often achieve top-tier accuracy and scalability; however, they involve lengthy training processes, making them also more suitable for offline or batch detection.

In contrast to traditional ML, DL models learn feature representations automatically from raw or partially processed data, making them well-suited to unstructured and evolving attack patterns [30]. Common architectures include Multi-Layer Perceptron (MLP) [27, 29], often combined with feature selection to handle high-dimensional inputs, and Long Short-Term Memory (LSTM) networks [32, 36], which excel at capturing sequential dependencies in APTs. More specialised techniques like Multi-Task Learning (MTL) [37], Vision Transformer [38], and Extreme Learning Machine (ELM) [39–41] address multi-class detection, image-based network flows, and real-time training speed, respectively, making them valuable for contexts ranging from multivector attacks to low-latency scenarios.

Some DL methods generate or adapt data distributions for improved detection. Generative Adversarial Networks (GANs) [24] create synthetic traffic to enrich minority classes (e.g., zero-day or insider threats); however, training stability remains a challenge [42]. Other techniques such as Deep Embedded Regularisation (DER) [43] enhance generalisation against complex threats like polymorphic malware, and Domain-Adversarial Neural Networks (DANN) [37] facilitate knowledge transfer across different data domains, which demonstrates adaptability when bridging synthetic and real-world scenarios.

In parallel, Reinforcement Learning (RL)–based approaches have been increasingly adopted for zero-day and

previously unseen attack scenarios, particularly in dynamic environments where labelled data are unavailable [44]. By learning detection or mitigation policies through reward-driven interaction with the environment, RL enables IDS to adapt their behaviour online without relying on predefined attack signatures.

Hybrid model approaches combine the strengths of traditional ML and deep learning, aiming for robust anomaly detection across diverse scenarios. By leveraging the interpretability and efficiency of classical methods while taking advantage of neural networks, hybrid architectures typically offer improved accuracy and greater adaptability. Many of these solutions appear as new proposals because researchers find it straightforward to experiment by chaining or nesting different algorithms.

Some authors stack or concatenate existing techniques, such as ML with ML, DL with DL, or ML with DL. A notable example is the Convolutional Neural Network (CNN)+LSTM configuration [36], which combined convolutional feature extraction with the temporal learning power of recurrent networks, outperforming standalone CNNs or LSTM in multi-vector attack detection. Similarly, RF+LSTM [45] improves the decision-tree strengths of Random Forest along with LSTM sequential insights, achieving accuracy above 99% on UNSW-NB15. There are also simpler pairings like NB+SVM [34, 46] that reduce false positives via feature selection, yielding 92-95% accuracy on NSL-KDD.

Recent literature has demonstrated more sophisticated or domain-specific hybrid systems:

1. **Ensemble Methods**. Zhou et al. [47] proposed adding a "Feature Penalisation Attribute" to RF in order to reduce overfitting, achieving 94% accuracy on UKM-IDS20.
2. **Swarm-Based Approaches**. Sheikhi et al. [48] combined Particle Swarm Optimisation (PSO) and Grey Wolf Optimisation (GWO) within a Backpropagation Neural Network (BPNN). A Genetic Algorithm (GA) was used to select relevant features, attaining 98.9% accuracy on UNSW-NB15.
3. **Hybrid Optimisation**. ElDahshan et al. [39] introduced an African Buffalo Algorithm (AOA) and Harris Hawk Optimisation (HBA) to refine the parameters of an ELM. This dual-optimisation approach yielded 99.62% accuracy in UNSW-NB15, highlighting notably low false alarm rates. Al-Daweri et al. [37] also combined Hierarchical Optimisation Evolutionary (HOE) techniques with Dynamic Artificial Neural Networks (DyANN), surpassing the baseline DyANN by 1.06%.
4. **Multi-Classifier Architectures**. Turukmane et al. [49] employed the Mud Ring Optimisation Algorithm to build a multilayer SVM pipeline, called M-MultiSVM, along with the Advanced Synthetic Minority Oversampling Technique (ASmoT) and Singular Value Decomposition (SVD). Their system achieved 99.89% accuracy on CICIDS2018 and 97.54% on UNSW-NB15, outperforming RF, SVM, and k-NN in imbalanced settings.
5. **Autoencoders**. Yang et al. [50] introduced Bayesian Autoencoders (BAE) to estimate uncertainty in rare or emerging attack scenarios, reaching 98.6% accuracy in UNSW-NB15 and 84.6% in CICIDS2017. By quantifying the predictive uncertainty, these autoencoders are better suited for low-frequency attacks and previously unseen threats.

5.2 Analysis of AI Detection Models

Table 5 presents a comparative analysis of the AI-based models for intrusion detection. Each model is evaluated across several practical criteria, including suitability for batch or real-time processing, scalability, computational efficiency, explainability, robustness to data imbalance, and adaptability to novel or evolving threats. The goal of this analysis is to provide actionable information on the strengths and limitations of each type of model, helping practitioners and researchers select the most appropriate techniques based on the deployment constraints and security requirements.

The AI detection models examined illustrate a clear trend: there is a growing preference for hybrid approaches that integrate both ML and DL methodologies. These models offer distinct advantages and highlight specific trade-offs that must be carefully balanced in practical deployments.

For example, the RF-based model augmented with a Forest-PA feature-selection layer demonstrated robust performance in environments characterised by high-dimensional, noisy data. Its strength lies in its ability to efficiently handle complexity and extract the most informative attributes. However, this power comes at a cost: every prediction must traverse hundreds of trees, so inference latency grows linearly with ensemble size. Because Random Forests are batch learners, coping with concept or feature drift means periodically rebuilding or grafting trees, interrupting the pipeline and adding further compute overhead. Consequently, while the approach achieves 94% accuracy on the UKM-IDS20 benchmark, recent surveys warn that traditional RFs are only "partially suitable" for real-time use and that "real-time detection is still a challenging question" unless the system is re-engineered with distributed or parallel inference to keep pace with high-volume streams.

In contrast, the traditional ML method based on SVM, as observed in the M-MultiSVM approach, excels at binary classification tasks due to its high precision. However, SVM performance tends to diminish when applied to larger and more complex datasets, where the clear boundaries required for accurate classification become blurred. This limitation is reflected in its dependency on well-defined class separations and can be a significant drawback in dynamic real-world environments where data complexity is the norm.

Table 5 Comparative summary of AI-based intrusion detection models. ✓= strong suitability; ∼ = suitable with limitations; ✗= not suitable.

Model	Batch Processing	Real-Time Capability	Scalability	Computational Efficiency	Explainability	Imbalance Robustness	Adaptability
RF	✓	✓	✓	∼	∼	∼	✗
DT	✓	✓	✓	✓	✓	∼	✗
XGBoost	✓	✓	✓	∼	∼	∼	✗
LR	✓	✓	✓	✓	✓	∼	✗
k-NN	✓	∼*	✗	✗	✗	✗	✗
NB	✓	✓	✓	✓	∼	✗	✗
SVM	✓	∼*	✗	✗	✗	∼	✗
LGBM	✓	✓	✓	∼	∼	∼	✗
MLP	✓	✓	∼	∼	✗	∼	✗
LSTM	✓	∼	∼	✗	✗	∼	∼
Vision Transformer	✓	✗	∼	✗	✗	∼	✗
GAN	✓	∼	∼	✗	✗	✓	✓
CNN + LSTM	✓	∼	∼	✗	✗	∼	✗
RF + LSTM	✓	∼	∼	✗	✗	∼	✗
NB + SVM	✓	∼	∼	∼	✗	∼	✗
BAE	✓	∼	∼	✗	✗	✓	✓
RF + Forest PA	✓	✓	✓	∼	∼	∼	✗
AOA/HBA-ELM	✓	✓	✓	∼	✗	∼	✗
HPSOGWO-BPNN	✓	✓	∼	✗	✗	∼	✗
HOE-DANN	✓	∼	∼	∼	✗	∼	✓
MultiSVM	✓	∼	∼	∼	✗	∼	✗

Note: *Real-time capability for k-NN and SVM is limited by computational cost on large data. Most traditional supervised models tend to bias towards the majority class under class-imbalanced datasets, reducing minority attack detection (marked as ∼ in robustness). Unsupervised methods like GAN and BAE focus on learning "normal" behaviour, making them more robust to imbalance and capable of flagging novel attacks outside the training distribution. Each symbol reflects general tendencies reported in recent literature (2020–2024) for the given model in intrusion/anomaly detection contexts.

On the deep learning front, LSTM networks have demonstrated their capability to capture temporal dependencies, an essential feature for detecting APTs that evolve over time. Their sequential processing nature makes them well suited for analysing time-series data, but this comes at the cost of higher computational requirements and potential latency issues in real-time applications. Similarly, CNNs are adept at automatically extracting complex patterns from high-dimensional data, which makes them effective for tasks involving spatial data correlation. However, CNN-based models often suffer from a lack of interpretability, which can limit their acceptance in environments in which explainability is paramount.

The table also presents several hybrid models that attempt to mitigate the limitations inherent in single-method approaches. For example, the AOA-ELM and HBA-ELM models employ Extreme Learning Machines tuned via metaheuristics (AOA and HBA, respectively), achieving impressive accuracy scores of 99.62% and 98.65% on the UNSW-NB15 dataset. These models benefit from the speed and generalisation ability of ELMs while leveraging optimisation techniques to fine-tune neuron performance, thereby striking a balance between efficiency and accuracy.

Another notable model is BAE, which integrates a variational lower bound and noise modelling. This approach not only yields high detection accuracy (98.6% on UNSW-NB15) but also incorporates a probabilistic framework that can offer a degree of uncertainty estimation, potentially aiding decision-making processes. However, the performance disparity observed on different datasets (84.6% on CIC-IDS-2017) suggests that the model's robustness may vary depending on the characteristics of the data, which underscores the need for further validation across diverse environments.

The HPSOGWO-BPNN model represents another hybrid strategy by combining PSO and GWO with a BPNN. This method capitalises on the complementary strengths of swarm intelligence techniques and neural networks, achieving an accuracy of 98.9% on the UNSW-NB15 dataset. Although promising, reliance on metaheuristic optimisation can introduce additional complexity in tuning and may require substantial computational resources during the training phase.

Finally, the HOE-DANN model, which integrates a DyANN with the HOE framework, achieved moderate accuracy of 94.66% on the UKM-IDS20 dataset. Although its performance might appear lower than that of other models, its design emphasises adaptability and dynamic response, which are crucial features in rapidly changing threat landscapes.

Overall, these hybrid models demonstrate a critical evolution in AI detection systems by combining the best aspects of ML and DL techniques. They reconcile the trade-offs between accuracy, scalability, computational efficiency, and explainability. By leveraging optimisation strategies and hybrid architectures, these models provide a more balanced

solution to address the diverse challenges posed by modern cybersecurity threats. The table serves as a comprehensive comparison that not only highlights the strengths of each approach but also underscores the inherent challenges, such as computational overhead and explainability, that remain active research areas. Future studies should focus on refining these hybrid techniques, exploring new optimisation paradigms, and validating models across varied operational contexts to enhance their robustness and practical applicability.

6 Preprocessing Techniques for Anomaly Detection

In this section, we shift our focus from model architectures to the preprocessing techniques essential for effective anomaly detection. Unlike models, which directly handle the decision-making tasks, preprocessing methods play a crucial supporting role by preparing and optimising the data before it reaches these algorithms. Proper preprocessing ensures that ML, DL, and hybrid detection methods receive clean, relevant, and structurally coherent inputs, significantly improving their predictive performance and reliability.

Specifically, preprocessing techniques address inherent challenges present in network traffic data by transforming and refining the dataset. Effective preprocessing not only enhances detection accuracy but also ensures computational efficiency and adaptability, enabling models to maintain robust performance even as data characteristics evolve over time. In the following subsections, we categorise these preprocessing approaches systematically and review their practical implementation and theoretical benefits within the broader anomaly detection pipeline.

6.1 Categorisation of Preprocessing Techniques

Raw network traffic data are typically noisy, imbalanced, and high-dimensional; thus, preprocessing is an indispensable step in data training. Preprocessing techniques were used to prepare datasets for training ML and DL models. They consist of algorithms and techniques used to ensure that the data used to train an AI model are clean, relevant, and structured in a way that maximises accuracy and enhances the overall performance of the proposed model.

For this analysis, we focused on what we called Feature Engineering (FE), which mainly consists of the techniques developed to change the format and presentation of the features to be later used as a better input. This includes reducing the complexity of high-dimensional datasets, recursive feature elimination, genetic algorithm usage, and Information Gain (IG) gain to select the most informative features [39, 47, 51].

The conceptual model of preprocessing techniques presented in Figure 5 outlines a categorisation of the methods observed in the literature. The proposed structured framework provides a foundation for analysing the strengths, limitations, and application contexts of each technique.

This taxonomy divides preprocessing approaches into key categories that can work together and are not exclusive to each other: Cleaning, Transformation, Data Resampling and Augmentation, and Feature Engineering.

Data cleaning involves removing noise, filling missing values, and resolving outliers. Datasets like CICIDS2017 often contain irrelevant or corrupted entries that must be treated before training [29, 52].

The Transformation ensures that data are in a suitable format. Scaling (e.g., MinMax, StandardScaler, Zero-Mean) improves convergence and stability in models like SVM, CNN, and LSTM, while encoding (e.g., one-hot, label) translates categorical features for ML consumption [43]. In some studies, entire datasets are reformatted, for example by converting into RGB image sequences for Vision Transformers [38].

Resampling and Augmentation mitigate class imbalance. Techniques like Synthetic Minority Oversampling Technique (SMOTE), SMOTE-Tomek, and NearMiss either oversample the minority class or undersample the majority [45, 53]. More advanced approaches like Synthetic Data Augmentation Techniques (S-DATE) synthesise new malicious instances for deep models [51], while downsampling offers a lightweight alternative for traditional ML pipelines [29].

FE is critical on high-dimensional datasets in which not all attributes contribute meaningfully to detection. The proposed FE method improves both performance and interpretability while reducing training overhead. Different techniques for Feature Engineering fall into several categories:

Filter methods such as IG, Gini Impurity, and SelectKBest select features based on statistical relevance without involving model training [36, 46, 54]. These are efficient and are commonly used to pre-trim features before applying more complex algorithms.

The embedded methods evaluate features as part of the model training process. Examples include Intersectional Correlated Feature Selection (ICFS), Chaotic Crow Search Algorithm (CCSA), and Binary Grey Wolf Optimisation (BGWO), which have shown strong improvements in detection accuracy when integrated into hybrid models such as AOA-ELM and HBA-ELM [34, 52]. These improvements occur because embedded methods select features based on their direct contribution to predictive performance during training, optimising the feature set specifically for the model at hand. By dynamically adapting feature selection within the model's learning process, embedded methods capture complex interactions, resulting in models better tuned to the nuances of the data.

Wrapper methods, like Sequential Forward Selection (SFS), Sequential Backward Selection (SBS) or Recursive Feature Elimination (RFE) are computationally expensive but provide high performance by evaluating subsets iteratively based on model output [39].

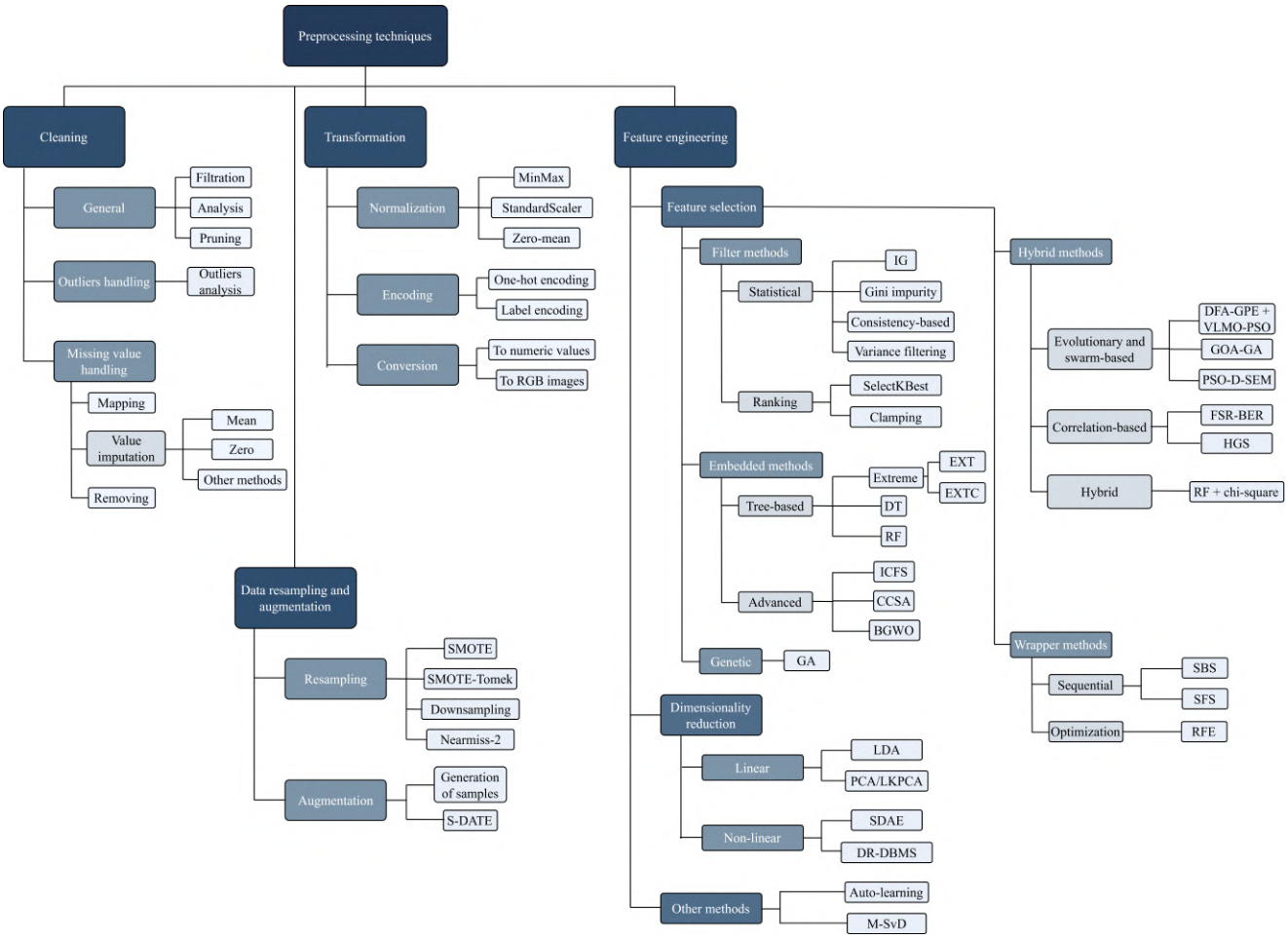

Fig. 5 Categorisation of the preprocessing techniques employed for AI detection models

Hybrid methods combine genetic, swarm, and statistical approaches. Techniques such as Grasshopper Optimization Algorithm with GA (GOA-GA), PSO-based Diverse-Self Ensemble Model (D-SEM), and Dynamic Feature Aware Genetic Propagation Ensemble (DFA-GPE) dynamically optimise feature subsets, improving generalisation and recall on minority-class threats [51, 55, 56]. Some also integrate evolutionary strategies with classifiers, like in Hybrid Hunger Games Search (HGS), or in Forward Selection Ranking (FSR) and Backward Elimination Ranking (BER); or statistical scoring like chi-square to refine selection [45, 53].

Dimensionality reduction further enhances learning, particularly in DL. Linear methods, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), preserve variance or maximise class separability by projecting data onto lower-dimensional spaces where important distinctions among features remain clear, making patterns easier to capture. In contrast, nonlinear methods like Stacked Denoising Auto Encoder (SDAE) and Dimensionality Reduction in Database Management Systems (DR-DBMS) compress data representations by learning intricate, hierarchical relationships among input features. These nonlinear approaches effectively isolate essential characteristics of the data, thus filtering out irrelevant variations or noisy inputs that might otherwise mislead models during training. As a result, such compression techniques improve model robustness, generalisation, and resilience by focusing training efforts on the most informative and stable patterns within datasets [28, 57].

Finally, automated FE techniques like Auto-Learning with CNNs [58] or M-SVD with MultiSVM [49], allow models to autonomously identify critical input features, minimising preprocessing effort and accelerating deployment.

6.2 Analysis of Preprocessing Techniques for AI Detection Models

Table 6 presents a comparative analysis of the preprocessing techniques reviewed for anomaly detection. Each technique is evaluated across practical criteria, including its capability to handle noise and data imbalance, effectiveness in di-

Table 6 Comparative summary of preprocessing techniques used for anomaly detection. ✓= strong suitability, ✗= not suitable, and ∼ = suitable with limitations.

Technique	Handles Noise	Handles Class Imbalance	Reduces Dimensionality	Computational Efficiency	Model Independence
Missing value imputation	✗	✗	✗	✓	✓
Outlier removal	✓	✗	✗	✓	✓
Data scaling (normalization, standardization)	✗	✗	✗	✓	✓
Categorical encoding (one-hot, target encoding)	✗	✗	✗	✓	✓
SMOTE	✗	✓	✗	∼	✓
SMOTE–Tomek	✓	✓	✗	∼	✓
NearMiss	✗	✓	✗	∼	✓
S-DATE (synthetic data augmentation)	∼	✓	✗	✓	✓
Information Gain (IG)	✗	✗	✓	✓	✓
Gini impurity / importance	✗	✗	✓	✓	✓
ICFS (correlation-based)	✗	✗	✓	✓	✓
CCSA (chaotic crowd search)	✗	✗	✓	∼	✗
BGWO (binary grey wolf opt.)	✗	✗	✓	✗	✗
RFE (recursive feature elimination)	✗	✗	✓	✗	✗
GOA–GA	✗	✗	✓	✗	✗
PSO–D–SEM	✗	✗	✗	✗	✗
PCA	∼	✗	✓	∼	✓
LDA	✗	✗	✓	✓	✓
SDAE (stacked denoising autoencoder)	✓	✗	✓	✗	✓
DR–DBMS (dim. reduction in DBMS)	✗	✗	✓	✓	✗
Auto-learning CNN (representation learning)	✗	✗	✓	✗	✗
Modified SVD	✗	✗	✓	✗	✓

Note: Symbols synthesise empirical evaluations from recent literature (2020–2025) on preprocessing for anomaly detection. Techniques marked as ∼ under "Handles Noise" or "Computational Efficiency" provide conditional or partial utility depending on dataset size and implementation details.

mensionality reduction, computational efficiency, and model independence. The aim of this analysis is to provide clear insights into the strengths and limitations of each preprocessing method, enabling practitioners and researchers to select the optimal strategies based on their specific data characteristics and deployment scenarios.

Feature engineering plays a crucial role in enhancing model performance, and traditional methods, such as RFE and SFS, remain widely used due to their simplicity and effectiveness in terms of reducing overfitting. These approaches are particularly beneficial for tree-based models, which often suffer feature redundancy and high dimensionality. However, their limitations are evident when handling complex datasets with intricate feature interactions. Advanced methods such as BGWO and CCSA, address these limitations by exploring feature spaces more thoroughly and preventing local optima. Although these advanced techniques provide significant improvements, they are associated with substantial computational overhead, making them impractical for real-time applications or large-scale datasets. This trade-off highlights the ongoing challenge of balancing interpretability, efficiency, and scalability in feature selection.

Normalisation techniques, including Min-Max Scaling and Z-score Normalisation, are essential for handling numerical disparities in datasets. These methods are particularly effective for models sensitive to feature magnitude, such as SVM and neural networks, where differences in scale can significantly impact convergence speed and overall performance. Despite their effectiveness, these techniques do not address class imbalance or feature drift; thus, they are insufficient when handling dynamically evolving attack patterns. While normalisation ensures a stable input distribution, adaptive

feature selection and augmentation techniques must complement it to maintain long-term robustness.

Data resampling techniques, particularly SMOTE, have become the standard for addressing imbalanced datasets, which is a common issue in cybersecurity applications where malicious traffic is significantly outnumbered by benign activity. Although SMOTE effectively improves recall and reduces false negatives, it struggles to generate realistic samples for rare attack types, such as infiltration and zero-day exploits. To overcome this limitation, hybrid approaches such as SMOTE-Tomek and NearMiss-2, combine oversampling and undersampling, which enhances the dataset balance while preserving the statistical integrity of attack distributions. However, such techniques introduce a potential drawback: by modifying the original dataset composition, they may distort real-world traffic distributions, leading to an overfitting risk when applied to live network data. For DL models, S-DATE offer a more sophisticated approach by generating synthetic samples that better mimic real-world behaviour, yet their effectiveness depends heavily on the quality of the generative model used.

7 Key Findings

In this section, we present a series of key findings derived from our experimental evaluation using publicly available datasets, AI detection models, and previously described preprocessing techniques. All experiments were conducted on a workstation equipped with an AMD Ryzen 7 5800H CPU (8 cores, 16 threads), 16 GB DDR4 RAM, and NVIDIA RTX 3060 GPU running Ubuntu 24.04 LTS. The software environment included Python 3.12.8 using key libraries such as scikit-learn (1.6.0), TensorFlow (2.18.0) with Keras (3.8.0), and NumPy (2.0.2), among others. By applying these methods across diverse data sources and attack scenarios, we identify recurring patterns, performance trends, and practical challenges. Each subsection introduces a thematic focus followed by a statement of the corresponding key finding. Together, these insights provide a deeper understanding of the factors that drive successful AI-powered intrusion detection and highlight areas for further research and optimisation.

7.1 Hybrid Models Achieve Superior Performance Across Datasets

To validate the comparative performance of the AI-based detection models, we conducted a series of controlled experiments using the benchmark datasets introduced in Section 4. For each dataset, we applied a representative set of traditional ML, DL, and MLHA models along with standard preprocessing techniques. Evaluation metrics such as accuracy, F1-score, training and inference time were used to assess model behaviour under balanced and imbalanced conditions.

Figure 6 shows the accuracy (top) and F1-score (bottom) of different AI models on benchmark datasets and preprocessing pipelines. Each row corresponds to a specific dataset combined with a preprocessing strategy, namely PCA, top-k feature selection, with or without SMOTE. The value of k was selected, iterating through different values and testing the overall accuracy in the baseline model. The lowest k value representing the local maximum is selected for that specific dataset. Here, each column represents a detection model.

The experimental results provide compelling evidence in favour of hybrid models that are the most robust and generalizable solution for intrusion detection tasks. As shown in Figure 6, the hybrid architecture (Logistic Regression on a Stacked RF + LGBM) outperformed all other models across all datasets, achieving near-perfect accuracy and F1-score values in most scenarios. This trend is consistent across preprocessing setups, including both PCA and top-k feature selection, with or without the application of SMOTE. Such consistency highlights the capacity of hybrid models to combine the strengths of traditional ML and DL architectures, yielding superior generalisation and adaptability.

However, this advantage comes with a significant drawback. Hybrid models incur considerable computational overhead, resulting in high training times. The training times for the evaluated models varied across datasets and preprocessing setups. Notably, the hybrid model consistently required substantially longer training durations compared to traditional ML models. For instance, on the CIC-DDoS2019 dataset using PCA dimensionality reduction without SMOTE, the LR[RF+LGBM] model exhibited the highest training time of approximately 1289.95 seconds, significantly exceeding simpler models such as RF (8.10 s) or LGBM (6.11 s).

Even for relatively less demanding scenarios, such as the UNSW-NB15 dataset combined with PCA and no SMOTE, the hybrid model (1210.38 s) still outperformed traditional methods like RF (6.63 s) and k-NN (2.29 s) by a wide margin. This trend underlines the trade-off of employing hybrid architectures: despite their superior accuracy and generalisation, their complexity incurs high computational costs, which may limit their applicability in environments where frequent retraining or rapid deployment is essential.

It is important to note that the reported computational overhead primarily reflects training-time costs, which are incurred offline and do not directly affect operational deployment. In practical detection scenarios, however, inference efficiency is a critical factor for real-time applicability, measured in terms of throughput and latency. To account for this aspect, we conducted an additional throughput-oriented evaluation using the same models already trained in the previous experiment, thereby avoiding any retraining bias. The evaluation was performed using the same dataset and maintaining the same execution environment, focusing exclusively on obtaining the average inference performance for each model

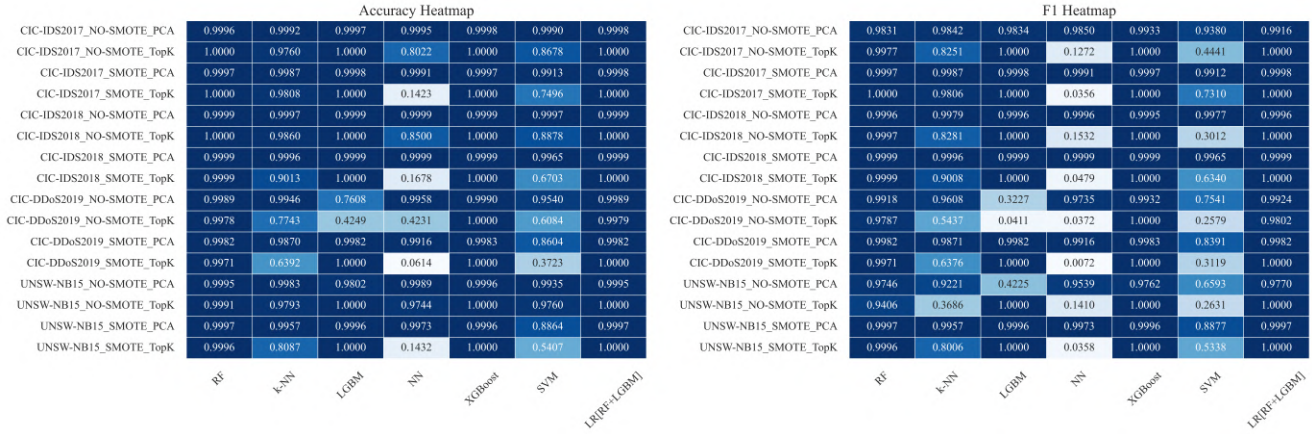

Fig. 6 Accuracy and F1-score heatmaps for models under all dataset and preprocessing conditions

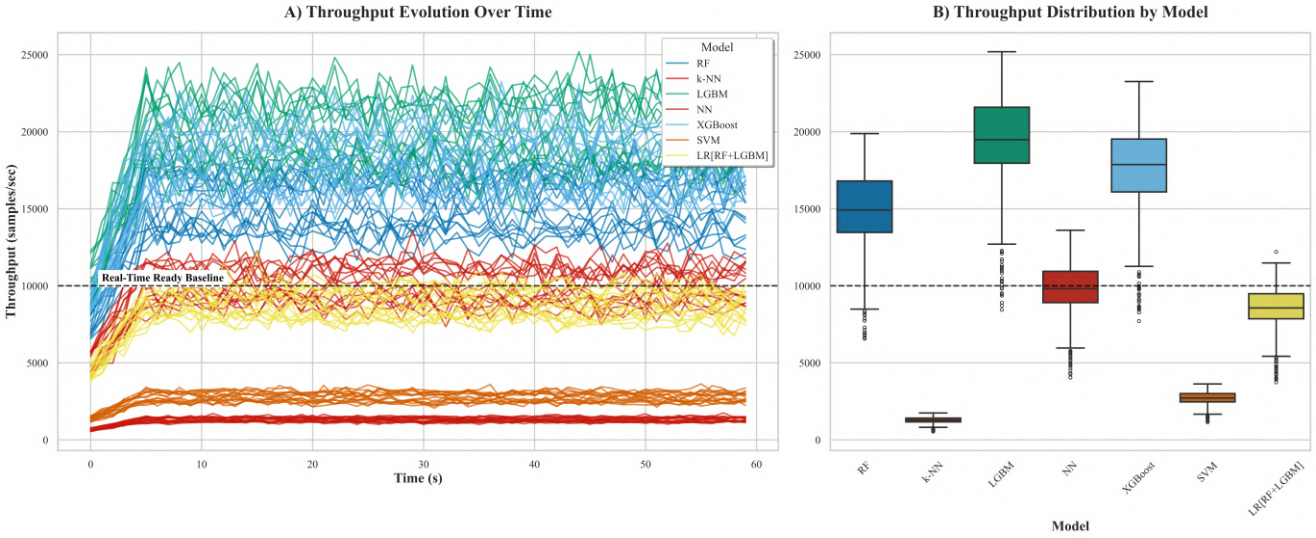

Fig. 7 Inference throughput analysis comparing all trained model configurations. (a) Temporal evolution of throughput over a 60-second test window. The dashed line at 10,000 samples/sec indicates the minimum baseline required for real-time deployment. (b) Statistical distribution of throughput by model.

across their trained configurations. The resulting throughput measurements are reported in Figure 7.

As illustrated in the Figure, the experimental results differentiate between three distinct performance tiers relative to the operational baseline of 10,000 samples/second. This baseline represents a self-established minimum throughput required to handle network flows in a standard real-time deployment without inducing packet loss or queuing delays.

The first group, comprising tree-based ensemble methods, demonstrates superior inference efficiency, consistently exceeding the baseline with peak throughputs ranging between 15,000 and 20,000 samples/second. The second group includes the Neural Network (NN) and the proposed Hybrid architecture (LR[RF+LGBM]). Notably, while the Hybrid model achieved the highest detection accuracy and F1-scores as detailed in Figure 6, its inference throughput hovers near the 10,000 samples/second threshold. This behaviour quantifies the cost of the stacking strategy: while the meta-learner significantly improves generalisation, it introduces a latency penalty compared to single-model architectures. However, unlike the third group, the Hybrid model remains operationally viable for near real-time prevention.

The third group consists of distance-based and kernel-based algorithms. Despite offering competitive accuracy in specific preprocessing configurations, their throughput falls significantly below the operational baseline ($< 3,000$ samples/second). Consequently, while these models may be valuable for offline forensic analysis where latency is not a constraint, they are not suited for live traffic analysis.

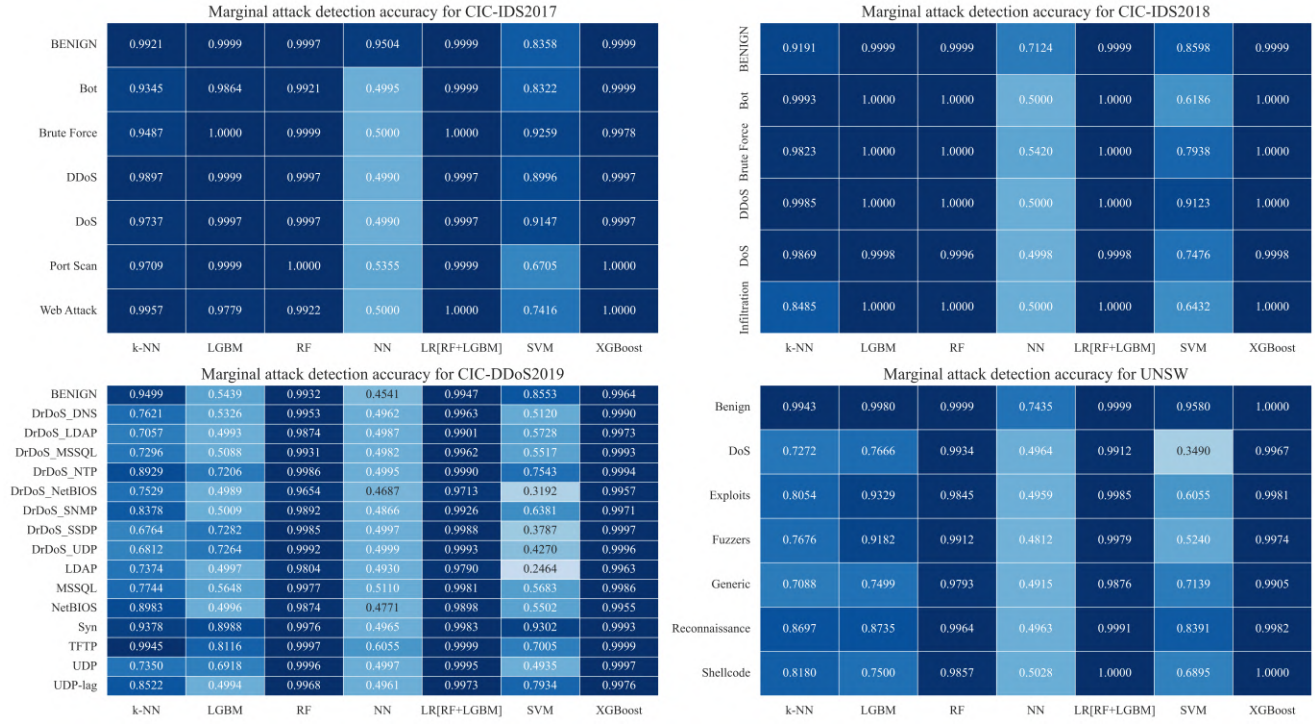

Fig. 8 Heatmaps of marginal attack detection accuracy across individual datasets

Integrating diverse modelling strategies into unified hybrid architectures provides clear advantages for AI-based intrusion detection models, particularly in imbalanced and evolving threat scenarios. By combining dimensionality reduction techniques with class imbalance handling, these architectures maximise model generalisation and robustness while maintaining inference throughput, thus validating their operational viability despite the inherent increase in computational complexity.

7.2 Detection Effectiveness Varies by Attack Type

The results obtained in the previous key finding reflect the overall model performance; however, they do not account for how effectively each model detects specific types of attacks. In real-world scenarios, intrusion detection systems must be capable of not only classifying general traffic but also of recognising several attack behaviours, some of which may exhibit rare patterns.

To explore this dimension, we conducted a more granular analysis by computing the detection accuracy per attack type for each model across the evaluated datasets. For each dataset, we selected the same models as in the last key finding and constructed individual heatmaps, where each cell shows the accuracy of a model when detecting a specific attack type for each specific dataset. This enables a comparative view of model sensitivity to different threat categories, available in Figure 8.

The results demonstrate that ensemble models achieve the highest accuracy across most attack types and datasets, especially when detecting more complex attacks. On the other hand, neural network models perform poorly likely due to their sensitivity to class imbalance and lack of robust preprocessing. The hybrid model LR[RF+LGBM] shows stable and reliable results across different attacks. Detection performance clearly depends on both the type of attack and the dataset used, which highlights the importance of using balanced data and choosing the right model for each scenario.

Detection accuracy for individual attack types is highly nonuniform. Our analysis shows that ensemble methods achieve superior performance, particularly on complex or low-frequency attacks, whereas DL models often suffer from decreased sensitivity under class imbalance. This highlights the advantage of hybrid architectures in achieving reliable, high-fidelity intrusion detection across diverse attack categories.

7.3 Multi-Dataset Training Enhances Generalisation

After discussing the performance of different models on various benchmark datasets, we focused on understanding generalisation across experiments. Our approach begins with the observation that our datasets can be seamlessly merged

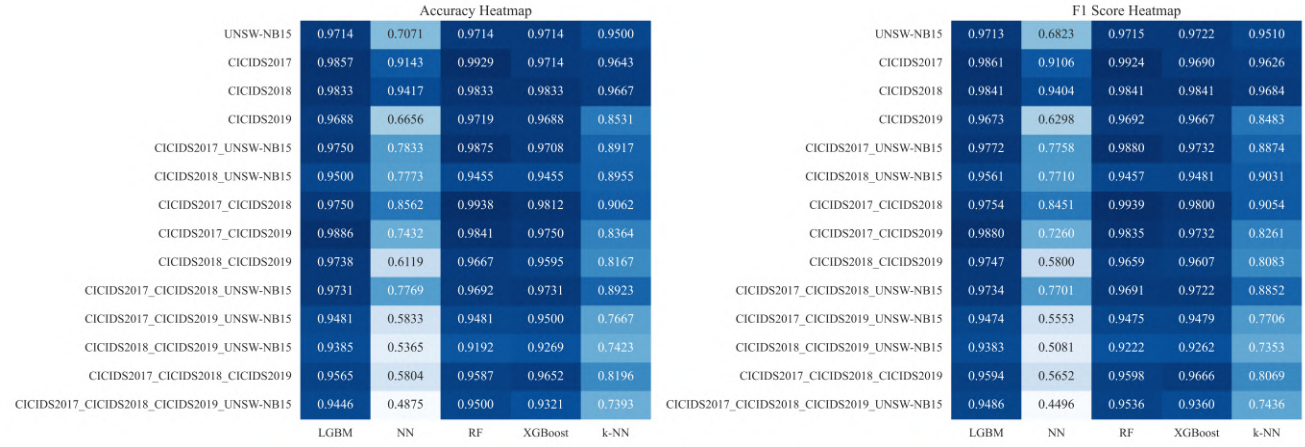

Fig. 9 Model performance metrics evaluated on all newly created datasets

by selecting common features, which allows us to create multi-dataset configurations.

In Table 7, these combinations are presented along with useful statistics such as the total number of samples, the proportion of attack instances, the distinct attack types present and the computed class entropy. This entropy is a measure of how evenly the classes are distributed in a dataset. It helps us understand how balanced or imbalanced the dataset is. A low entropy means that one class dominates, whereas a high entropy means the classes are more evenly distributed.

The formula used to calculate entropy is:

$$H = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i) \tag{2}$$

Here, p_i is the proportion of instances that belong to class i, and n is the total number of different classes. If most instances belong to one class, the entropy is close to 0. If all classes have similar proportions, the entropy is higher.

The first experiment establishes baseline models for each newly created merged dataset alongside the original datasets to serve as controls. Figure 9 shows the performance metrics obtained for each model on each newly created dataset. We can see that tree-based ensembles dominate across every merged-dataset scenario: RF, closely followed by XGBoost, deliver the highest accuracy and F1 scores because their bagging/boosting strategies smooth out the noise and heterogeneity introduced when datasets are combined. LGBM remains competitive on moderately mixed sets but loses ground once a single attack family overwhelms the class balance, hinting at over-specialisation from its leaf-wise growth. By contrast, the plain NN and, to a lesser extent, k-NN collapse when severe imbalance or feature-scale disparities appear (especially in CIC-DDoS2019-heavy blends), confirming their sensitivity to biased label distributions and mismatched feature spaces.

In the subsequent experiment, we employed models that were previously trained across different merged configurations. We hypothesise that models trained within a specific dataset will perform well when applied to either that same dataset or a new multi-dataset that includes it, and that performance will degrade when applied to datasets that do not share the same underlying characteristics. The results of these experiments are presented in Figure 10.

Training on multiple datasets modestly improves generalisation, but cross-dataset performance remains lim-

Table 7 Summary of the combined and individual datasets showing sample size, attack–benign distribution, attack type diversity, and class entropy

Dataset	Total Samples	Attack Instances	Benign Instances	Distinct Attack Types	Class Entropy
CICIDS2017	2,830,540	557,645 (19.70%)	2,272,895 (80.30%)	8	1.0537
CICIDS2018	15,799,748	2,354,250 (14.90%)	13,445,498 (85.10%)	6	0.8840
CICDDoS2019	10,850,095	10,753,549 (99.11%)	96,546 (0.89%)	18	2.7393
UNSW-NB15	3,480,335	89,583 (2.57%)	3,390,752 (97.43%)	9	0.2284
CICIDS2017_CICIDS2018	18,630,288	2,911,895 (15.63%)	15,718,393 (84.37%)	9	0.9464
CICIDS2017_CICDDoS2019	13,680,635	11,311,194 (82.68%)	2,369,441 (17.32%)	26	3.0835
CICIDS2017_UNSW-NB15	6,310,875	647,228 (10.26%)	5,663,647 (89.74%)	16	0.7137
CICIDS2018_CICDDoS2019	26,649,843	13,107,799 (49.19%)	13,542,044 (50.81%)	24	2.5833
CICIDS2018_UNSW-NB15	19,280,083	2,443,833 (12.68%)	16,836,250 (87.32%)	14	0.8121
CICIDS2017_CICIDS2018_CICDDoS2019	29,480,383	13,665,444 (46.35%)	15,814,939 (53.65%)	27	2.5267
CICIDS2017_CICIDS2018_UNSW-NB15	22,110,623	3,001,478 (13.57%)	19,109,145 (86.43%)	17	0.8768
CICIDS2017_CICDDoS2019_UNSW-NB15	17,160,970	11,400,777 (66.43%)	5,760,193 (33.57%)	34	2.9021
CICIDS2018_CICDDoS2019_UNSW-NB15	30,130,178	13,197,382 (43.80%)	16,932,796 (56.20%)	32	2.4204
CICIDS2017_CICIDS2018_CICDDoS2019_UNSW-NB15	32,960,718	13,755,027 (41.73%)	19,205,691 (58.27%)	35	2.3774

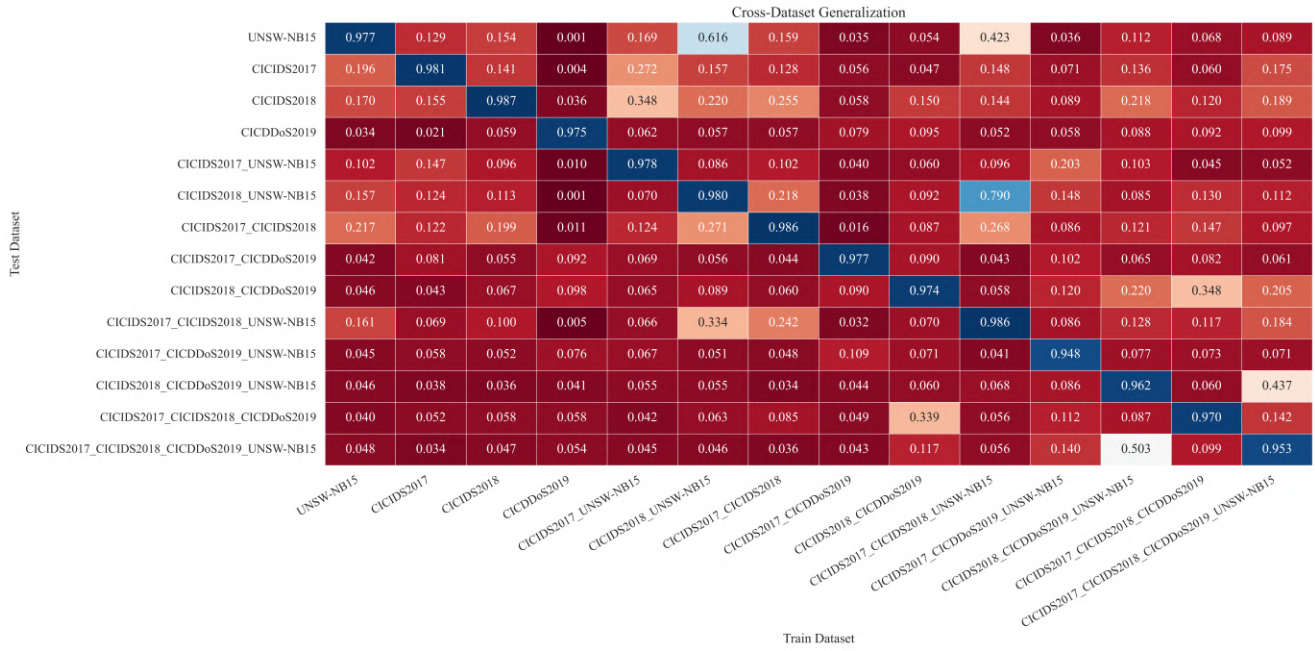

Fig. 10 Weighted average accuracy obtained for each cross-combination of datasets, aggregated over all executed models

ited, highlighting the challenges of dataset heterogeneity in intrusion detection. While training on complete datasets increases exposure to diverse attack types, the resulting models still struggle to generalise robustly across unseen datasets. The average cross-domain accuracy remains low, underlining the need for advanced domain adaptation or normalization techniques to effectively leverage multi-dataset training.

7.4 Adaptive Models Perform Better Under Data Drift

AI-based intrusion detection models are increasingly exposed to evolving adversarial threats and dynamic data environments that challenge their long-term performance and reliability. In particular, models capable of retraining or adapting to changes in the input distribution offer a promising direction for maintaining detection accuracy over time.

Adversarial Machine Learning (AML) attacks exploit vulnerabilities in AI systems to manipulate model behaviour, either during training or deployment [59, 60]. Training-time threats, such as data poisoning, model/parameter poisoning

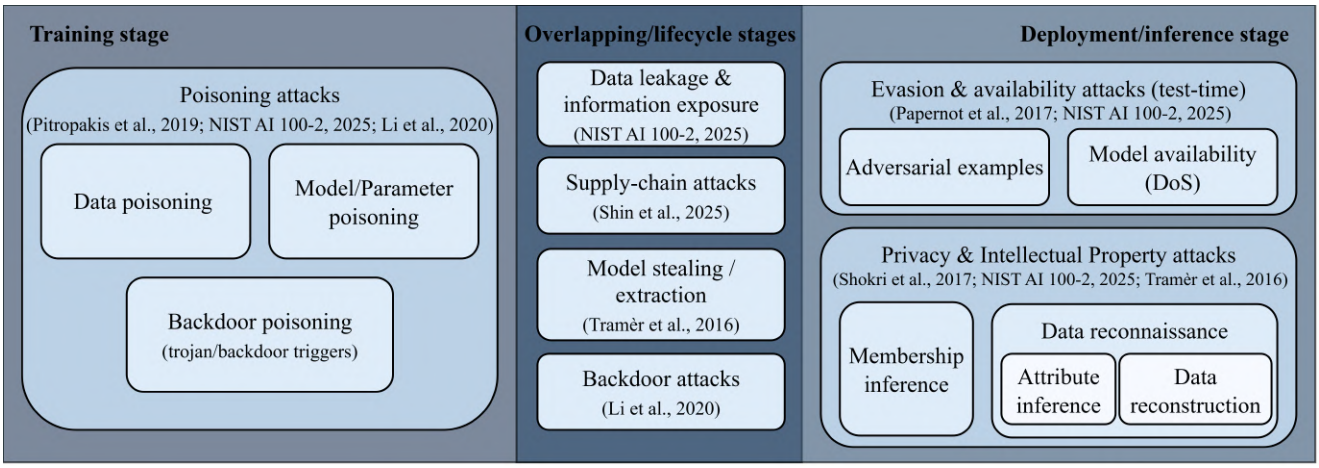

Fig. 11 Graphic representation of different AML attacks and their target stage, adapted from [59–65].

and backdoor poisoning, corrupt the learning process and can embed persistent malicious behaviours in the model [59, 61]. Deployment-time attacks include evasion via adversarial examples, which perturb inputs to evade detection, and attacks on model availability [60, 64]. Privacy- and IP-oriented attacks such as membership inference, attribute inference, data reconstruction and model stealing target either the confidentiality of the training data or the proprietary model itself [60, 63, 65]. Some attacks span multiple phases of the lifecycle, notably supply-chain attacks and backdoor attacks that are injected during training but triggered at inference time [61, 62]. This categorisation, adapted from prior work on AML taxonomies, is presented in Figure 11. These threats are particularly harmful in intrusion detection, where attackers can iteratively adapt and subtly mimic benign behaviour to remain undetected over time.

In parallel, another critical challenge facing AI detection systems is managing the impact of data drift. Concept drift refers to changes in the underlying data distribution over time that affect the target class relationships, either abruptly, incrementally, or in recurring patterns, leading to model performance degradation. In contrast, feature drift involves shifts in the relevance or importance of input features. Both types of drift are inevitable in real network environments due to evolving attack vectors, user behaviours, and traffic features.

Mitigating these effects requires integrating maintainability and verifiability into the model design. Adaptive methods such as sliding window retraining, ensemble learning with dynamic updates, and incremental learning allow models to remain responsive to shifting patterns. Techniques such as adversarial training, randomised smoothing and differential privacy strengthen models against crafted adversarial inputs and ensure robustness during continuous updates. Feature selection strategies that adapt over time and are triggered by performance drops, are essential to managing feature drift, especially in high-dimensional spaces.

The schematic in Figure 12 introduces an architecture that explicitly couples drift detection with adversarial-robust retraining, which are capabilities not present in any of the baseline models evaluated earlier. All preceding pipelines were trained in a static, offline fashion; even the best-performing hybrids assume a stationary feature distribution and offer, at most, batch retraining. Likewise, the few stream-oriented variants handle distribution shifts but provide no built-in defence against data-poisoning or evasion attacks. By embedding two dedicated feedback loops, one that flags concept/feature drift and triggers incremental updates, and another that hardens the refreshed model through adversarial training or randomised smoothing, the proposed framework closes this gap, allowing an IDS to maintain accuracy as traffic patterns evolve and remain resilient to crafted inputs aimed at exploiting the update process itself.

Following the flow shown in Figure 12, the proposed architecture begins by applying robust preprocessing techniques to ensure consistent input quality. A dedicated drift identification module continuously monitors the error metrics and statistical characteristics of the incoming data. When significant shifts are detected, this module distinguishes between concept drift and feature drift, thereby triggering appropriate adaptive mechanisms, such as sliding window retraining, dynamic ensemble updates, or incremental learning. This strategic separation allows the model to adapt effectively to evolving attack patterns, user behaviours, and traffic conditions in real-world environments.

To defend against adversarial threats, the proposed architecture incorporates verifiability measures such as adversarial training or randomised smoothing. These techniques are integrated throughout the pipeline to mitigate poisoning, evasion, and backdoor attacks, ensuring that the model remains robust despite both data drift and crafted adversarial inputs.

To validate that the proposed adaptive architecture can learn "on the fly," we ran a proof-of-concept streaming study.

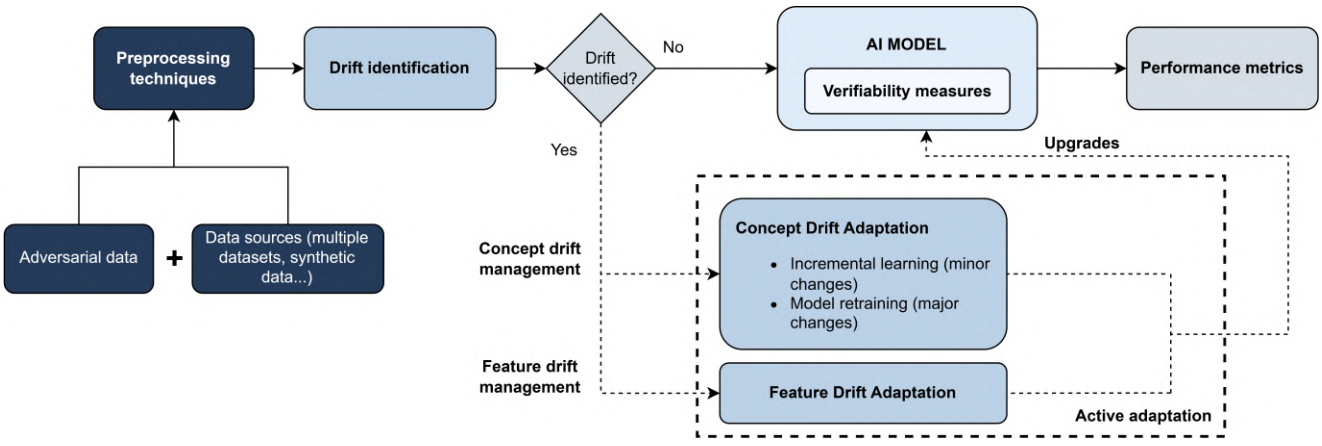

Fig. 12 Schema of adaptive AI detection models addressing data drift and adversarial resilience

Using the raw CIC-IDS2017 and CIC-IDS2018 flows, we replayed traffic in batches of 3,000–5,000 records and injected seven controlled drifts: four concept-level (gradual, sudden, recurring, incremental) and three feature-level (feature removal, feature addition, correlation shift). A Hoeffding-based monitor watched the error rate and SHAP patterns, triggering either fine-tuning of the existing weights or a full retrain of the hybrid NN + RF/LGBM ensemble whenever drift severity crossed a preset threshold.

Every drift was flagged in ≤ 4 batches; lightweight fine-tuning ($\approx$ 13 s) was sufficient for incremental, gradual and recurring drifts, restoring macro-accuracy to $\geq 96\%$. Destructive shifts, such as dropping an important feature, required a full retraining step ($\approx$ 74–77 s) but still recovered accuracy to $> 94\%$.

A second experiment assessed adversarial robustness. We poisoned up to 30% of the live stream with mislabeled or subtly perturbed flows. Without defences, baseline accuracy fell below 60% and MSE skyrocketed. After activating the hybrid RONI + K-Means sanitiser, the pipeline preserved $>$ 99% accuracy and kept MSE close to the clean baseline.

Together, these proof-of-concept runs demonstrate that the architecture not only detects both concept and feature drift quickly but also chooses the least-cost corrective action and remains resilient to data-poisoning attacks, capabilities none of the static baselines in earlier sections provide.

Adaptive AI detection architectures that integrate dynamic drift management with adversarial resilience may preserve detection accuracy and robustness in complex and evolving environments.

8 Discussion: Vulnerabilities, Adversarial Threats, and Model Trade-offs

While the experimental results presented in Section 7 demonstrate the superior performance of hybrid architectures, quantitative metrics alone do not fully capture the operational risks associated with deploying AI in real environments. In response to the need for a critical synthesis of security implications, this section analyses the intrinsic vulnerabilities of these models, the adversarial attacks they face, and the trade-offs inherent in current detection approaches.

AI-based IDSs inherit structural vulnerabilities that extend beyond simple classification errors. A primary weakness is the dependency on stationary data distributions; most supervised models assume that training and testing data share identical statistical characteristics, yet prior research indicates that relying on accurate modelling of normal traffic patterns often results in instability under fluctuating conditions. Network traffic, as proven in real scenarios, is highly non-stationary, and as noted in our drift analysis, models lacking dynamic update mechanisms degrade rapidly when exposed to concept drift or feature drift. Furthermore, DL models, while achieving high accuracy in theoretical benchmarks, often function as "black boxes". This lack of explainability creates a significant vulnerability, as security analysts cannot easily verify why a flow was flagged, making it difficult to distinguish between a genuine threat and a bias in the training data, while also hindering the auditing of models for potential backdoors injected during the training phase. Additionally, AI models exhibit extreme sensitivity to data quality and class imbalance. As evidenced by the performance drops in the CIC-DDoS2019 and UNSW-NB15 datasets discussed in Section 7.1 and Section 7.2, models without rigorous preprocessing tend to bias towards the majority class, leaving the system vulnerable to low-frequency attacks such as infiltration or zero-day exploits. To mitigate this limitation, several recent IDS proposals incorporate RL components, allowing models to continuously adapt detection or response strategies based on observed feedback. Such RL-driven mechanisms are particularly suited to zero-day contexts, where adaptive behaviour is required to cope with evolving and previously unknown attack patterns.

Beyond passive vulnerabilities, IDS models are active targets for AML. As categorized in the taxonomy presented in Figure 11, these threats exploit the learning principles of the models themselves to bypass detection. During the deployment or inference phase, attackers may employ evasion attacks by crafting adversarial examples. Without the adversarial training measures discussed in Section 7.4, the decision boundaries of standard ML and DL models remain fragile against such gradient-based perturbations. Threats also extend to the training phase through poisoning and backdoor attacks. In scenarios where IDSs are retrained on collected network logs, attackers can inject malicious data to gradually shift the decision boundary or implant a backdoor trigger. A triggered backdoor allows specific malicious traffic to pass undetected while the model performs normally on standard metrics. Furthermore, high-fidelity explanations or confidence scores returned by an IDS can be exploited for model extraction. This allows attackers to train surrogate models to test evasion strategies offline before launching a live attack, thereby compromising the system's intellectual property and security posture.

Synthesizing the experimental findings from Section 7 and the architectural analysis in Section 5 reveals distinct trade-offs between the three primary detection families. Traditional Machine Learning models offer high interpretability and lower computational costs for inference, making them robust in structured, tabular data contexts. However, their reliance on manual feature engineering limits their ability to detect complex, non-linear attack patterns in raw high-dimensional traffic, and they are generally less effective against novel attacks without frequent retraining. In contrast, Deep Learning models excel at automatic feature extraction

and modelling sequential data. Despite these advantages, they suffer from high training latency and significant computational resource demands, and their opacity makes them harder to trust in critical infrastructure. Hybrid Models consistently offer the best balance of generalization and accuracy across diverse datasets by combining the representational power of DL with the decision robustness of ML ensembles. However, this advantage comes with architectural complexity and operational overhead, as maintainability becomes challenging when drift detection and retraining pipelines must be coordinated across multiple heterogeneous components.

In addition to accuracy and robustness considerations, inference-time behaviour constitutes a critical but often underexplored dimension of operational risk in AI-based IDS deployments. While training-time complexity primarily affects offline model updates, inference latency and throughput directly constrain real-time detection capabilities. As demonstrated in the throughput analysis presented in Section 7.1, models with excessive inference overhead may fail to sustain continuous monitoring under high traffic loads, resulting in delayed alerts or dropped flows. This limitation introduces a systemic vulnerability, as attackers may intentionally generate traffic bursts to overwhelm the detection pipeline rather than evade the model's decision logic itself.

From a security perspective, inference inefficiency can therefore be exploited as a denial-of-service vector against the IDS, particularly in resource-constrained or edge-deployed environments. Models operating close to real-time throughput thresholds leave little margin for traffic surges or additional defensive layers such as explainability modules, logging, or adversarial checks. In contrast, ensemble-based and hybrid architectures that maintain stable inference throughput—despite higher training complexity—offer a more resilient operational profile, as they decouple detection effectiveness from traffic volume fluctuations.

These observations suggest that model selection should not be driven solely by detection performance on static benchmarks, but must jointly consider inference scalability, robustness under sustained load, and the ability to integrate additional security controls without degrading real-time responsiveness. Consequently, inference-aware evaluation emerges as a necessary complement to traditional accuracy-centric assessments when designing IDSs for adversarial and high-throughput environments.

The analysis of these vulnerabilities underscores that high accuracy on static datasets is insufficient for real-world security. The adaptive architecture proposed in Section 7.4 (see Figure 12) directly addresses these gaps through a multi-layered defence strategy. By decoupling concept drift from feature drift, the framework moves away from static paradigms and triggers incremental learning or full retraining only when statistical divergence is verified. To defend against adversarial threats, the integration of verifiability measures, such as randomized smoothing and robust preprocessing by using hybrid sanitizers like RONI + K-Means, serves as a defence against evasion and poisoning attempts. While standard preprocessing handles noise, these specialized defences are necessary to ensure the model's decision boundaries are not easily manipulated by crafted inputs. Finally, our cross-dataset analysis in Section 7.3 confirms that standardized feature engineering reduces the attack surface by normalizing inputs, making it harder for attackers to exploit outliers or unscaled feature values to skew model predictions.

9 Conclusion

This study presents a comprehensive analysis of AI-based approaches for anomaly and intrusion detection in networked environments. Through systematic experimentation using multiple public datasets, a broad set of preprocessing techniques, and diverse model families, we identified configurations that provide strong generalisation, robustness, and adaptability. Hybrid models emerged as the most balanced and effective solutions, particularly when tackling real-world complexities such as class imbalance or diverse and evolving attack patterns. More importantly, these findings reinforce a shift towards maintainable AI-driven detection, where long-term effectiveness, adaptability, and verifiability are prioritised over isolated accuracy gains.

Beyond model architecture, the results underscore the foundational role of data preprocessing in shaping both detection performance and operational efficiency. Feature selection, transformation, and resampling strategies were shown to substantially influence learning stability and inference behaviour, often determining whether a model remains viable under changing conditions.

At the same time, the analysis highlights that several challenges persist when transitioning AI-based detection systems from controlled benchmarks to operational environments. High-performing models frequently impose substantial computational demands, constraining their applicability in real-time or resource-limited settings. Furthermore, most intrusion detection pipelines remain largely static after deployment, making them vulnerable to concept and feature drift as network behaviours evolve. Adversarial manipulation also remains a critical concern, as carefully crafted inputs can undermine detection reliability if robustness mechanisms are not explicitly considered. Similarly, while explainability techniques such as LIME and SHAP are increasingly adopted, their predominantly post-hoc and model-agnostic nature limits their integration into operational workflows and their practical usefulness for security analysts. These issues are further exacerbated by the emergence of new technological environments, including 5G infrastructures, blockchain-based systems, and quantum-enabled platforms, which introduce novel traffic patterns and attack surfaces.

Building on these conclusions, future research should focus on creating adaptive, efficient, and maintainable detection systems that can perform in real time, adapt to new situations, and resist attacks. It is equally important to improve the quality and diversity of available datasets, promote comprehensive evaluation protocols, and ensure that AI applications in cybersecurity remain transparent, auditable, and secure as technology evolves.

Taken together, the findings of this study indicate that maintainability should be treated as a central criterion in the design and evaluation of AI-driven anomaly and threat detection systems. Rather than focusing exclusively on architectural complexity or marginal performance gains, sustained effectiveness in operational environments requires coherence between data, processing pipelines, and deployment constraints. This perspective offers a grounded foundation for advancing AI-based intrusion detection research toward solutions that remain effective, interpretable, and usable over time.

Acknowledgements

This publication is part of the project "CiberIA: Investigación e Innovación para la Integración de Ciberseguridad e Inteligencia Artificial" (Proyecto C079/23), financed by "European Union NextGeneration-EU, the Recovery Plan, Transformation and Resilience", through INCIBE. It has also been partially supported by the project SecAI (PID2022-139268OB-I00) funded by the Spanish Ministerio de Ciencia e Innovacion, and Agencia Estatal de Investigacion. Funding for open access charge: Universidad de Málaga / CBUA.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and Code Availability

All source code, experimental configurations, and binary datasets used in this study are openly available at the following OSF repository: https://osf.io/qubtk/?view_only=d57641c276294918b8cd5734640b080c.

This includes model implementations, preprocessing scripts, evaluation notebooks, and the full setup used for the proof-of-concept experiments described in Section 7. The repository is shared under a view-only link to ensure reproducibility and facilitate future research.

References

1. F. Sabahi and A. Movaghar. Intrusion detection: A survey. *2008 Third International Conference on Systems and Networks Communications*, pages 23–26, 2008. https://doi.org/10.1109/ICSNC.2008.44.
2. Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20):4396, October 2019. https://doi.org/10.3390/app9204396.
3. Oluwadamilare Harazeem Abdulganiyu, Taha Ait Tchakoucht, and Yakub Kayode Saheed. A systematic literature review for network intrusion detection system (ids). *International Journal of Information Security*, 22(5):1125–1162, March 2023. https://doi.org/10.1007/s10207-023-00682-2.
4. Wael Badawy. Ai-powered cybersecurity: A technical review of intrusion detection models, tools, and metrics. In *2025 International Telecommunications Conference (ITC-Egypt)*, pages 173–178. IEEE, 2025.
5. Robert A. Bridges, Tarrah R. Glass-Vanderlan, Michael D. Iannacone, Maria S. Vincent, and Qian Chen. A survey of intrusion detection systems leveraging host data. *ACM Computing Surveys (CSUR)*, 52:1–35, 2019.
6. Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, January 2016. https://doi.org/10.1016/j.jnca.2015.11.016.
7. Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2, 2019.
8. Monowar H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1):303–336, 2014. https://doi.org/10.1109/SURV.2013.052213.00046.
9. Y.S. Kalai Vani and Krishnamurthy. Survey anomaly detection in network using big data analytics. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pages 3366–3369, 2017. https://doi.org/10.1109/ICECDS.2017.8390083.
10. Anna Drewek-Ossowicka, Mariusz Pietrołaj, and Jacek Rumiński. A survey of neural networks usage for intrusion detection systems. *Journal of Ambient Intelligence and Humanized Computing*, 12(1):497–514, May 2020. https://doi.org/10.1007/s12652-020-02014-x.
11. Paulo Angelo Alves Resende and André Costa Drummond. A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys*, 51(3):1–36, May 2018. https://doi.org/10.1145/3178582.
12. Prachiti Parkar and Ansh Bilimoria. A survey on cyber security ids using ml methods. *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 352–360, 2021. https://doi.org/10.1109/ICICCS51141.2021.9432210.
13. Mohammed Sayeeduddin Habeeb and T. Ranga Babu. Network intrusion detection system: A survey on artificial intelligence-based techniques. *Expert Systems*, 39(9), July 2022. https://doi.org/10.1111/exsy.13066.
14. Murtaza Lokhandwala. Ai-powered intrusion detection systems for evolving cyber threats. *International Journal of Engineering & Extended Technologies Research (IJEETR)*, 7(5):10555–10558, 2025.
15. Robert Mitchell and Ing-Ray Chen. A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv.*, 46(4), mar 2014. https://doi.org/10.1145/2542049.
16. Antonia Nisioti, Alexios Mylonas, Paul D. Yoo, and Vasilios Katos. From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods. *IEEE Communications Surveys &*

Tutorials, 20(4):3369–3388, 2018. https://doi.org/10.1109/COMST.2018.2854724.

17. Ankit Thakkar and Ritika Lohiya. A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, 55(1):453–563, July 2021. https://doi.org/10.1007/s10462-021-10037-9.
18. Muhammad Rehan Naeem and Muhammad Farhan. *Bridging the gap: explainable AI in threat detection and cybersecurity*, page 23–47. The Institution of Engineering and Technology, October 2025. http://dx.doi.org/10.1049/PBSE027E_ch2.
19. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116, January 2018. https://doi.org/10.5220/0006639801080116.
20. Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A. Ghorbani. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8, 2019. https://doi.org/10.1109/CCST.2019.8888419.
21. Nour Moustafa and Jill Slay. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *Military Communications and Information Systems Conference (MilCIS)*, 11 2015. https://doi.org/10.1109/milcis.2015.7348942.
22. Hesamodin Mohammadian, Arash Habibi Lashkari, and Ali A. Ghorbani. Poisoning and evasion: Deep learning-based nids under adversarial attacks. In *2024 21st Annual International Conference on Privacy, Security and Trust (PST)*, page 1–9. IEEE, August 2024. https://doi.org/10.1109/pst62714.2024.10788064.
23. Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6), jun 2004. https://doi.org/10.1103/physreve.69.066138.
24. Ouail Mjahed, Salah El Hadaj, El Mahdi El Guarmah, and Soukaina Mjahed. Improved supervised and unsupervised metaheuristic-based approaches to detect intrusion in various datasets. *Computer Modeling in Engineering & Sciences*, 137(1):265–298, 2023. https://doi.org/10.32604/cmes.2023.027581.
25. Jorge Buzzio-García, Jaime Vergara, Santiago Ríos-Guiral, Christian Garzón, Sergio Gutiérrez, Juan F. Botero, Jose Luis Quiroz-Arroyo, and Jesús Arturo Pérez-Díaz. Exploring traffic patterns through network programmability: Introducing sdnflow, a comprehensive openflow-based statistics dataset for attack detection. *IEEE Access*, 12:42163–42180, 2024. https://doi.org/10.1109/access.2024.3378271.
26. Mahmoud Said Elsayed, Nhien-An Le-Khac, and Anca D. Jurcut. Insdn: A novel sdn intrusion dataset. *IEEE Access*, 8:165263–165284, 2020. https://doi.org/10.1109/access.2020.3022633.
27. Abdulsalam O. Alzahrani and Mohammed J. F. Alenazi. Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet*, 13(5):111, April 2021. https://doi.org/10.3390/fi13050111.
28. Hanafi Hanafi, Alva Hendi Muhammad, Ike Verawati, and Richki Hardi. An intrusion detection system using sdae to enhance dimensional reduction in machine learning. *JOIV: International Journal on Informatics Visualization*, 6(2):306, June 2022. https://doi.org/10.30630/joiv.6.2.990.
29. James Halladay, Drake Cullen, Nathan Briner, Jackson Warren, Karson Fye, Ram Basnet, Jeremy Bergen, and Tenzin Doleck. Detection and characterization of ddos attacks using time-based features. *IEEE Access*, 10:49794–49807, 2022. https://doi.org/10.1109/access.2022.3173319.
30. Johan Note and Maaruf Ali. Comparative analysis of intrusion detection system using machine learning and deep learning algorithms. *Annals of Emerging Technologies in Computing*, 6(3):19–36, July 2022. https://doi.org/10.33166/aetic.2022.03.003.
31. Robin Duraz, David Espes, Julien Francq, and Sandrine Vaton. Cyber informedness: A new metric using cvss to increase trust in intrusion detection systems. In *European Interdisciplinary Cybersecurity Conference*, EICC 2023, page 7. ACM, June 2023. https://doi.org/10.1145/3590777.3590786.
32. Noe Marcelo Yungaicela-Naula, Cesar Vargas-Rosales, and Jesus Arturo Perez-Diaz. Sdn-based architecture for transport and application layer ddos attack detection by using machine and deep learning. *IEEE Access*, 9:108495–108512, 2021. https://doi.org/10.1109/access.2021.3101650.
33. Khattab M. Ali Alheeti, Abdulkareem Alzahrani, Omar Hammad Jasim, Duaa Al-Dosary, Hamsa M. Ahmed, and Muzhir Shaban Al-Ani. Intelligent detection system for multi-step cyber-attack based on machine learning. In *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, page 7. IEEE, January 2023. https://doi.org/10.1109/dese58274.2023.10100226.
34. Treepop Wisanwanichthan and Mason Thammawichai. A double-layered hybrid approach for network intrusion detection system using combined naive bayes and svm. *IEEE Access*, 9:138432–138450, 2021. https://doi.org/10.1109/access.2021.3118573.
35. Mohamed Azalmad, Rachid El Ayachi, and Mohamed Biniz. Unveiling the performance insights: Benchmarking anomaly-based intrusion detection systems using decision tree family algorithms on the cicids2017 dataset. *Lecture Notes in Business Information Processing*, page 202–219, 2023. https://doi.org/10.1007/978-3-031-37872-0_15.
36. Asmaa Halbouni, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Murad Halbouni, Mira Kartiwi, and Robiah Ahmad. Cnn-lstm: Hybrid deep neural network for network intrusion detection system. *IEEE Access*, 10:99837–99849, 2022. https://doi.org/10.1109/access.2022.3206425.
37. Muataz Salam Al-Daweri, Salwani Abdullah, and Khairul Akram Zainol Ariffin. An adaptive method and a new dataset, ukm-ids20, for the network intrusion detection system. *Computer Communications*, 180:57–76, December 2021. https://doi.org/10.1016/j.comcom.2021.09.007.
38. Chi Mai Kim Ho, Kin-Choong Yow, Zhongwen Zhu, and Sarang Aravamuthan. Network Intrusion Detection via Flow-to-Image Conversion and Vision Transformer Classification. *IEEE Access*, 10:97780–97793, 1 2022. https://doi.org/10.1109/access.2022.3200034.
39. Kamal A. ElDahshan, AbdAllah A. AlHabshy, and Bashar I. Hameed. Meta-heuristic optimization algorithm-based hierarchical intrusion detection system. *Computers*, 11(12):170, November 2022. https://doi.org/10.3390/computers11120170.
40. Qasem Abu Al-Haija, Shahad Altamimi, and Mazen AlWadi. Analysis of extreme learning machines (elms) for intelligent intrusion detection systems: A survey. *Expert Systems with Applications*, 253:124317, November 2024. https://doi.org/10.1016/j.eswa.2024.124317.
41. Shahad Altamimi and Qasem Abu Al-Haija. Maximizing intrusion detection efficiency for iot networks using extreme learning machine. *Discover Internet of Things*, 4(1), July 2024. https://doi.org/10.1007/s43926-024-00060-x.
42. Ankush Singla, Elisa Bertino, and Dinesh Verma. Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '20, page 14. ACM, October 2020. https://doi.org/10.1145/3320269.3384718.

43. Sai Prasath, Kamalakanta Sethi, Dinesh Mohanty, Padmalochan Bera, and Subhransu Ranjan Samantaray. Analysis of continual learning models for intrusion detection system. *IEEE Access*, 10:121444–121464, 2022. https://doi.org/10.1109/access.2022.3222715.
44. Muhammad Rehan Naeem, Rashid Amin, Muhammad Farhan, Faisal S. Alsubaei, Eesa Alsolami, and Muhammad D. Zakaria. Cyber security enhancements with reinforcement learning: A zero-day vulnerabilityu identification perspective. *PLOS One*, 20(5):e0324595, May 2025. http://dx.doi.org/10.1371/journal.pone.0324595.
45. Ruki Harwahyu, Fajar Henri Erasmus Ndolu, and Marlinda Vasty Overbeek. Three layer hybrid learning to improve intrusion detection system performance. *International Journal of Electrical and Computer Engineering (IJECE)*, 14(2):1691, April 2024. https://doi.org/10.11591/ijece.v14i2.pp1691-1699.
46. Devika Jaiswal, Kunj Arora, Manan Varshney, and Trasha Gupta. A hybrid method for network intrusion detection. In *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, page 7. IEEE, May 2023. https://doi.org/10.1109/icsccc58608.2023.10176754.
47. Yuyang Zhou, Guang Cheng, Shanqing Jiang, and Mian Dai. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks*, 174:107247, June 2020. https://doi.org/10.1016/j.comnet.2020.107247.
48. Saeid Sheikhi and Panos Kostakos. A novel anomaly-based intrusion detection model using psogwo-optimized bp neural network and ga-based feature selection. *Sensors*, 22(23):9318, November 2022. https://doi.org/10.3390/s22239318.
49. Anil V Turukmane and Ramkumar Devendiran. M-multisvm: An efficient feature selection assisted network intrusion detection system using machine learning. *Computers & Security*, 137:103587, February 2024. https://doi.org/10.1016/j.cose.2023.103587.
50. Tengfei Yang, Yuansong Qiao, and Brian Lee. Towards trustworthy cybersecurity operations using bayesian deep learning to improve uncertainty quantification of anomaly detection. *Computers & Security*, 144:103909, September 2024. https://doi.org/10.1016/j.cose.2024.103909.
51. Furqan Rustam and Anca Delia Jurcut. Malicious traffic detection in multi-environment networks using novel s-date and pso-d-sem approaches. *Computers &; Security*, 136:103564, January 2024. https://doi.org/10.1016/j.cose.2023.103564.
52. Hussein Al-Zoubi and Samah Altaamneh. A feature selection technique for network intrusion detection based on the chaotic crow search algorithm. In *2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, page 10. IEEE, September 2022. https://doi.org/10.1109/idsta55301.2022.9923108.
53. Shahriar Mohammadi and Mehdi Babagoli. A novel hybrid hunger games algorithm for intrusion detection systems based on nonlinear regression modeling. *International Journal of Information Security*, 22(5):1177–1195, April 2023. https://doi.org/10.1007/s10207-023-00684-0.
54. Meng Wang, Yiqin Lu, and Jiancheng Qin. A dynamic mlp-based ddos attack detection method using feature selection and feedback. *Computers & Security*, 88:101645, January 2020. https://doi.org/10.1016/j.cose.2019.101645.
55. Mhamad Bakro, Rakesh Ranjan Kumar, Mohammad Husain, Zubair Ashraf, Arshad Ali, Syed Irfan Yaqoob, Mohammad Nadeem Ahmed, and Nikhat Parveen. Building a cloud-ids by hybrid bio-inspired feature selection algorithms along with random forest model. *IEEE Access*, 12:8846–8874, 2024. https://doi.org/10.1109/access.2024.3353055.
56. Mustafa Sabah Noori, Ratna K. Z. Sahbudin, Aduwati Sali, and Fazirulhisyam Hashim. Feature drift aware for intrusion detection system using developed variable length particle swarm optimization in data stream. *IEEE Access*, 11:128596–128617, 2023. https://doi.org/10.1109/access.2023.3333000.
57. Deepa Manikandan and Jayaseelan Dhilipan. Machine learning approach for intrusion detection system using dimensionality reduction. *Indonesian Journal of Electrical Engineering and Computer Science*, 34(1):430, 2 2024. https://doi.org/10.11591/ijeecs.v34.i1.pp430-440.
58. Didik Sudyana, Ying-Dar Lin, Miel Verkerken, Ren-Hung Hwang, Yuan-Cheng Lai, Laurens D'Hooge, Tim Wauters, Bruno Volckaert, and Filip De Turck. Improving generalization of ml-based ids with lifecycle-based dataset, auto-learning features, and deep learning. *IEEE Transactions on Machine Learning in Communications and Networking*, 2:645–662, 2024. https://doi.org/10.1109/tmlcn.2024.3402158.
59. Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Elias Anastasiadis, and George Loukas. A taxonomy and survey of attacks against machine learning. *Computer Science Review*, 34:100199, 2019. https://doi.org/10.1016/j.cosrev.2019.100199.
60. Apostol Vassilev et al. Adversarial machine learning: A taxonomy and terminology of attacks and mitigations. Technical Report NIST AI 100-2e2025, National Institute of Standards and Technology, Gaithersburg, MD, 2025. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2025.pdf.
61. Yuntao Li, Longfei Lyu, Xingjun Ma, Jing Jiang, James Bailey, Feng Chen, and Shouling Yu. Backdoor attacks and defenses in deep learning: A survey. *IEEE Access*, 8:124205–124222, 2020. https://doi.org/10.1109/ACCESS.2020.3007470.
62. Seoyoung Shin et al. Down the rabbit hole of backdoors in the AI supply chain. *arXiv preprint*, 2025. https://arxiv.org/abs/2510.05159.
63. Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *Proceedings of the 25th USENIX Security Symposium*, pages 601–618, 2016. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer.
64. Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519, 2017. https://doi.org/10.1145/3052973.3053009.
65. Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017. https://doi.org/10.1109/SP.2017.41.