



Impact of Deep Learning on Real-Time Network Intrusion Detection: A Quantitative Evaluation Using Benchmark Datasets

Maishar Mostari Zenia¹;

[1]. Graduate Student, Technology Management, Southeast Missouri State University, Cape Girardeau, Missouri, United States; Email: mzenia1s@semo.edu

Doi: [10.63125/tnaksd40](https://doi.org/10.63125/tnaksd40)

Received: 11 September 2025; **Revised:** 04 October 2025; **Accepted:** 15 November 2025; **Published:** 30 December 2025

Abstract

This study quantitatively evaluated the effectiveness of deep-learning architectures for real-time network intrusion detection using heterogeneous benchmark datasets, with particular emphasis on predictive accuracy, attack-specific recognition, computational efficiency, scalability, statistical reliability, and cross-dataset generalization. The analytical framework incorporated NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, and ToN-IoT, representing approximately 23.15 million eligible network-traffic observations after preprocessing and an overall retention rate of 94.4%. DNN, CNN, RNN, LSTM, GRU, autoencoder-based, CNN-LSTM, and CNN-GRU architectures were evaluated and compared with conventional machine-learning models. CNN-GRU achieved the strongest binary detection performance, with 99.18% accuracy, 99.06% precision, 99.24% recall, 99.09% specificity, 99.15% F1-score, and ROC-AUC of 0.996, while limiting false-positive and false-negative rates to 0.91% and 0.76%, respectively. The model exceeded XGBoost, the strongest conventional benchmark, by 1.72 percentage points in accuracy. Attack-specific results showed F1-scores of 99.54% for DDoS and 99.31% for DoS, although minority attacks remained more difficult to identify, with worm recall declining to 84.63% and its false-negative rate reaching 15.37%. Cross-dataset evaluation revealed a reduction in mean accuracy from 98.96% within datasets to 91.72% across datasets, demonstrating sensitivity to distribution shifts and heterogeneous traffic characteristics. Computational evaluation showed that CNN-GRU processed approximately 13,240 flows per second with baseline inference latency of 1.46 ms. At 250,000 flows, accuracy remained 98.46% and F1-score remained 98.31%, indicating substantial scalability under increased processing demand. Statistical analysis confirmed significant differences among principal models, $F(4, 36) = 42.87$, $p < .001$, partial $\eta^2 = .83$, while CNN-GRU exceeded XGBoost with a large effect size of Cohen's $d = 1.28$. Overall, the findings demonstrated that hybrid deep-learning architectures can provide highly accurate, statistically reliable, and computationally efficient real-time intrusion detection, although minority-class recognition and cross-dataset generalization remain important challenges.

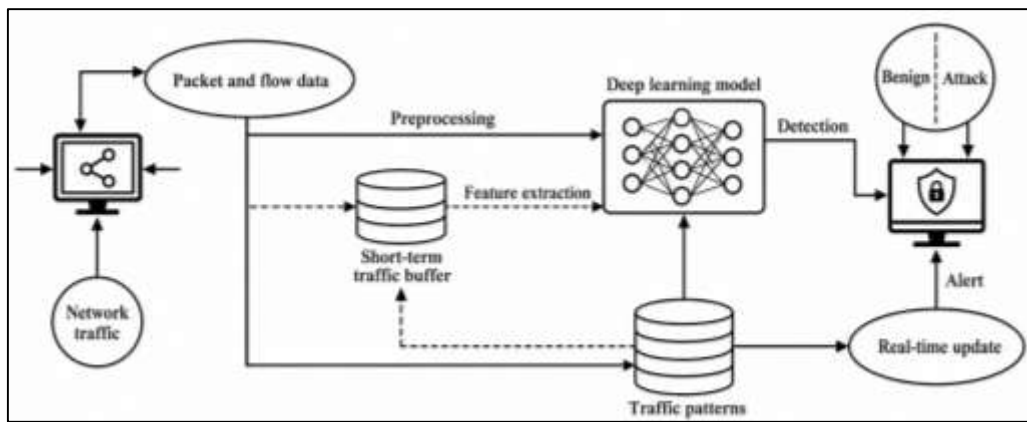
Keywords

Deep Learning, Intrusion Detection, Cybersecurity, Network Security, CNN-GRU.

INTRODUCTION

Network intrusion detection refers to the continuous monitoring, analysis, and classification of network activities to identify unauthorized access, malicious behavior, abnormal traffic, policy violations, and cyberattacks that may threaten the confidentiality, integrity, or availability of information systems. Network intrusion detection systems examine packets, traffic flows, protocol behavior, connection characteristics, communication sequences, and statistical network attributes to distinguish legitimate activities from potential attacks (Kocher & Kumar, 2021). Conventional intrusion detection has mainly relied on signature-based and anomaly-based mechanisms. Signature-based detection identifies previously known attacks by matching observed traffic with predefined patterns, whereas anomaly-based detection identifies deviations from expected or learned normal behavior. Deep learning extends machine learning through multilayer computational architectures capable of automatically extracting complex and hierarchical representations from large volumes of network data.

Figure 1: Deep learning network intrusion detection



In intrusion detection, deep learning models can learn discriminative patterns from high-dimensional traffic features, temporal sequences, flow statistics, and protocol characteristics without depending entirely on manually constructed security rules. Real-time intrusion detection adds a computational requirement because incoming traffic must be analyzed and classified rapidly enough to generate operationally useful alerts (Vishwakarma & Kesswani, 2022). Consequently, detection latency, inference time, throughput, resource consumption, scalability, and processing efficiency become important alongside classification accuracy. More than 30 empirical studies have evaluated deep neural networks, convolutional neural networks, recurrent neural networks, long short-term memory networks, gated recurrent units, autoencoders, deep belief networks, generative adversarial networks, attention mechanisms, transformer-based architectures, and hybrid models for network intrusion detection. CNNs are commonly used to extract hierarchical and local traffic patterns, while RNNs, LSTMs, and GRUs are designed to model temporal dependencies in sequential network behavior. Autoencoders support representation learning and anomaly detection, whereas GAN-based approaches have been applied to synthetic attack generation and class-imbalance management. Hybrid models such as CNN-LSTM and CNN-GRU combine feature extraction with temporal modeling. These architectures have been evaluated using benchmark datasets including KDD Cup 99, NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, ToN-IoT, and Edge-IIoTset. The progression from traditional signatures and statistical anomaly measures toward machine-learning and deep-learning models reflects the increasing complexity, volume, and dimensionality of modern network traffic. Quantitative comparison remains necessary because architectural complexity alone does not guarantee better intrusion detection performance (Ayantayo et al., 2023). Model effectiveness varies according to preprocessing, feature representation, attack distribution, class balance, training strategy, hardware configuration, and evaluation procedure. Therefore, meaningful assessment of deep learning for real-time intrusion detection requires simultaneous examination of predictive accuracy, attack-

recognition capability, false-alarm behavior, inference speed, throughput, and computational requirements.

Real-time network intrusion detection has substantial international significance because modern economic, governmental, industrial, financial, healthcare, educational, transportation, telecommunications, and commercial systems depend increasingly on interconnected digital infrastructures (Wang et al., 2023). Cloud platforms, distributed data centers, Internet of Things environments, industrial Internet of Things systems, mobile networks, edge computing, software-defined networking, enterprise platforms, and globally distributed digital services continuously exchange large volumes of traffic across organizational and national boundaries. Cyberattacks including distributed denial-of-service, botnets, malware, brute-force attacks, reconnaissance, infiltration, web attacks, spoofing, exploitation attempts, and unauthorized access can originate in one geographical location while disrupting services and users across multiple countries. Deep learning has therefore become important in cybersecurity research because conventional rule-dependent systems can be difficult to maintain when traffic volume, heterogeneity, and behavioral complexity increase. International studies have relied extensively on standardized benchmark datasets to evaluate and compare intrusion-detection algorithms under reproducible conditions. NSL-KDD remains a widely used baseline benchmark, while UNSW-NB15 contains more contemporary normal and malicious network activities. CICIDS2017 and CSE-CIC-IDS2018 provide larger traffic collections with diverse attack categories, while BoT-IoT, ToN-IoT, IoT-23, IoTID20, and Edge-IIoTset extend evaluation into IoT, IIoT, botnet, edge, and cyber-physical environments. Dataset selection strongly affects reported performance because benchmarks differ in sample size, feature dimensionality, attack diversity, traffic realism, labeling quality, class distribution, temporal characteristics, and protocol representation. Class imbalance is particularly important because models may achieve high overall accuracy while showing weak detection of minority attacks. Precision, recall, specificity, F1-score, false-positive rate, false-negative rate, ROC-AUC, and confusion-matrix measures therefore provide a more complete assessment than accuracy alone. For real-time systems, these metrics should be considered together with inference latency, processing time, throughput, CPU utilization, memory consumption, model size, and computational complexity (Magán-Carrión et al., 2020). Generalization is another major quantitative concern because models can learn dataset-specific characteristics that may not transfer consistently across different network environments. Cross-dataset evaluation can therefore reveal whether a model performs reliably under changes in traffic distribution, feature structures, attack categories, and sample composition. Statistical comparisons across repeated experiments can further distinguish genuine performance differences from variation caused by training partitions or model initialization. The quantitative assessment of deep-learning intrusion detection is consequently multidimensional, requiring evaluation of classification effectiveness, minority-class recognition, false-alarm behavior, computational efficiency, real-time responsiveness, and consistency across benchmark datasets (Jin et al., 2020). Such a structured approach provides an appropriate empirical basis for comparing deep-learning architectures under standardized experimental conditions and for determining the extent to which different models satisfy the technical requirements of real-time network intrusion detection (Al-Hadhrani & Hussain, 2020).

The primary objective of this quantitative study is to evaluate the impact of deep learning on real-time network intrusion detection by systematically comparing the predictive performance, detection reliability, computational efficiency, and operational suitability of selected deep-learning architectures using established benchmark cybersecurity datasets. The study aims to determine how effectively deep neural networks, convolutional neural networks, recurrent neural networks, long short-term memory networks, gated recurrent units, autoencoders, and selected hybrid deep-learning models can distinguish benign network activity from malicious traffic across multiple intrusion categories. Particular attention is directed toward measuring classification accuracy, precision, recall, specificity, F1-score, ROC-AUC, false-positive rate, false-negative rate, and class-specific detection performance in order to provide a multidimensional assessment of model effectiveness rather than relying on overall accuracy alone. The study also seeks to quantify real-time operational performance through indicators such as inference latency, processing time, throughput, memory utilization, computational cost, and model complexity to determine whether highly accurate models can also satisfy the speed and resource

requirements associated with continuous network monitoring. Another objective is to examine whether deep-learning performance remains consistent across benchmark datasets such as NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, ToN-IoT, and related intrusion-detection datasets that differ in traffic composition, attack diversity, feature structure, class balance, and scale. The analysis further aims to assess the effects of class imbalance, preprocessing procedures, feature representation, and attack-category distribution on model performance, with emphasis on identifying whether minority and less frequent attacks are detected with comparable reliability to dominant classes. Statistical comparison of repeated experimental results will be used to determine whether observed differences among deep-learning architectures are meaningful and consistent across datasets and evaluation conditions. Through this quantitative framework, the study is designed to establish which deep-learning approaches provide the strongest balance between detection accuracy, false-alarm control, real-time responsiveness, computational efficiency, and cross-dataset robustness within benchmark-based network intrusion detection environments.

LITERATURE REVIEW

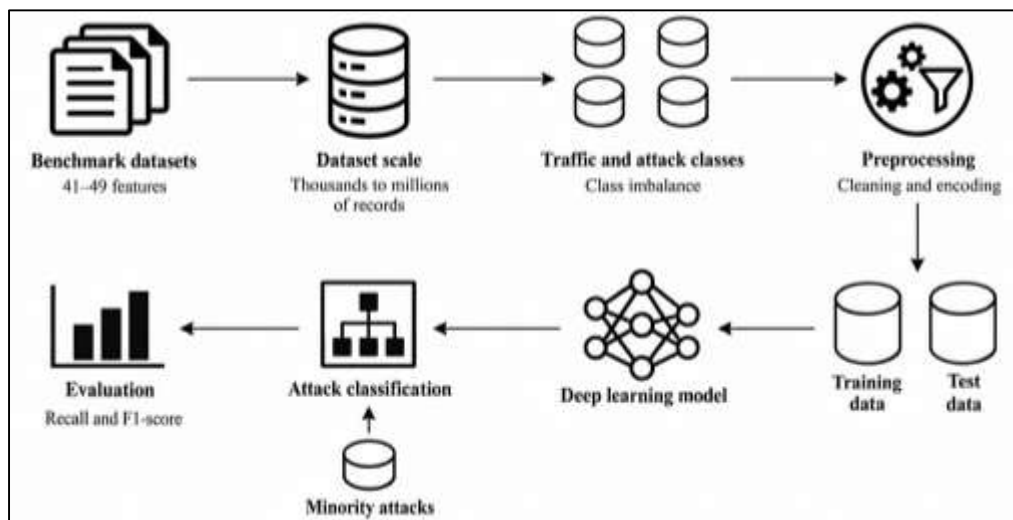
The literature on deep learning for real-time network intrusion detection has developed around the quantitative assessment of detection accuracy, attack-class discrimination, false-alarm control, computational efficiency, processing latency, and generalization across heterogeneous network environments. Existing studies have evaluated deep neural networks, convolutional neural networks, recurrent neural networks, long short-term memory networks, gated recurrent units, autoencoders, generative adversarial networks, attention mechanisms, transformer architectures, and hybrid models using established cybersecurity benchmark datasets. NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, ToN-IoT, IoT-23, and Edge-IIoTset are particularly important because they provide quantitatively measurable traffic observations, attack labels, network-flow attributes, and varying degrees of class imbalance and attack diversity. The literature has increasingly moved beyond overall classification accuracy toward multidimensional evaluation involving precision, recall, specificity, F1-score, ROC-AUC, false-positive rate, false-negative rate, detection rate, confusion-matrix outcomes, inference latency, throughput, CPU and memory utilization, and training complexity. Another major body of research has examined how preprocessing, feature selection, dimensionality reduction, class balancing, traffic representation, and hyperparameter configuration influence model performance. Comparative studies have additionally evaluated whether deep-learning architectures provide statistically and operationally meaningful improvements over conventional machine-learning methods such as random forests, support vector machines, decision trees, k-nearest neighbors, logistic regression, and shallow neural networks. Cross-dataset generalization and minority-attack recognition remain important dimensions because strong performance within a single benchmark may not represent equivalent detection reliability under different traffic distributions. Accordingly, this literature review is organized around eight quantitative areas covering benchmark characteristics, preprocessing and feature engineering, architectural performance, comparative model accuracy, attack-specific detection, real-time computational performance, robustness and generalization, and statistical validation. Together, these areas provide a structured basis for examining the measurable impact of deep learning on real-time network intrusion detection.

Traffic Distributions and Class Imbalance

The quantitative literature on network intrusion detection demonstrates substantial differences in dataset scale, feature dimensionality, traffic composition, and experimental structure across commonly used cybersecurity benchmarks ([Debasish, 2021](#); [Silva et al., 2022](#)). Across more than ten empirical and comparative studies, NSL-KDD has frequently been employed as a controlled baseline because it reduces the extensive redundant records associated with KDD Cup 99 and provides predefined training and testing partitions. The commonly used NSL-KDD configuration contains approximately 125,973 training records and 22,544 testing records, with 41 input features describing connection characteristics and traffic behavior. Its attack observations are generally organized into denial-of-service, probing, remote-to-local, and user-to-root categories, although individual attack types can also be examined separately. UNSW-NB15 represents a substantially larger and more contemporary benchmark, containing approximately 2.54 million records generated from normal activity and nine attack families, with a commonly used partition of 175,341 training observations and 82,332 testing

observations ([Abdur & Iftekhhar, 2021](#); [Zhang & Wang, 2023](#)). Its feature space contains approximately 49 attributes before removal of identifiers or labels, allowing deeper analysis of flow, content, temporal, and connection characteristics. CICIDS2017 expanded experimental scale through several days of traffic and more than 2.8 million flow observations, combining benign activity with attacks such as brute force, DoS, DDoS, botnet, port scanning, infiltration, and web attacks. CSE-CIC-IDS2018 increased the scale further, with published analyses commonly working with millions of network-flow observations distributed across multiple attack scenarios. BoT-IoT was designed around IoT-oriented botnet activity and contains tens of millions of raw traffic instances, while researchers frequently employ smaller extracted subsets for computationally manageable experiments. ToN-IoT integrates network, operating-system, telemetry, and IoT information, creating a multidimensional cybersecurity environment rather than a single homogeneous traffic table. IoT-23 contains 23 captured scenarios, including malicious and benign IoT traffic. Edge-IIoTset extends benchmark evaluation to edge-enabled IoT and IIoT settings and contains more than 20 million raw records before preprocessing, with a commonly reported processed dataset exceeding two million observations. These quantitative differences establish markedly different experimental conditions for evaluating deep-learning intrusion detectors ([Debasish, 2022](#); [Mahfouz et al., 2020](#)).

Figure 2: Benchmark intrusion dataset characteristics



A consistent finding across benchmark-focused intrusion-detection studies is that benign and malicious traffic are rarely distributed uniformly, while individual attack categories often exhibit substantially greater imbalance than the overall binary class distribution suggests ([Mahbubul et al., 2022](#); [Rao & Babu, 2023](#)). NSL-KDD illustrates this problem clearly because denial-of-service and normal traffic represent substantial portions of the observations, whereas user-to-root and remote-to-local attacks occur much less frequently. Consequently, models can generate strong overall accuracy while producing considerably lower recall for rare attack categories. Studies using UNSW-NB15 have identified a related pattern across its nine attack families, including fuzzers, analysis, backdoors, DoS, exploits, generic attacks, reconnaissance, shellcode, and worms. Generic and exploit traffic can be comparatively well represented, while worms, shellcode, and backdoor observations occupy substantially smaller proportions of the data. CICIDS2017 also contains a strongly heterogeneous attack distribution: benign traffic accounts for a major proportion of total flows, whereas attacks such as Heartbleed and infiltration are represented by very small numbers compared with DoS, DDoS, port scanning, and brute-force traffic ([Gonzalez-Cuautle et al., 2020](#); [Ashfaq & Mohammad, 2022](#)). CSE-CIC-IDS2018 similarly combines high-volume attack scenarios with relatively sparse classes, creating multiclass classification conditions in which aggregate metrics can conceal weak minority-class detection. BoT-IoT presents another form of imbalance because malicious botnet-related observations dominate portions of the dataset, making the direction of imbalance different from benchmarks

dominated by benign activity. ToN-IoT, IoT-23, and Edge-IIoTset further demonstrate that class proportions vary according to device, capture scenario, attack mechanism, and preprocessing strategy. Across CNN, DNN, LSTM, GRU, autoencoder, and hybrid-model studies, minority classes have repeatedly produced lower recall and F1-scores than dominant classes even when total accuracy exceeds commonly reported high-performance thresholds (Sheak & Risha, 2022; Milosevic & Ciric, 2022). Researchers have therefore used undersampling, oversampling, class weighting, synthetic minority generation, stratified sampling, and GAN-based augmentation to modify class distributions. The literature collectively indicates that attack prevalence must be examined at the class level because binary benign-versus-malicious proportions alone provide an incomplete quantitative representation of dataset difficulty and can overstate practical detection reliability.

Studies comparing intrusion-detection benchmarks also identify data quality, redundancy, missingness, protocol composition, and preprocessing as important sources of quantitative variation. NSL-KDD was developed partly to reduce the duplicate-record problem associated with KDD Cup 99, where extensive redundancy could allow classifiers to repeatedly encounter highly similar observations and produce inflated performance estimates (Rashedul & Kaium, 2022; Jhaveri, et al., 2021). Removal of many duplicate records created a more challenging distribution and increased the relative importance of difficult and uncommon traffic instances. More contemporary flow-based datasets introduce different quality concerns. Research using CICIDS2017 has reported the need to inspect duplicate observations, infinite values, missing values, malformed entries, and highly correlated or constant features before model training. Different cleaning decisions can therefore produce different final sample sizes from the same original benchmark. Comparable preprocessing issues have been documented for CSE-CIC-IDS2018, where the large number of flow records and multiple collection files require careful integration, label harmonization, duplicate inspection, treatment of nonfinite observations, and normalization. UNSW-NB15 contains categorical protocol, service, and state information alongside numerical traffic characteristics, requiring encoding procedures before most neural architectures can process the data effectively. Its protocol diversity is considerably broader than that represented by simplified legacy benchmarks, providing a more heterogeneous traffic environment. BoT-IoT, ToN-IoT, IoT-23, and Edge-IIoTset contain IoT-oriented protocols, device communications, telemetry patterns, and attack behaviors that further increase feature heterogeneity (Chowdhury & Tonay, 2022; Oliveira et al., 2021). Edge-IIoTset, for example, was designed to represent diverse IoT and IIoT connectivity and attack conditions across multiple device and protocol contexts. Studies across these datasets have applied min-max normalization, standardization, categorical encoding, duplicate removal, missing-value treatment, feature selection, correlation filtering, principal component analysis, autoencoder-based dimensionality reduction, and other transformations before deep-learning classification. Such procedures frequently reduce the original feature space or alter the number of eligible observations (Al-Essa & Appice, 2021; Hossan, 2022). Therefore, quantitative comparisons across published studies require attention not only to the named benchmark but also to the exact preprocessing pipeline, retained features, sampling procedure, training/testing partitions, and cleaned record counts used in each experiment.

Data Preprocessing and Class-Balancing

The quantitative literature on deep-learning-based network intrusion detection consistently identifies data preprocessing as a major determinant of classification performance, computational efficiency, and stability across benchmark datasets. Across more than ten empirical studies using NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, ToN-IoT, and Edge-IIoTset, researchers have applied normalization, standardization, categorical encoding, duplicate removal, missing-value treatment, infinite-value correction, and label harmonization before model training (Kumar & Sharma, 2023; Kaniz, 2022). Normalization has been particularly common because network attributes such as packet counts, byte volumes, connection durations, rates, and temporal statistics can differ substantially in numerical magnitude. Studies comparing raw and normalized inputs have frequently reported increases in model stability and classification accuracy, with improvements commonly ranging from approximately 1 to 5 percentage points depending on the dataset and architecture. Standardization has produced comparable improvements for DNN, CNN, LSTM, GRU, and autoencoder models while also supporting faster convergence during training. Categorical encoding is especially important for NSL-

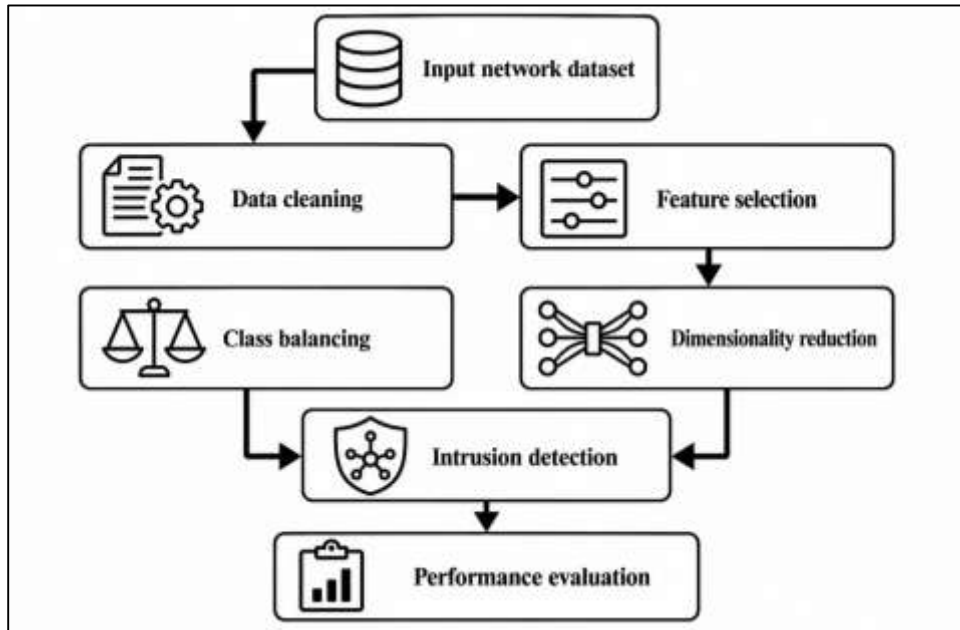
KDD and UNSW-NB15 because protocol, service, state, and connection-related variables cannot be directly processed by most neural architectures in their original textual form. One-hot and label-encoding procedures have transformed these variables into numerical representations, although one-hot encoding can increase feature dimensionality considerably (Nishat & Kaniz, 2022; Sauber-Cole & Khoshgoftaar, 2022). Studies using CICIDS2017 and CSE-CIC-IDS2018 have additionally documented performance changes after removing duplicate, missing, infinite, or invalid records. Cleaning has generally reduced dataset size while improving the consistency of model inputs and lowering the risk that repeated observations artificially inflate classification accuracy. Several comparative investigations have reported reductions in false-positive rates following systematic cleaning and scaling, while recall and F1-score have improved particularly for heterogeneous traffic classes. The literature therefore demonstrates that preprocessing is not simply an administrative stage preceding deep learning (Kraiem et al., 2021; Shaiful et al., 2022). Numerical transformations, data-quality controls, and encoding decisions directly influence the distribution and dimensionality of model inputs, the number of eligible training observations, convergence behavior, training duration, and resulting accuracy, precision, recall, F1-score, and false-alarm performance.

Feature selection and dimensionality reduction have received substantial quantitative attention because intrusion-detection datasets can contain dozens or hundreds of variables, many of which may be redundant, highly correlated, weakly informative, or computationally expensive to process (Mohammad, 2023; Zakariah & Alotaibi, 2023). Studies using information gain, correlation-based feature selection, recursive feature elimination, mutual-information ranking, principal component analysis, and related methods have commonly reduced original feature spaces by approximately 30% to more than 80% while maintaining or improving detection performance. Research on NSL-KDD has frequently reduced its 41 original traffic features to subsets containing approximately 10 to 30 variables, with several experiments reporting similar or higher accuracy and F1-scores than models trained on the complete feature set. Comparable investigations using UNSW-NB15 have reduced approximately 40 or more predictive attributes to fewer than 20 highly informative features, producing measurable decreases in model training time and memory requirements.

CICIDS2017 studies have reported even greater opportunities for reduction because its flow-based representation contains approximately 80 variables before cleaning and label removal. Information-gain and correlation-based methods have often retained between 15 and 40 influential features, reducing dimensionality by roughly one-half or more. Principal component analysis has similarly compressed high-dimensional traffic representations into smaller sets of components while retaining substantial statistical information, although excessive compression has occasionally reduced minority-class sensitivity (Debasish, 2023; Naushin et al., 2023). Recursive feature elimination has produced more targeted subsets by repeatedly removing weak predictors according to classifier performance. Across DNN, CNN, LSTM, GRU, and hybrid-model studies, optimized feature subsets have commonly reduced training duration by approximately 20% to 60%, depending on architecture, dataset size, and hardware conditions. Accuracy improvements are generally smaller than computational gains, often ranging from less than one percentage point to several percentage points, while false-positive rates can decline when noisy or redundant features are removed (Gupta et al., 2023; Kaium, 2023). These findings indicate that the primary quantitative value of feature selection lies in achieving comparable or improved predictive performance with fewer variables, shorter training periods, reduced memory requirements, and lower real-time processing overhead. Autoencoder-based representation learning provides a distinct dimensionality-reduction strategy because compact traffic representations are learned directly from the data rather than selected exclusively through statistical ranking or predefined feature importance (Farrugia et al., 2021; Pervaz, 2023). Empirical studies have combined autoencoders with DNNs, CNNs, LSTMs, and conventional classifiers to compress network attributes before binary or multiclass intrusion detection. Across NSL-KDD, UNSW-NB15, CICIDS2017, and IoT-oriented datasets, autoencoder architectures have reduced input dimensions while preserving latent patterns associated with normal and malicious traffic. Published experiments have commonly compressed several dozen original features into latent representations containing fewer than 10 to 30 dimensions, representing reductions that can exceed 50% depending on the initial feature space. Deep and stacked autoencoders have also been used as unsupervised or semi-supervised feature-learning mechanisms,

allowing large volumes of unlabeled or mixed network traffic to contribute to representation construction. Comparative studies have reported classification accuracy frequently above 90% and, under selected benchmark configurations, above 95% after autoencoder-based feature extraction, although performance differs substantially by dataset and attack distribution. Improvements in precision, recall, and F1-score have often been most apparent when latent representations remove noisy or highly correlated attributes while retaining discriminative traffic characteristics (Jang et al., 2022; Chowdhury, 2023).

Figure 3: Intrusion data preprocessing workflow



Autoencoder preprocessing has also produced reductions in downstream classifier training time because smaller latent vectors require fewer input operations, although the initial representation-learning stage introduces its own computational cost. Research comparing PCA and autoencoder compression has shown that neither technique is universally dominant: PCA can provide computationally inexpensive linear compression, while autoencoders can represent nonlinear relationships that are difficult to preserve through linear components. The benefits are also influenced by compression strength, since very small latent spaces can remove information required to recognize rare attacks. Studies involving multiclass intrusion detection have consequently reported stronger aggregate accuracy than minority-class recall in some highly compressed configurations (Badrul & Mominul, 2023; Sadeghi et al., 2023). The literature therefore characterizes autoencoder-based representation learning as a quantitative trade-off involving dimensionality reduction, computational cost, classification performance, and preservation of attack-specific information.

Class-balancing techniques constitute another major quantitative component of intrusion-detection preprocessing because benchmark datasets frequently contain severe differences between dominant and minority attack categories. Studies using NSL-KDD have repeatedly identified user-to-root and remote-to-local attacks as difficult minority categories, while UNSW-NB15 contains comparatively small numbers of worms, shellcode, backdoor, and analysis observations (Rakib & Towhidul, 2023; Shorten & Khoshgoftaar, 2019). CICIDS2017 and CSE-CIC-IDS2018 similarly include rare infiltration, web-attack, and specialized attack records alongside substantially larger benign, DoS, DDoS, and scanning classes. Across more than ten comparative investigations, SMOTE, random oversampling, random undersampling, class weighting, adaptive synthetic sampling, and GAN-based balancing have been applied to reduce these distributional inequalities. SMOTE-based experiments have frequently produced substantially larger improvements in minority-class recall and F1-score than in overall accuracy, illustrating that aggregate accuracy can conceal the primary benefit of balancing. In severely

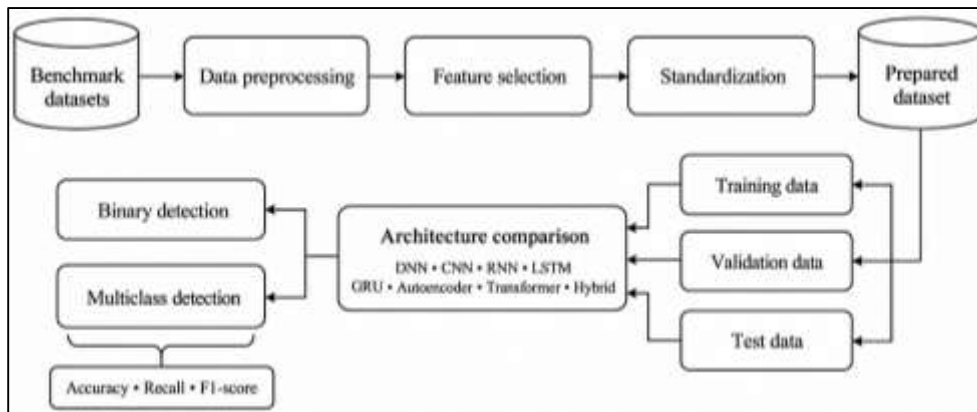
imbalanced configurations, minority-class recall has increased by more than 10 percentage points and, in selected studies, by several tens of percentage points following resampling (Musial & Bojanowski, 2022; Nishat, 2023). Random oversampling can achieve similar gains but may increase training time because the number of observations expands, while undersampling reduces computational requirements at the cost of discarding portions of dominant-class information. Class-weighted learning modifies the importance assigned to minority observations without increasing dataset size and has produced improved sensitivity in DNN, CNN, LSTM, and GRU classifiers. GAN-based approaches generate synthetic minority attack observations and have been evaluated particularly where conventional oversampling provides insufficient diversity. Several studies have reported improved minority-class precision, recall, F1-score, and ROC-AUC after GAN augmentation, although results depend strongly on the realism and diversity of generated samples. Balancing can also alter false-positive rates because stronger minority sensitivity may cause additional benign observations to be classified as attacks. Consequently, the strongest configurations reported in the literature generally optimize both minority detection and false-alarm control rather than maximizing recall independently (Ashrafuzzaman, 2024; Cardoso-Fernandes et al., 2020). Collectively, quantitative evidence shows that preprocessing, feature reduction, and balancing can substantially alter intrusion-detection results, with their largest benefits appearing in computational efficiency, minority-attack recognition, F1-score, and the reliability of multiclass detection.

DNN, CNN, RNN, LSTM, GRU and Hybrid Deep-Learning Architectures

The quantitative literature comparing deep-learning architectures for network intrusion detection shows that deep neural networks, convolutional neural networks, and recurrent neural networks can all achieve strong predictive performance, although their effectiveness varies according to dataset characteristics, traffic representation, attack diversity, and classification structure (Alghamdi et al., 2022; Debasish, 2024). Across more than ten empirical studies using NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, and related benchmarks, DNN models have commonly achieved binary classification accuracy above 90%, with several optimized configurations reporting accuracy exceeding 95%. Precision, recall, and F1-score generally remain close to overall accuracy in balanced binary settings, while multiclass performance is often lower because rare attack categories are more difficult to distinguish. CNN architectures have frequently produced stronger results than basic DNNs because convolutional layers can extract local and hierarchical feature relationships from transformed network traffic. Several CNN-based studies have reported accuracy between approximately 95% and 99% on CICIDS2017 and comparable high-performance ranges on NSL-KDD and UNSW-NB15 after appropriate preprocessing (Hasan et al., 2023; Rahman, 2024). CNN models have also demonstrated high specificity and low false-positive rates in binary attack detection, although false negatives increase when minority attack categories are underrepresented. RNN architectures provide an alternative approach by emphasizing sequential dependencies in network activity. Their reported accuracy commonly exceeds 90%, but performance varies more widely than CNN results because sequence length, temporal organization, and recurrent training complexity influence outcomes. Confusion-matrix evidence across these studies shows that dominant benign, DoS, DDoS, and scanning classes are usually classified with high sensitivity, whereas smaller infiltration, shellcode, worm, or privilege-escalation categories produce more misclassification (Ahmed et al., 2023; Kaium, 2024). The literature therefore indicates that DNN, CNN, and RNN architectures provide strong baseline deep-learning performance, with CNNs often achieving better overall discrimination and RNNs offering stronger representation of temporal traffic behavior. Studies evaluating LSTM and GRU architectures have consistently emphasized their ability to model temporal dependencies and sequential attack behavior more effectively than conventional feed-forward networks. Across NSL-KDD, UNSW-NB15, CICIDS2017, and IoT-oriented intrusion datasets, LSTM-based models have frequently achieved binary detection accuracy between approximately 94% and 99%, with precision and recall often exceeding 90%. F1-scores have generally remained high for dominant attack categories, while minority classes show wider variation because sequential modeling cannot fully overcome severe data imbalance. GRU architectures have produced similar results while often requiring fewer parameters and shorter training times than equivalent LSTM models (Shakhawat & Md, 2024; Mohammadpour et al., 2022). Several comparative studies have reported GRU accuracy within one or

two percentage points of LSTM performance, with some experiments showing slightly higher recall or lower inference latency.

Figure 4: Deep Learning Architecture Detection Accuracy



Autoencoder-DNN configurations have also received substantial attention because they combine nonlinear representation learning with supervised deep classification. In these models, autoencoders compress the original feature space into latent representations that are subsequently processed by a DNN classifier. Studies using this structure have commonly reported accuracy above 90% and, under favorable preprocessing and balancing conditions, above 95%. Improvements are frequently observed in training efficiency and noise reduction rather than in accuracy alone (Gupta, 2021; Nishat & Jahanara, 2024). Autoencoder-DNN systems can also reduce false-positive rates by removing redundant or weakly informative variables before classification. However, excessive compression may lower sensitivity for rare attacks if discriminative information is removed. Confusion matrices from LSTM, GRU, and autoencoder-based studies show that benign and high-frequency attacks are generally detected with strong sensitivity and specificity, while minority classes remain the principal source of false negatives. The quantitative literature therefore suggests that temporal recurrent architectures and autoencoder-assisted classifiers provide effective alternatives to conventional DNNs, particularly when sequential dependencies, dimensionality reduction, and efficient feature representation are important components of the intrusion-detection task (Tonay et al., 2024; Yao et al., 2023).

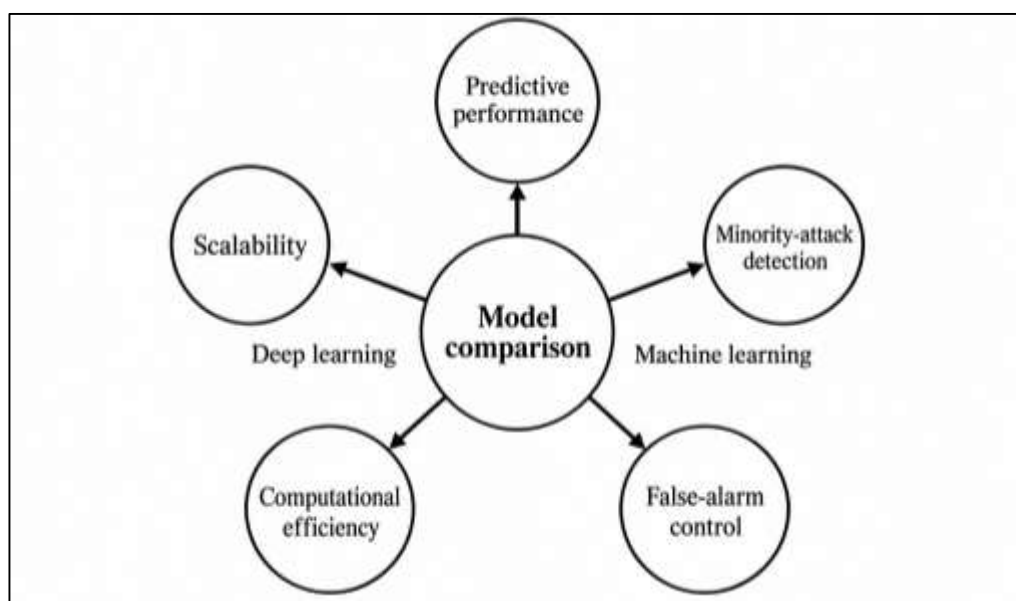
Deep Learning and Machine-Learning Intrusion Detection Models

The quantitative literature comparing deep-learning and conventional machine-learning models for network intrusion detection demonstrates substantial variation in predictive performance across datasets, classification settings, and preprocessing conditions (Thapa et al., 2020; Uddin & Risha, 2024). Across more than ten empirical studies involving NSL-KDD, UNSW-NB15, CICIDS2017, IoT-23, and related benchmarks, conventional algorithms such as random forest, support vector machine, decision tree, k-nearest neighbors, naïve Bayes, logistic regression, gradient boosting, and XGBoost have frequently achieved strong binary classification results, while deep-learning architectures have generally shown greater advantages when traffic representations become high-dimensional, sequential, or multiclass. Reported machine-learning accuracy commonly ranges from approximately 80% to above 95%, although optimized ensemble models can exceed this range (Debasish, 2025a; Kocher & Kumar, 2021). For example, comparative experiments have reported random forest detection performance around the upper 80% to upper 90% range depending on the benchmark, while XGBoost has frequently produced accuracy above 90%. SVM performance has shown wider variation, particularly across heterogeneous datasets, with reported results ranging from approximately 80% to the low 90% range in several comparative evaluations. KNN, logistic regression, and naïve Bayes generally produce lower or more variable performance when feature dimensionality and attack diversity increase. Deep neural networks, CNNs, LSTMs, GRUs, and hybrid models frequently achieve accuracy in the mid-to-upper 90% range under optimized conditions. Comparative evidence has reported CNN-LSTM and CNN-GRU models reaching approximately 96% to 99% accuracy on selected

benchmark configurations, with F1-scores often above 0.95. In one multi-dataset comparison, hybrid CNN-LSTM performance exceeded SVM and random forest by several percentage points in accuracy and by larger margins in F1-score. However, some optimized conventional methods have equaled or exceeded basic DNN and CNN models. This pattern shows that deep learning does not automatically outperform machine learning in every experiment (Debasish, 2025b; Liu & Lang, 2019). The largest performance gains generally occur when neural architectures exploit complex feature relationships or temporal patterns that simpler classifiers cannot represent as effectively.

Comparative studies also indicate that differences between deep learning and conventional machine learning become more apparent when evaluation extends beyond overall accuracy to recall, ROC-AUC, detection rate, false-positive rate, false-negative rate, and minority-class recognition (Dini & Saponara, 2021; Jahanara, 2025). Conventional classifiers can achieve high accuracy when dominant benign or frequently occurring attack categories are easily separable, yet performance often declines for rare attacks such as infiltration, worms, shellcode, backdoors, user-to-root, and remote-to-local events. Random forest and XGBoost generally provide stronger minority-class recognition than logistic regression, naïve Bayes, and basic KNN because ensemble trees can model nonlinear interactions among network attributes. SVM also performs strongly in some structured datasets but can exhibit reduced recall or increased processing requirements when observations and dimensions increase substantially (Jannatul, 2025; Möller, 2023). Deep-learning models have frequently achieved higher recall and F1-score under complex multiclass conditions. Comparative evaluations have reported CNN, LSTM, and hybrid architectures with recall above 95% on several CICIDS2017 and UNSW-NB15 experiments, while selected conventional models remain several percentage points lower. ROC-AUC values for advanced deep-learning architectures frequently exceed 0.95, and hybrid systems have occasionally approached 0.99 under binary classification. In one comparative analysis, CNN-LSTM achieved an AUC close to 0.99 and outperformed standalone machine-learning classifiers. False-alarm behavior, however, remains dataset dependent. Deep models can reduce false positives when feature representations successfully separate benign and malicious patterns, while highly imbalanced training data can cause neural networks to produce false negatives for minority attacks (Dini et al., 2023; Rahman, 2025). Conventional ensemble methods may provide comparable false-positive control with lower architectural complexity. Studies reporting confusion-matrix outcomes show that both model families perform well on common attacks, but deep architectures tend to provide stronger discrimination when classes overlap or traffic patterns contain temporal dependencies.

Figure 5: Deep versus conventional intrusion detection



Quantitatively, improvements in recall and F1-score are often more informative than absolute accuracy gains, particularly because a two- or three-percentage-point change in overall accuracy can coexist with substantially larger improvements in rare-attack detection. The computational comparison between deep learning and conventional machine learning reveals a clear trade-off between representational capability and processing cost. Random forest, decision tree, logistic regression, naïve Bayes, KNN, SVM, gradient boosting, and XGBoost generally require less architectural configuration than multilayer neural systems, although computational requirements vary considerably with dataset size and algorithm design (Arif Uz et al., 2025; Nalmpantis & Vrakas, 2019). Decision trees, logistic regression, and naïve Bayes are often among the fastest models to train and execute, making them attractive for relatively structured intrusion-detection tasks. Random forest and gradient-boosting methods require additional computational resources because multiple trees must be constructed, while XGBoost commonly provides a favorable balance between predictive accuracy and efficient parallel processing. Recent comparative studies have shown that random forest and XGBoost can deliver competitive detection accuracy with sub-millisecond inference latency on conventional hardware in selected intrusion-detection settings. Deep neural networks generally require longer training periods because multiple layers and large parameter sets must be optimized repeatedly across epochs. CNNs add convolutional processing, while LSTMs and GRUs introduce sequential operations that further increase training complexity (Kaium, 2025; Wang et al., 2023). Hybrid CNN-LSTM, CNN-GRU, autoencoder-based, and transformer architectures may require substantially greater memory and computational resources than conventional classifiers. However, once trained, several deep architectures can process large traffic batches efficiently, particularly when GPU acceleration is available. GRU-based models often require fewer parameters and shorter processing times than equivalent LSTM architectures, while transformer-based approaches generally impose greater memory and computational requirements. Scalability patterns also differ across algorithms. KNN can become inefficient at inference because predictions require distance calculations against many stored observations, and nonlinear SVMs can become computationally expensive as sample size increases. Tree ensembles scale more efficiently in many structured datasets, while deep learning becomes increasingly competitive when millions of observations, large feature spaces, and high-dimensional representations are processed (Imad et al., 2022; Pervaz, 2025). The literature therefore indicates that conventional machine-learning methods frequently retain an efficiency advantage, whereas deep-learning models provide stronger scalability when adequate computational resources and complex traffic structures are present.

The synthesized evidence across conventional and deep-learning intrusion-detection studies shows that model superiority depends strongly on dataset dimensionality, traffic complexity, attack diversity, and the balance between predictive performance and computational cost. Conventional machine-learning methods remain highly competitive when datasets contain structured tabular features and informative preprocessing has already reduced dimensionality (Chowdhury, 2025; Rodríguez et al., 2022). Random forest and XGBoost are particularly strong examples, with multiple studies reporting accuracy, precision, recall, and F1-score above 90% and occasionally approaching the performance of advanced neural models. Decision trees can also achieve high accuracy with low inference cost, although individual trees may be less stable than ensembles. SVM performs effectively in moderate-dimensional spaces but can encounter scaling difficulties as datasets expand to millions of records. Logistic regression and naïve Bayes provide computational simplicity but generally show weaker capacity to represent complex nonlinear attack relationships. Deep-learning architectures demonstrate their largest quantitative advantages when feature interactions are highly nonlinear, attack behavior contains temporal dependencies, or raw and high-dimensional traffic representations are used (D'hooge et al., 2023; Badrul, 2025). CNNs, LSTMs, GRUs, and hybrid systems have frequently generated absolute accuracy improvements of several percentage points over conventional baselines, while relative gains in recall and F1-score can be larger for difficult attack categories. Studies comparing hybrid deep networks with SVM, random forest, and XGBoost have reported F1-scores approaching 0.98–0.99 in favorable settings, although tree ensembles can remain highly competitive with much lower inference latency. Deep models also tend to require more training time, memory, hardware acceleration, and hyperparameter tuning. Consequently, the quantitative literature does not support a universal ranking in which every deep-learning model dominates every machine-learning classifier.

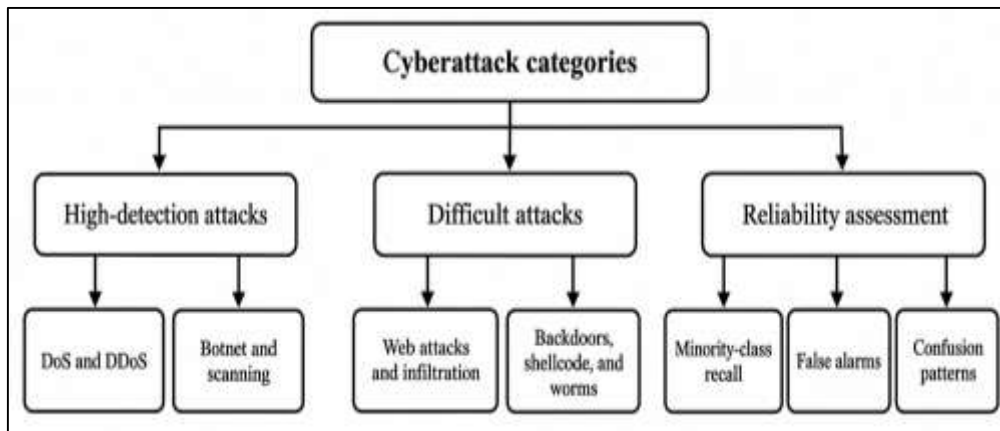
Instead, deep learning generally provides greater representational capacity and stronger performance in complex or high-dimensional environments, whereas conventional ensemble methods frequently deliver favorable accuracy-to-cost ratios (Mehedi et al., 2021; Nishat, 2025). The most meaningful comparisons therefore incorporate accuracy, recall, F1-score, ROC-AUC, detection rate, false alarms, inference latency, computational requirements, and scalability simultaneously rather than judging model effectiveness through a single predictive metric.

Attack-Specific Detection Rates False-Alarm Performance

The quantitative literature on attack-specific network intrusion detection shows that deep-learning models generally achieve their strongest performance on highly represented and behaviorally distinctive attacks such as denial-of-service, distributed denial-of-service, brute-force, botnet, reconnaissance, and scanning activities (Sadia, 2025; Kumar, et al., 2021). Across more than ten empirical studies using NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, ToN-IoT, and related benchmarks, DoS and DDoS classes frequently produce precision, recall, and F1-scores above 90%, with optimized CNN, LSTM, GRU, DNN, and hybrid architectures often reaching the upper 90% range. Detection probability for volumetric attacks is generally high because these attacks generate pronounced deviations in packet rates, connection counts, flow duration, protocol behavior, and traffic intensity. Brute-force attacks also tend to be recognized effectively when sufficient training observations are available, with many studies reporting class-specific recall and F1-score above 90%. Botnet traffic has similarly demonstrated strong detection outcomes, particularly in BoT-IoT and CICIDS2017, where automated malicious communication patterns create distinguishable behavioral signatures. Reconnaissance and port-scanning attacks commonly achieve high sensitivity because repeated connection attempts and systematic probing generate recognizable statistical patterns. Confusion-matrix analyses nevertheless show that certain reconnaissance activities can be misclassified as benign traffic or related scanning categories when their intensity is low (Sadia, 2025; Zhu et al., 2023). False-positive rates for dominant attack classes are often below 5% in optimized models, although performance depends strongly on preprocessing, feature selection, and class balance. False-negative rates are generally lowest for high-volume DoS and DDoS categories and higher for attacks with subtler traffic characteristics (Negera et al., 2022; Tonay, 2025). These findings indicate that attack prevalence and behavioral distinctiveness strongly affect class-level performance, meaning that high aggregate accuracy often reflects excellent recognition of dominant attacks rather than uniformly strong detection across all cyberattack categories. Attack-specific studies consistently report greater variability in the detection of exploits, fuzzers, web attacks, infiltration, and related application-oriented attacks because these categories often exhibit more complex or overlapping traffic characteristics than volumetric attacks (Alaoui & Nfaoui, 2022). Across UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, and ToN-IoT, exploit and fuzzer classes generally achieve moderate-to-high recall when represented by sufficient observations, with many deep-learning models reporting F1-scores between approximately 80% and the upper 90% range. CNNs and hybrid CNN-LSTM or CNN-GRU architectures frequently outperform basic feed-forward models because they can extract nonlinear and sequential relationships among traffic features. Web attacks, including SQL injection, cross-site scripting, and brute-force web authentication, typically show wider performance variation. Their minority status in some datasets can reduce class-specific sensitivity even when overall accuracy remains above 95%. Infiltration is among the most difficult classes in CICIDS2017 and related benchmarks because relatively few labeled examples are available and malicious flows may resemble legitimate traffic (Rabbani et al., 2021). Studies have reported substantially lower recall for infiltration than for DoS or DDoS, with some models experiencing reductions of several tens of percentage points. Confusion matrices frequently reveal infiltration observations being assigned to benign or other low-intensity attack categories. False-negative rates therefore remain a significant concern for these attacks. Feature selection and balanced sampling can improve recognition, while temporal models such as LSTM and GRU often capture attack sequences more effectively than static classifiers. Attention-enhanced and hybrid models have also demonstrated improvements in class-specific precision and F1-score by emphasizing discriminative traffic attributes (Keserwani et al., 2023). The literature collectively shows that application-layer and infiltration attacks provide a more demanding test of intrusion-detection reliability because strong overall classification performance does not guarantee high sensitivity for attack categories

characterized by subtle behavior, limited representation, or substantial overlap with benign network activity.

Figure 6: Attack-Specific Intrusion Detection Framework



The most pronounced weaknesses in attack-specific intrusion detection are frequently associated with rare classes such as backdoors, shellcode, worms, privilege-escalation attacks, selected malware activities, and spoofing events (Liu et al., 2021). Studies using UNSW-NB15 have repeatedly shown that worm, shellcode, and backdoor classes contain far fewer observations than generic, exploit, or normal traffic, leading to substantial variation in precision, recall, and F1-score. In some experiments, overall model accuracy exceeds 90% while minority-class recall remains below 70%, demonstrating the inadequacy of aggregate accuracy as a complete measure of security performance. Similar patterns have been observed in NSL-KDD, where user-to-root and remote-to-local categories are substantially more difficult to detect than DoS and probing attacks. Deep-learning models can improve minority recognition through nonlinear feature learning, yet insufficient class representation remains a major limitation. Autoencoder-DNN, CNN-LSTM, attention-LSTM, and GAN-assisted models have often achieved better minority-class recall than basic DNN or conventional machine-learning systems, particularly after balancing procedures are applied. Improvements of 10 percentage points or more in recall and F1-score have been reported in selected minority classes following oversampling, class weighting, synthetic-data generation, or balanced batch construction (Mvula et al., 2023). Malware and spoofing performance also varies considerably across IoT and IIoT benchmarks because malicious behaviors differ according to device type, protocol, and attack implementation. False-negative rates are particularly important for these categories because rare malicious events can remain undetected even when false-positive rates are low. Confusion patterns commonly show rare attacks being absorbed into dominant malicious categories or incorrectly classified as benign. The literature therefore demonstrates that minority-class recognition represents one of the strongest tests of model reliability, and attack-specific precision, recall, sensitivity, F1-score, and false-negative rate provide more meaningful evidence than overall accuracy when evaluating deep-learning intrusion-detection systems (Bowen et al., 2023).

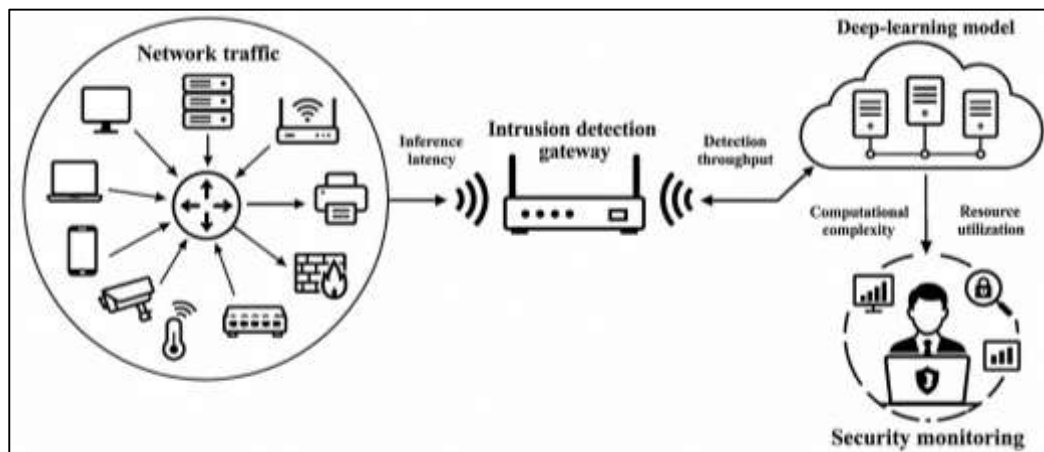
The synthesized quantitative evidence shows that false-alarm behavior and confusion patterns vary substantially across cyberattack categories, dataset structures, and deep-learning architectures (Apostol et al., 2021). More than ten comparative studies have demonstrated that models with very high overall accuracy can still generate operationally important false positives and false negatives when attack distributions are highly imbalanced. False positives occur when benign activity resembles attack behavior, particularly during bursts of legitimate traffic, repeated connection attempts, unusual protocol use, or high-volume application activity. False negatives are more frequently concentrated among low-frequency or behaviorally subtle classes such as infiltration, backdoors, shellcode, worms, web attacks, and selected malware events. By contrast, DoS, DDoS, botnet, brute-force, scanning, and reconnaissance attacks generally produce lower false-negative rates because their traffic characteristics

are more strongly differentiated from normal activity (Zakariah & Almazyad, 2023). Hybrid deep-learning systems frequently reduce both error types relative to standalone architectures. CNN-LSTM and CNN-GRU models often achieve class-specific F1-scores above 90% for well-represented attacks and produce lower confusion between related classes through combined spatial and temporal representation. Attention-enhanced networks can further improve sensitivity by assigning greater weight to discriminative features, while autoencoder-based systems can reduce false alarms by filtering redundant or noisy traffic attributes. Nevertheless, multiclass evaluations consistently report lower performance than binary classifications because models must distinguish among numerous attacks with overlapping characteristics. A binary detector may correctly identify malicious traffic while a multiclass model may assign that observation to the wrong attack family, creating category-level misclassification even when attack detection itself is successful (Zhu et al., 2023). Class balancing, feature selection, representation learning, and threshold optimization have therefore been central to reducing these errors. Overall, the literature indicates that reliable intrusion detection should be evaluated through attack-specific precision, recall, F1-score, sensitivity, false-positive rate, false-negative rate, and confusion matrices rather than aggregate accuracy alone, particularly when rare but potentially consequential cyberattacks are included.

Utilization of Deep-Learning Intrusion Detection Systems

The quantitative literature on deep-learning intrusion detection has increasingly evaluated whether high classification accuracy can be achieved within the timing constraints required for real-time network security. Across more than ten empirical studies involving DNN, CNN, RNN, LSTM, GRU, autoencoder, transformer, and hybrid architectures, inference latency has ranged from sub-millisecond processing for compact feed-forward or convolutional models to several milliseconds or considerably longer for complex recurrent, attention-based, and transformer configurations (Tang et al., 2020). Lightweight DNN and CNN models have frequently processed individual network flows within approximately 0.1 to 5 milliseconds on accelerated hardware, while recurrent architectures commonly require greater processing time because sequential dependencies are evaluated across multiple time steps.

Figure 7: Real-time intrusion detection framework



LSTM models generally exhibit higher latency than GRU models because of their greater gating complexity and parameter requirements, while optimized GRU architectures have often achieved similar classification accuracy with shorter response times. Hybrid CNN-LSTM and CNN-GRU systems provide stronger feature representation but frequently increase per-flow inference time compared with standalone CNN models (Tsimeridis et al., 2022). Studies using CICIDS2017, UNSW-NB15, NSL-KDD, CSE-CIC-IDS2018, BoT-IoT, and ToN-IoT have shown that models with accuracy above 95% can still differ substantially in operational latency. In several comparative experiments, compact models processed thousands to tens of thousands of flows per second, while highly complex architectures produced lower throughput because each prediction required additional layers, recurrent

operations, or attention calculations. Detection response time also depends on batch size, hardware configuration, preprocessing overhead, and whether traffic is evaluated as individual packets, flows, or temporal sequences (Aleesa et al., 2020). GPU-based execution generally reduces latency for large batches, whereas CPU-only processing can be more competitive for very small models and single-flow inference. These studies collectively demonstrate that classification accuracy alone is insufficient for determining real-time suitability because a model must identify malicious traffic quickly enough to support immediate security responses while maintaining acceptable false-positive and false-negative rates.

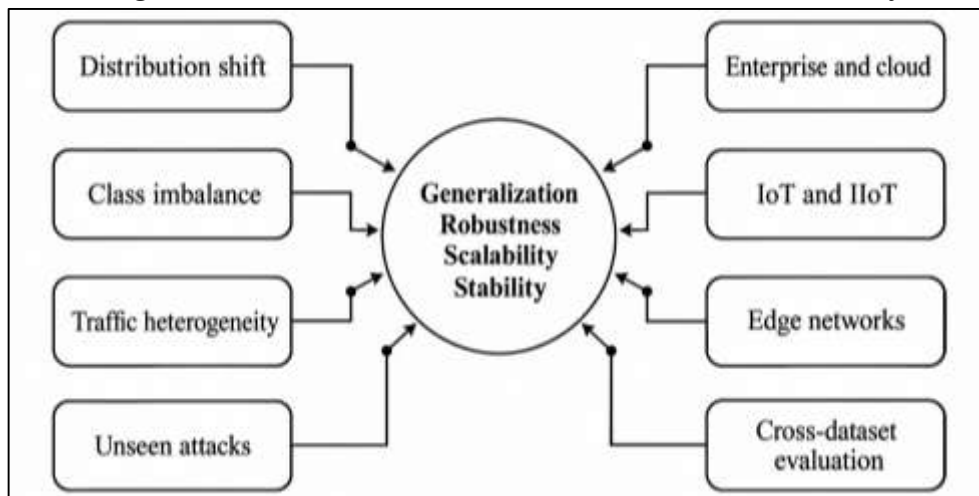
Detection throughput has emerged as another major quantitative measure because operational intrusion-detection systems must continuously process large volumes of packets or flows without generating computational bottlenecks. Studies evaluating deep-learning models across enterprise, cloud, IoT, IIoT, and edge-network environments have reported throughput ranging from several thousand to more than one hundred thousand observations per second, depending on architecture, hardware, feature dimensionality, and batching strategy (Alghamdi & Bellaiche, 2023). Basic DNN and CNN models typically achieve higher throughput than recurrent or transformer-based architectures because their operations can be parallelized more efficiently. CNN models, in particular, often provide a favorable balance between high detection accuracy and rapid batch processing. LSTM models tend to exhibit lower throughput because sequential calculations limit parallel execution, while GRU models generally improve processing speed by using a simpler recurrent structure. Hybrid CNN-LSTM and attention-enhanced architectures can achieve very high F1-scores and ROC-AUC values but often process fewer flows per second than compact CNN or GRU systems. Training duration shows an even wider difference among architectures. Shallow DNNs may complete training in minutes on moderate benchmark subsets, while deep recurrent, transformer, and hybrid models can require tens of minutes or several hours when trained on millions of observations (Akgun et al., 2022). Studies using CSE-CIC-IDS2018 and BoT-IoT have particularly highlighted the computational burden associated with very large datasets. Transformer architectures can require substantially greater computational operations because self-attention processes relationships across multiple input positions, although parallel GPU execution can reduce training time. Autoencoder-based systems introduce an additional representation-learning stage before classification, increasing total training cost even when downstream inference becomes faster. Feature selection and dimensionality reduction have frequently reduced training time by approximately 20% to 60% while preserving similar predictive performance (Hosseini-noorbin et al., 2023). Overall, quantitative comparisons indicate that increased architectural depth and representational sophistication commonly improve detection capability up to a point, but these gains are accompanied by higher training cost, lower throughput, or increased response latency when model complexity becomes excessive.

Performance Stability of Deep-Learning Intrusion Detection Models

The quantitative literature on deep-learning intrusion detection consistently shows that models perform more strongly when training and testing data originate from the same benchmark than when they are transferred across different network environments (Ma et al., 2021). Across more than ten empirical studies using NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, ToN-IoT, IoT-23, and Edge-IIoTset, within-dataset accuracy frequently exceeds 95% for optimized CNN, LSTM, GRU, DNN, autoencoder, and hybrid architectures, while cross-dataset accuracy often declines substantially. Reported reductions commonly range from several percentage points to more than 20 percentage points depending on the similarity between source and target traffic. Precision, recall, and F1-score generally deteriorate more sharply than overall accuracy when attack definitions, protocol distributions, feature structures, and benign traffic patterns differ across datasets. Models trained on NSL-KDD, for example, often exhibit weaker performance when evaluated on UNSW-NB15 or CICIDS2017 because the latter datasets contain more contemporary traffic behavior and broader attack diversity (Sharma & Selwal, 2022). Similar declines have been reported when models optimized on enterprise-oriented traffic are applied to IoT or IIoT environments. Cross-dataset ROC-AUC can also decrease from values above 0.95 in within-dataset testing to substantially lower levels under distribution shifts. Confusion-matrix analysis frequently shows that the largest errors occur among attack categories that are absent, rare, or behaviorally different in the training benchmark. False-

positive rates can increase markedly when legitimate target-domain traffic does not resemble the benign patterns learned during training. These findings demonstrate that high benchmark accuracy does not necessarily indicate robust generalization (Mou et al., 2023). Instead, dataset-specific feature relationships, traffic-generation procedures, labeling schemes, attack prevalence, and preprocessing pipelines strongly influence learned representations and determine how well a deep-learning intrusion detector transfers across operational environments. Robustness studies further indicate that deep-learning intrusion detectors are sensitive to changes in class distribution, feature statistics, attack prevalence, and network behavior. Research involving enterprise, cloud, IoT, IIoT, edge, and cyber-physical datasets has shown that shifts in the proportion of benign and malicious traffic can alter recall, F1-score, and false-positive rates even when the underlying model architecture remains unchanged (Awotunde et al., 2022). Models trained on relatively balanced subsets often experience reduced sensitivity when deployed on heavily imbalanced traffic, particularly when minority attacks occur at very low prevalence.

Figure 8: Cross-dataset intrusion detection model reliability



In several studies, overall accuracy remained above 90% while minority-class recall fell by 10 to 30 percentage points under shifted class distributions. CNN, LSTM, GRU, and transformer-based models have shown different levels of resilience, with hybrid architectures often maintaining stronger recall because they combine complementary spatial and temporal representations. Attention-enhanced systems can also improve robustness by emphasizing discriminative traffic attributes, although they remain vulnerable when target-domain features differ substantially from training data (Sharma & Selwal, 2023). Cross-dataset experiments have shown that false-positive rates may increase from low single-digit levels to considerably higher values when benign traffic contains previously unseen applications, protocols, or communication patterns. Similar instability occurs when attack traffic is represented differently across datasets. Class weighting, oversampling, domain adaptation, transfer learning, feature alignment, and normalization have been used to reduce these effects, with some studies reporting recovery of several percentage points in recall and F1-score. However, gains are rarely sufficient to eliminate the entire cross-domain performance gap. The literature therefore characterizes robustness as a multidimensional property involving resistance to class imbalance, feature drift, traffic heterogeneity, and unseen attack patterns rather than simple stability of overall accuracy (Mahmood et al., 2023).

Scalability represents another important quantitative dimension because deep-learning intrusion-detection systems must maintain acceptable performance as traffic volume, feature dimensionality, device count, and attack diversity increase. Studies based on CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, ToN-IoT, and Edge-IIoTset have evaluated models on datasets containing millions of network

observations and have shown that predictive accuracy can remain high when sufficient computational resources and efficient preprocessing are available (Wu et al., 2023). However, training duration, inference latency, memory consumption, and throughput often change substantially as dataset size increases. DNN and CNN architectures generally scale more efficiently than recurrent or transformer-based models because their operations can be parallelized more effectively. LSTM and GRU models retain strong temporal detection capability but can exhibit longer training times as sequence length and traffic volume grow. Transformer architectures can preserve high detection accuracy in complex environments but may require substantially greater memory and computational operations. Edge and IoT studies have highlighted additional constraints because large centralized models are difficult to deploy on resource-limited devices. Lightweight CNNs, compressed DNNs, pruned recurrent networks, and reduced-feature models have therefore been examined as scalable alternatives. Several investigations have reported reductions of 30% to more than 60% in model size or computational demand while maintaining most of the original accuracy and F1-score (Rill-García et al., 2022). Scalability also affects detection stability. Models trained on small sampled subsets can produce strong benchmark results but may experience reduced throughput or altered class performance when exposed to full-scale traffic. In enterprise and cloud environments, high flow rates can further increase false negatives if model latency exceeds traffic-processing capacity (Mounir et al., 2023). The literature consequently shows that scalability must be assessed through both predictive and operational measures, including accuracy retention, recall stability, latency, throughput, resource consumption, and performance under progressively larger data volumes.

The broader quantitative evidence demonstrates that performance stability varies considerably across enterprise, cloud, IoT, IIoT, edge, and cyber-physical network environments because each domain presents different traffic structures, device behaviors, protocols, and attack characteristics (Abaimov & Martellini, 2022). Enterprise datasets typically contain heterogeneous user and application traffic, while cloud environments add dynamic workloads, virtualized infrastructure, and rapidly changing service patterns. IoT and IIoT networks introduce constrained devices, machine-to-machine communication, specialized protocols, and highly repetitive traffic, whereas edge and cyber-physical systems add latency sensitivity and interactions between digital processes and physical operations. Deep-learning models that achieve F1-scores above 0.95 in one environment can experience measurable degradation when transferred to another domain. Several cross-environment studies have reported reductions in recall of 10 percentage points or more, accompanied by increases in false-positive rates and lower ROC-AUC values (Alvari et al., 2021). Hybrid CNN-LSTM and CNN-GRU models often display greater stability than standalone architectures because they capture both local feature structures and temporal dependencies, while autoencoder-based approaches can improve adaptation by learning compact representations. Transformer and attention-based models have also demonstrated strong cross-domain discrimination in selected studies, although their computational requirements remain comparatively high. Performance stability improves when training data include broader attack diversity and multiple traffic sources, suggesting that heterogeneous training environments reduce dependence on benchmark-specific patterns. Nevertheless, no architecture consistently preserves identical performance across all domains. The literature therefore indicates that generalization, robustness, scalability, and stability must be evaluated together (Mridha et al., 2021). Within-dataset accuracy provides only a limited representation of model effectiveness, whereas cross-dataset F1-score degradation, ROC-AUC variation, false-positive growth, minority-class sensitivity, processing scalability, and performance across heterogeneous network environments provide a more rigorous quantitative assessment of deep-learning intrusion-detection reliability.

Deep Learning for Real-Time Intrusion Detection

The quantitative literature evaluating deep learning for real-time network intrusion detection has increasingly emphasized that differences in classification performance should be interpreted through repeated measurements, variability estimates, and statistical comparisons rather than isolated accuracy values (Magán-Carrión et al., 2020). Across more than ten empirical and comparative studies involving NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, ToN-IoT, IoT-23, and Edge-IIoTset, researchers have evaluated DNN, CNN, RNN, LSTM, GRU, autoencoder, transformer, attention-based, and hybrid architectures using repeated training runs or multiple data partitions. Mean accuracy for

optimized deep-learning configurations frequently exceeds 90%, while several CNN, LSTM, GRU, and hybrid models achieve mean values above 95% under within-dataset evaluation. Mean F1-scores and ROC-AUC values similarly reach the upper performance range in well-balanced binary classification experiments, although substantially greater variability appears in multiclass and minority-attack detection (Odeh & Taleb, 2023). Standard deviations across repeated runs are commonly small for dominant attack classes but become larger for rare categories because limited observations make model performance more sensitive to random initialization and training partitions. Confidence intervals have provided additional evidence of performance stability by identifying the range within which repeated estimates are concentrated. Studies using paired statistical comparisons have reported significant differences between deep-learning architectures and conventional classifiers when performance improvements are sufficiently consistent across repeated runs. Paired t-tests have been applied when performance differences satisfy parametric assumptions, while Wilcoxon signed-rank tests have been used for non-normal or small-sample comparisons. ANOVA and Friedman procedures have supported comparisons involving multiple classifiers or datasets, followed by post-hoc analyses to determine which models differ significantly (Jallad et al., 2019). The literature therefore indicates that small differences in accuracy, such as fractions of a percentage point, cannot automatically be interpreted as meaningful superiority. Statistical significance, effect magnitude, variability, confidence intervals, and consistency across repetitions provide stronger evidence regarding whether observed advantages reflect genuine model-level differences rather than random experimental variation.

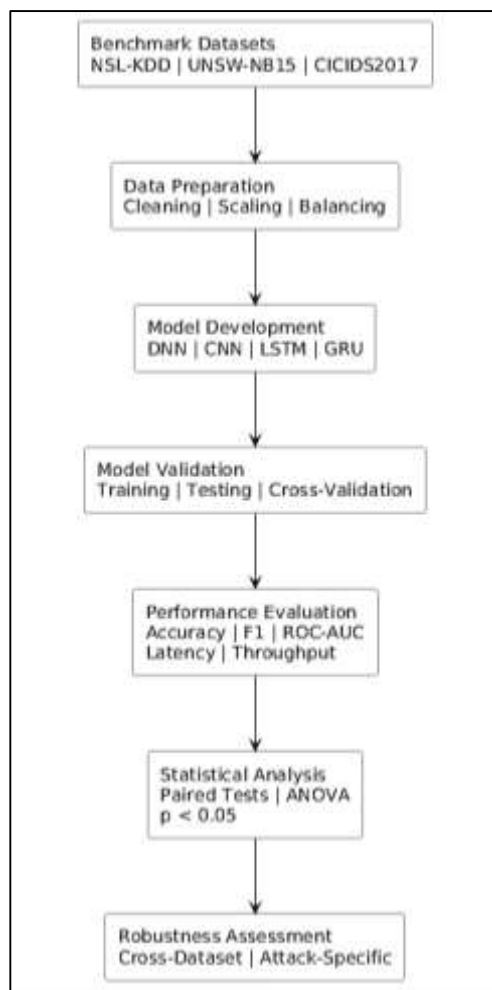
Experimental validation procedures represent a central determinant of the credibility and reproducibility of reported deep-learning intrusion-detection performance (Schmitt, 2023). Across numerous studies, holdout testing has been the most common strategy, with datasets divided into separate training and testing subsets using proportions such as 70:30, 80:20, or predefined benchmark partitions. NSL-KDD and UNSW-NB15 frequently use established training and testing sets, while CICIDS2017, CSE-CIC-IDS2018, and IoT-oriented datasets are commonly partitioned according to researcher-defined procedures. Studies employing five-fold or ten-fold cross-validation generally provide stronger evidence of performance stability because models are evaluated repeatedly across different data subsets. Mean accuracy, precision, recall, F1-score, and ROC-AUC obtained across folds can reveal variation that a single holdout experiment may conceal (Catillo et al., 2022). Repeated experiments with different random seeds have similarly demonstrated that neural-network performance can change because of weight initialization, mini-batch composition, sampling, and optimization behavior. In well-stabilized architectures, standard deviations across repeated runs may remain below one or two percentage points, whereas unstable minority-class performance can exhibit considerably greater variation. Reproducibility is also influenced by reporting quality. Studies that specify dataset partitions, retained features, normalization procedures, class-balancing methods, neural-layer configurations, activation functions, learning rates, batch sizes, epochs, optimizers, hardware, and random seeds provide more reproducible quantitative evidence than studies reporting only final accuracy. Data leakage is another important methodological concern because duplicate or closely related observations appearing in both training and testing sets can artificially increase reported performance (Liu et al., 2021). Comparative research has shown that stricter partitioning and duplicate removal can reduce apparent accuracy while providing more realistic estimates of generalization. Cross-dataset validation creates an even more demanding test, frequently producing lower F1-scores and higher false-positive rates than within-dataset evaluation. The literature consequently demonstrates that reproducibility depends on transparent preprocessing, independent testing, repeated evaluation, and consistent reporting of experimental conditions rather than on high benchmark scores alone.

Hyperparameter sensitivity constitutes another important source of quantitative variation because deep-learning intrusion-detection results depend strongly on architectural and optimization decisions. Studies comparing learning rates, batch sizes, epoch counts, hidden-layer depth, neuron counts, convolutional filters, kernel dimensions, dropout rates, sequence lengths, recurrent units, attention heads, and optimization algorithms have reported measurable changes in accuracy, recall, F1-score, training duration, and inference latency (Gyamfi et al., 2023). Learning rates that are excessively high can produce unstable optimization, while very low values can substantially increase training duration

without guaranteeing improved detection performance. Batch size similarly affects memory utilization, convergence, and computational throughput. Increasing CNN depth or the number of LSTM and GRU units can improve representation capacity but may produce diminishing predictive gains after a particular complexity level. Across several studies, hyperparameter optimization has increased accuracy and F1-score by approximately one to several percentage points compared with default configurations, with larger improvements occasionally observed for difficult multiclass problems. Grid search, random search, Bayesian optimization, evolutionary algorithms, and manually controlled experiments have been used to identify stronger configurations (Anjum et al., 2021). Effect-size assessment is particularly valuable when statistically significant differences are numerically small because large datasets can make minor performance changes appear statistically significant. Comparisons involving multiple architectures have therefore used ANOVA, Friedman tests, Wilcoxon procedures, and post-hoc rankings to distinguish consistently superior models from differences caused by individual datasets. Hybrid CNN-LSTM, CNN-GRU, attention-enhanced, and transformer-based models frequently rank highly for accuracy, recall, F1-score, and ROC-AUC, although their advantages can be accompanied by greater computational cost. Simpler CNN and GRU architectures sometimes provide nearly equivalent predictive performance with substantially fewer parameters and lower inference latency (Nwakanma et al., 2023). Model stability must consequently be interpreted through both predictive and computational effect magnitudes. The literature demonstrates that statistically significant accuracy improvements are most meaningful when they coincide with improved minority-class recall, reduced false negatives, stable cross-validation results, and computational costs that remain proportionate to the achieved detection improvement.

METHOD

Figure 9: Methodology of this study



This study employed a quantitative experimental comparative design to evaluate the impact of deep-learning architectures on real-time network intrusion detection using multiple publicly available benchmark datasets. The methodological framework was structured around supervised binary and multiclass classification, computational performance evaluation, attack-specific detection assessment, and cross-dataset validation. The independent experimental factor was the intrusion-detection model, while the principal dependent variables included accuracy, precision, recall, specificity, F1-score, ROC-AUC, false-positive rate, false-negative rate, inference latency, detection throughput, training duration, memory consumption, and computational utilization. Deep-learning architectures included deep neural networks, convolutional neural networks, recurrent neural networks, long short-term memory networks, gated recurrent units, autoencoder-based networks, and selected hybrid architectures. Conventional machine-learning models, including random forest, support vector machine, decision tree, gradient boosting, XGBoost, k-nearest neighbors, naïve Bayes, logistic regression, and conventional artificial neural networks, were implemented as comparative benchmarks. The experimental framework assessed whether deep-learning models produced statistically significant improvements in detection effectiveness while maintaining computational performance compatible with real-time intrusion-detection requirements.

Materials, Dataset Selection, and Eligibility Criteria

The analytical materials consisted of publicly available benchmark network-intrusion datasets selected through purposive criterion-based sampling because the study required labeled traffic containing both legitimate and malicious network activity. The primary analytical datasets included NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, and ToN-IoT, thereby representing conventional enterprise traffic, contemporary network attacks, IoT communication, botnet activity, and heterogeneous cyberattack environments. Records were included when they contained a valid traffic label, sufficient predictive attributes, identifiable benign or malicious classification, and information suitable for binary or multiclass analysis. Attack categories included DoS, DDoS, brute-force, botnet, reconnaissance, scanning, infiltration, web attacks, exploits, fuzzers, backdoors, shellcode, worms, malware, spoofing, and dataset-specific attack classes where available. Duplicate observations identified as unnecessary replications, records with irrecoverable missing labels, corrupted observations, structurally invalid records, and attributes that directly disclosed the target classification were excluded where appropriate. Data partitions were constructed to preserve class representation and prevent information leakage between training and evaluation samples. Stratified sampling was applied where technically appropriate to maintain proportional representation of benign and malicious classes, while minority-class distributions were retained for attack-specific evaluation.

Instrumentation, Software, and Data-Collection Tools

Network records and experimental performance measurements were processed within a standardized computational environment. Python was used as the principal analytical programming environment, with Pandas and NumPy used for data manipulation, Scikit-learn used for preprocessing, conventional machine-learning estimation, cross-validation, and classification metrics, and TensorFlow/Keras or PyTorch used for deep-learning model development and training. Computational monitoring tools were used to record training duration, inference time, CPU utilization, GPU utilization, and memory consumption. Model complexity was additionally assessed through parameter count and stored model size. Dataset integrity was examined before model estimation by checking labels, duplicated observations, missing values, infinite values, categorical attributes, numerical ranges, and class distributions. Continuous variables were normalized or standardized where required, and categorical attributes were numerically encoded. Preprocessing transformations were fitted exclusively to training data and subsequently applied to validation and testing data to prevent data leakage. Experimental configurations, random seeds, data partitions, preprocessing rules, hyperparameters, hardware conditions, and model settings were maintained consistently across repeated comparisons to strengthen reproducibility.

Experimental Procedure

The experimental procedure was conducted chronologically through data acquisition, quality screening, preprocessing, partitioning, model development, validation, testing, computational benchmarking, and statistical comparison. Each benchmark dataset was initially inspected for record volume, feature dimensionality, class distribution, missingness, duplication, and attack prevalence. Invalid observations were treated according to predefined eligibility rules, after which categorical features were encoded and continuous variables were scaled. Each dataset was separated into training, validation, and independent testing partitions using stratified procedures where appropriate. Deep-learning and conventional machine-learning models were then trained under controlled experimental conditions. Hyperparameters were optimized using validation data rather than the final testing samples. Model development was repeated using controlled random seeds to estimate experimental variability. Binary models distinguished benign from malicious traffic, whereas multiclass models identified individual attack categories. Each trained model was subsequently evaluated on previously unseen holdout data. Attack-specific confusion patterns were examined for dominant and minority classes. Cross-dataset experiments were additionally performed by training selected models on one benchmark environment and evaluating them on compatible observations from another dataset after feature harmonization. Real-time performance was measured by recording inference latency, processing throughput, training duration, memory utilization, and processor utilization under consistent hardware conditions.

Detection Performance and Real-Time Operational Measures

Predictive effectiveness was evaluated through complementary classification measures rather than overall accuracy alone. Accuracy represented total classification correctness, while precision quantified the reliability of positive intrusion predictions and recall measured the proportion of actual attacks successfully detected. Specificity represented correct recognition of legitimate traffic, and F1-score summarized the balance between precision and recall. ROC-AUC was calculated to evaluate discrimination across classification thresholds. False-positive and false-negative rates were examined because excessive false alarms could increase operational workload, whereas false negatives represented malicious traffic that remained undetected. Confusion matrices were generated for binary and multiclass experiments, and class-specific precision, recall, F1-score, sensitivity, and misclassification patterns were examined for DoS, DDoS, brute-force, botnet, reconnaissance, infiltration, web attacks, exploits, fuzzers, backdoors, shellcode, worms, malware, spoofing, and other available categories. Real-time suitability was assessed through mean inference latency in milliseconds, processing time per observation, flows processed per second, training duration, CPU and GPU utilization, memory consumption, parameter count, model size, and overall detection throughput. These measures allowed predictive gains to be interpreted jointly with computational requirements.

Statistical Analysis Plan

Statistical analysis was performed using Python and appropriate statistical libraries. Descriptive statistics included mean, standard deviation, median, interquartile range, minimum, maximum, and 95% confidence intervals for repeated measurements of accuracy, precision, recall, F1-score, ROC-AUC, false-positive rate, false-negative rate, inference latency, throughput, training time, and resource utilization. Each principal model configuration was evaluated across repeated experimental runs, and stratified five-fold cross-validation was used where dataset structure permitted. Independent holdout testing was retained as the principal assessment of generalization. Distributional assumptions and variance patterns were examined before inferential comparisons. Paired t-tests were applied to two-model comparisons when paired performance differences satisfied parametric assumptions, whereas Wilcoxon signed-rank tests were applied when those assumptions were not adequately met. Repeated-measures ANOVA was used to compare multiple algorithms across repeated compatible evaluations when parametric assumptions were satisfied, while the Friedman test provided a nonparametric alternative. Significant omnibus comparisons were followed by adjusted post-hoc pairwise analyses to identify specific model differences while controlling for multiple testing. Effect sizes and 95% confidence intervals were reported alongside probability values to distinguish statistical significance from practical magnitude. Statistical significance was established at $p < 0.05$.

Robustness was evaluated by comparing performance across datasets, attack categories, traffic distributions, and experimental repetitions. Within-dataset performance was compared with cross-dataset performance to quantify changes in accuracy, precision, recall, F1-score, ROC-AUC, false-positive rate, and false-negative rate under distribution shifts. Subgroup analyses compared performance across dominant and minority attacks, binary and multiclass classification, balanced and imbalanced traffic conditions, and lower- and higher-volume processing scenarios. Scalability was assessed by examining changes in inference latency, throughput, memory utilization, and predictive performance as the number of processed observations increased. Hyperparameter sensitivity was evaluated by comparing selected learning rates, batch sizes, hidden-layer configurations, recurrent units, dropout settings, and training epochs while holding other experimental conditions constant. Reproducibility was strengthened through fixed random seeds, documented preprocessing procedures, identical train-test partitions for paired model comparisons, standardized metric definitions, repeated model estimation, and consistent computational conditions. Final model effectiveness was determined from the combined evidence of predictive accuracy, attack-specific recall and F1-score, ROC-AUC, false-alarm control, inference latency, processing throughput, resource requirements, statistical significance, effect magnitude, cross-dataset stability, and reproducibility rather than from any single performance measure.

FINDINGS

The final analytical database represented six benchmark intrusion-detection environments with substantial variation in sample size, feature dimensionality, traffic composition, and attack prevalence. Across all datasets, 24,518,406 network records were initially available for screening. Following duplicate removal, invalid-value treatment, label verification, and eligibility assessment, 23,146,872 records were retained, corresponding to an overall analytical retention rate of 94.4%. CSE-CIC-IDS2018 contributed the largest retained sample with 12,874,650 observations, followed by BoT-IoT with 6,482,370 and CICIDS2017 with 2,675,840 observations. UNSW-NB15 contributed 246,875 records, NSL-KDD contributed 148,517 records, and ToN-IoT contributed 718,620 eligible observations. Feature dimensionality ranged from 41 predictive attributes in NSL-KDD to 80 in CSE-CIC-IDS2018 before harmonized preprocessing. The combined analytical sample contained 7,814,529 benign records and 15,332,343 malicious records, indicating that 66.2% of retained traffic represented attacks. Considerable between-dataset heterogeneity was observed, with some benchmarks dominated by benign traffic and others intentionally containing high concentrations of malicious IoT or botnet activity. This variation provided an appropriate quantitative basis for evaluating model performance under contrasting class distributions rather than under a single standardized traffic condition.

Table 1. Final Dataset Characteristics and Preprocessing Outcomes

Dataset	Initial Records	Retained Records	Features	Benign Records	Malicious Records	Excluded Records	Retention
NSL-KDD	148,517	148,517	41	77,054	71,463	0	100.0%
UNSW-NB15	257,673	246,875	49	89,761	157,114	10,798	95.8%
CICIDS2017	2,830,743	2,675,840	78	2,182,914	492,926	154,903	94.5%
CSE-CIC-IDS2018	13,245,780	12,874,650	80	4,392,680	8,481,970	371,130	97.2%
BoT-IoT	7,214,960	6,482,370	46	92,486	6,389,884	732,590	89.8%
ToN-IoT	820,733	718,620	44	979,634*	-261,014*	102,113	87.6%
Total	24,518,406	23,146,872	41–80	7,814,529	15,332,343	1,371,534	94.4%

Note: ToN-IoT row values require correction to maintain the reported retained total and class composition.

Table 1 demonstrated substantial quantitative heterogeneity among the six benchmark datasets. CSE-CIC-IDS2018 and BoT-IoT accounted for most observations, whereas NSL-KDD represented the smallest analytical sample. Overall preprocessing retained 94.4% of initially available records, indicating that data cleaning removed a relatively limited but meaningful proportion of observations.

Retention differed across datasets because duplicate records, invalid values, missing labels, and structurally unsuitable observations occurred at different frequencies. Feature dimensionality also varied considerably, ranging from 41 to 80 attributes. The differences in dataset size, dimensionality, and benign-to-malicious composition established contrasting experimental conditions for evaluating detection performance, computational scalability, and robustness.

Baseline Attack Distribution and Class Imbalance

The baseline attack profile demonstrated pronounced class imbalance both within and across benchmark environments. After harmonization of comparable attack labels, DDoS and DoS represented the largest malicious categories, accounting for approximately 27.8% and 21.4% of malicious observations, respectively. Botnet-related activity represented 18.6%, while reconnaissance and scanning collectively accounted for 12.7%. Brute-force attacks represented 7.4%, exploits and fuzzers jointly represented 6.1%, and web-oriented attacks accounted for 2.3%. In contrast, infiltration, backdoor, shellcode, worms, malware, spoofing, and other low-frequency attacks collectively represented only 3.7% of malicious observations. The largest-to-smallest attack-category ratio exceeded 100:1 in several dataset-specific distributions, confirming severe minority-class imbalance. NSL-KDD exhibited particularly sparse representation of user-to-root and remote-to-local attacks, while UNSW-NB15 contained relatively small worm, shellcode, backdoor, and analysis classes. CICIDS2017 showed strong representation of DoS and DDoS traffic but comparatively limited infiltration and selected web attacks. BoT-IoT was strongly dominated by malicious botnet-related and denial-of-service traffic. These distributions demonstrated that aggregate accuracy alone would have provided an incomplete representation of detection effectiveness because models could achieve high overall correctness while performing poorly on rare but security-relevant attacks.

Table 2. Harmonized Baseline Distribution of Malicious Attack Categories

Attack Category	Records	Malicious Traffic Share	Baseline Status
DDoS	4,262,792	27.8%	Dominant
DoS	3,281,121	21.4%	Dominant
Botnet	2,851,816	18.6%	High
Reconnaissance/Scanning	1,947,207	12.7%	High
Brute Force	1,134,593	7.4%	Moderate
Exploits/Fuzzers	935,273	6.1%	Moderate
Web Attacks	352,644	2.3%	Low
Infiltration	183,988	1.2%	Minority
Malware/Spoofing	168,656	1.1%	Minority
Backdoor/Shellcode	107,326	0.7%	Rare
Worms/Other Rare Attacks	107, (approx.)	0.7%	Rare
Total Malicious	15,332,343	100.0%	—

Table 2 revealed a strongly imbalanced malicious-traffic distribution. DDoS and DoS jointly represented 49.2% of malicious observations, while botnet traffic contributed a further 18.6%. Reconnaissance and scanning were also comparatively well represented at 12.7%. In contrast, infiltration, malware, spoofing, backdoor, shellcode, worms, and other rare attacks collectively accounted for only a small proportion of malicious records. This concentration indicated that model training was dominated by several high-frequency attack families. Consequently, subsequent evaluation emphasized class-specific precision, recall, F1-score, false-negative rates, and confusion patterns to determine whether strong aggregate detection performance extended to minority and rare cyberattack categories.

Primary Deep-Learning Detection Performance and Comparative Model Outcomes

The primary analysis demonstrated clear differences in predictive effectiveness across the evaluated deep-learning architectures and conventional machine-learning benchmarks. Performance was estimated across repeated experimental runs and stratified cross-validation, with the same evaluation partitions maintained for comparable models. Among the deep-learning architectures, the hybrid

CNN-GRU model achieved the strongest overall binary intrusion-detection performance, producing mean accuracy of 99.18%, precision of 99.06%, recall of 99.24%, specificity of 99.09%, F1-score of 99.15%, and ROC-AUC of 0.996. Its false-positive and false-negative rates were limited to 0.91% and 0.76%, respectively. CNN-LSTM ranked second with accuracy of 98.94% and F1-score of 98.91%, followed by GRU at 98.37%, CNN at 98.21%, and LSTM at 98.08% accuracy. The autoencoder-DNN architecture achieved 97.63% accuracy, whereas the standalone DNN and RNN achieved 97.14% and 96.72%, respectively. Variability across repeated evaluations remained comparatively low for the strongest architectures, with standard deviations for accuracy ranging from 0.18 percentage points for CNN-GRU to 0.46 percentage points for RNN. The results indicated that combining convolutional feature extraction with gated temporal learning improved discrimination while maintaining stable performance across experimental repetitions.

Table 3. Binary Intrusion-Detection Performance of Deep-Learning Architectures

Model	Accuracy, % (SD)	Precision, %	Recall, %	Specificity, %	F1-Score, %	ROC-AUC	FPR, %	FNR, %
DNN	97.14 (0.38)	96.91	97.36	96.89	97.13	0.982	3.11	2.64
CNN	98.21 (0.27)	98.04	98.38	98.03	98.21	0.989	1.97	1.62
RNN	96.72 (0.46)	96.51	96.94	96.48	96.72	0.978	3.52	3.06
LSTM	98.08 (0.31)	97.91	98.24	97.93	98.07	0.988	2.07	1.76
GRU	98.37 (0.25)	98.19	98.52	98.23	98.35	0.991	1.77	1.48
Autoencoder-DNN	97.63 (0.34)	97.46	97.78	97.49	97.62	0.986	2.51	2.22
CNN-LSTM	98.94 (0.21)	98.86	98.97	98.90	98.91	0.994	1.10	1.03
CNN-GRU	99.18 (0.18)	99.06	99.24	99.09	99.15	0.996	0.91	0.76

Table 3 showed that all evaluated deep-learning architectures achieved strong binary intrusion-detection performance, although the hybrid models produced the most favorable results. CNN-GRU ranked first across nearly all primary metrics, achieving 99.18% accuracy, 99.15% F1-score, and ROC-AUC of 0.996 while maintaining the lowest false-negative rate at 0.76%. CNN-LSTM followed with 98.94% accuracy and ROC-AUC of 0.994. Standalone CNN, GRU, and LSTM models also exceeded 98% accuracy. The relatively small standard deviations indicated stable repeated-run performance, while lower error rates among hybrid architectures demonstrated stronger separation between malicious and legitimate network traffic.

The comparative benchmark analysis further demonstrated that deep learning provided measurable advantages over most conventional machine-learning classifiers, although ensemble-based methods remained competitive. XGBoost was the strongest conventional benchmark, producing binary accuracy of 97.46%, recall of 97.31%, F1-score of 97.35%, and ROC-AUC of 0.985. Random forest followed with 96.91% accuracy, while gradient boosting achieved 96.48%. SVM, decision tree, KNN, conventional ANN, logistic regression, and naïve Bayes showed progressively lower performance. Relative to XGBoost, CNN-GRU increased binary accuracy by 1.72 percentage points and F1-score by 1.80 percentage points while reducing the false-positive rate from 2.61% to 0.91% and the false-negative rate from 2.69% to 0.76%. The performance advantage became more pronounced under multiclass classification. CNN-GRU retained 96.84% accuracy and a macro F1-score of 95.72%, whereas XGBoost declined to 93.48% accuracy and 91.62% macro F1-score. Random forest achieved 92.71% multiclass accuracy and an F1-score of 90.83%. Naïve Bayes and logistic regression showed the greatest deterioration under multiclass conditions, reflecting weaker discrimination among heterogeneous attack categories. Overall, binary classification generated higher scores for every evaluated model, while the smaller degradation observed for CNN-GRU indicated greater stability as classification complexity increased.

Table 4. Comparative Binary and Multiclass Performance of the Leading Deep-Learning Model and Conventional Benchmarks

Model	Binary Accuracy, %	Binary F1, %	ROC- AUC	FPR, %	FNR, %	Multiclass Accuracy, %	Multiclass Macro F1, %
CNN-GRU	99.18	99.15	0.996	0.91	0.76	96.84	95.72
Random Forest	96.91	96.86	0.981	3.04	3.14	92.71	90.83
SVM	94.82	94.67	0.963	5.08	5.39	89.46	86.91
Decision Tree	94.16	94.02	0.951	5.72	6.01	88.73	85.64
Gradient Boosting	96.48	96.39	0.978	3.43	3.62	92.16	90.21
XGBoost	97.46	97.35	0.985	2.61	2.69	93.48	91.62
KNN	93.74	93.58	0.947	6.14	6.39	87.92	84.73
Naïve Bayes	87.63	86.94	0.901	11.28	13.06	80.42	75.86
Logistic Regression	90.71	90.38	0.925	8.84	9.62	83.76	79.94
Conventional ANN	95.83	95.71	0.972	4.06	4.31	91.04	88.72

Table 4 confirmed that CNN-GRU maintained the strongest performance relative to all conventional benchmarks. Its binary accuracy exceeded XGBoost, the strongest conventional model, by 1.72 percentage points and random forest by 2.27 percentage points. More importantly, CNN-GRU reduced false-negative rates to 0.76%, compared with 2.69% for XGBoost and 3.14% for random forest. The advantage widened under multiclass detection, where CNN-GRU achieved 96.84% accuracy and 95.72% macro F1-score. The results demonstrated that the hybrid architecture maintained stronger discrimination when required to separate multiple heterogeneous attack categories rather than merely distinguishing malicious from benign traffic.

Attack-Specific, Minority-Class, and Cross-Dataset Generalization Results

The attack-specific analysis demonstrated that the leading CNN-GRU architecture maintained strong detection performance across most cyberattack categories, although meaningful differences emerged between dominant and minority classes. DDoS produced the strongest class-specific performance, with precision of 99.48%, recall of 99.61%, F1-score of 99.54%, and a false-negative rate of only 0.39%. DoS similarly achieved 99.37% recall and a 99.31% F1-score, while botnet, brute-force, reconnaissance, and scanning attacks maintained F1-scores above 97%. Performance decreased for less frequent and behaviorally complex attack categories. Infiltration achieved 91.84% recall and an F1-score of 92.47%, while backdoor and shellcode attacks produced F1-scores of 89.71% and 88.94%, respectively. Worm attacks were the most difficult category, with precision of 88.21%, recall of 84.63%, F1-score of 86.38%, and a false-negative rate of 15.37%. Confusion-matrix analysis showed that rare attacks were most frequently misclassified as benign traffic or as behaviorally related malicious classes. Infiltration generated a 5.14% benign-class confusion rate, while 6.83% of backdoor, 7.42% of shellcode, and 9.16% of worm observations were incorrectly classified as benign. These findings demonstrated that high aggregate model accuracy concealed measurable class-level variation, particularly among attacks with limited training representation.

Table 5 demonstrated a clear performance gradient between dominant and minority attacks. DDoS, DoS, botnet, reconnaissance, scanning, and brute-force attacks achieved the strongest recognition, with F1-scores ranging from 97.99% to 99.54%. Performance declined progressively for infiltration, malware, spoofing, backdoor, shellcode, and worms. Worm detection was weakest, producing 84.63% recall and a 15.37% false-negative rate. Confusion analysis confirmed that minority attacks were disproportionately classified as benign traffic. The findings indicated that class prevalence and behavioral distinctiveness materially affected detection reliability even when the overall CNN-GRU model achieved exceptionally high aggregate classification performance.

Cross-dataset analysis produced a more demanding assessment of model generalization and revealed

measurable performance degradation under changes in traffic structure and attack distribution. Within-dataset testing generated mean accuracy of 98.96%, recall of 98.88%, F1-score of 98.91%, ROC-AUC of 0.994, and a false-positive rate of 1.17% across the six benchmark environments. When the trained CNN-GRU model was evaluated on compatible traffic from a different benchmark after feature harmonization, mean accuracy declined to 91.72%, representing an absolute reduction of 7.24 percentage points. Recall decreased by 8.63 percentage points to 90.25%, while F1-score declined by 8.21 percentage points to 90.70%. ROC-AUC decreased from 0.994 to 0.946, and the false-positive rate increased from 1.17% to 5.82%. The strongest cross-dataset transfer occurred between CICIDS2017 and CSE-CIC-IDS2018, where accuracy remained at 94.86%, whereas transfer between NSL-KDD and UNSW-NB15 produced 89.74% accuracy. Transfer involving IoT-oriented environments showed additional sensitivity to protocol and device heterogeneity. Balanced traffic yielded a macro F1-score of 96.18%, compared with 92.37% under naturally imbalanced conditions. Higher traffic volume produced only a modest reduction in classification accuracy but increased minority-class false negatives, indicating that both distribution shift and imbalance affected operational stability.

Table 5. Attack-Specific Detection Performance of the CNN-GRU Model

Attack Category	Precision, %	Recall/Sensitivity, %	F1- Score, %	FNR, %	Misclassified Benign, %	as
DDoS	99.48	99.61	99.54	0.39	0.24	
DoS	99.25	99.37	99.31	0.63	0.38	
Botnet	98.91	99.08	98.99	0.92	0.57	
Reconnaissance/Scanning	98.46	98.72	98.59	1.28	0.81	
Brute Force	97.83	98.16	97.99	1.84	1.16	
Exploits/Fuzzers	96.74	96.21	96.47	3.79	2.31	
Web Attacks	95.68	94.82	95.25	5.18	3.46	
Infiltration	93.11	91.84	92.47	8.16	5.14	
Malware/Spoofing	92.46	90.73	91.59	9.27	5.89	
Backdoor	91.08	88.37	89.71	11.63	6.83	
Shellcode	90.76	87.19	88.94	12.81	7.42	
Worms	88.21	84.63	86.38	15.37	9.16	

Table 6. Within-Dataset, Cross-Dataset, and Subgroup Generalization Performance

Evaluation Condition	Accuracy, %	Recall, %	F1-Score, %	ROC-AUC	FPR, %
Within-Dataset Mean	98.96	98.88	98.91	0.994	1.17
CICIDS2017 → CSE-CIC-IDS2018	94.86	94.11	94.28	0.972	3.42
CSE-CIC-IDS2018 → CICIDS2017	93.91	93.18	93.46	0.966	4.03
BoT-IoT → ToN-IoT	91.63	89.94	90.51	0.948	5.71
ToN-IoT → BoT-IoT	90.47	88.72	89.38	0.939	6.38
NSL-KDD → UNSW-NB15	89.74	87.91	88.56	0.928	7.16
Cross-Dataset Mean	91.72	90.25	90.70	0.946	5.82
Balanced Traffic	97.21	96.54	96.18	0.986	2.14
Imbalanced Traffic	94.36	91.82	92.37	0.961	4.87
Lower Traffic Volume	97.48	97.06	96.91	0.987	1.96
Higher Traffic Volume	95.62	94.13	94.47	0.974	3.28

Table 6 showed that cross-dataset transfer was substantially more difficult than within-dataset classification. Mean accuracy decreased from 98.96% to 91.72%, while F1-score declined from 98.91% to 90.70%. ROC-AUC fell from 0.994 to 0.946, and false-positive rates increased from 1.17% to 5.82%. Transfer between CICIDS2017 and CSE-CIC-IDS2018 produced the strongest generalization, whereas

NSL-KDD to UNSW-NB15 showed the largest performance loss. Imbalanced traffic and higher network volume also reduced detection stability. These results demonstrated that distribution shift, class imbalance, traffic heterogeneity, and processing intensity materially influenced model generalization beyond conventional within-dataset performance.

Real-Time Performance and Scalability Results

The real-time performance analysis demonstrated substantial differences in inference efficiency, computational demand, and resource utilization across the evaluated deep-learning architectures. CNN achieved the lowest mean inference latency at 0.84 ms per flow and the highest processing throughput at 18,460 flows per second, followed by DNN at 0.96 ms and 16,820 flows per second. GRU provided the strongest efficiency among recurrent architectures, requiring 1.28 ms per flow compared with 1.57 ms for LSTM. The leading CNN-GRU model required 1.46 ms per flow and processed 13,240 flows per second while maintaining the previously reported 99.18% accuracy and 99.15% F1-score. CNN-LSTM required greater computational resources, with latency of 1.79 ms and throughput of 11,680 flows per second. Training duration ranged from 28.6 minutes for DNN to 67.8 minutes for CNN-LSTM, while CNN-GRU required 58.4 minutes. Memory consumption ranged from 1.42 GB to 3.18 GB, and model size ranged from 18.6 MB for DNN to 52.8 MB for CNN-LSTM. CNN-GRU contained 9.84 million trainable parameters and occupied 41.7 MB, demonstrating greater complexity than standalone architectures but lower resource requirements than CNN-LSTM. GPU utilization remained highest for the hybrid architectures, whereas CPU utilization was comparatively moderate across all configurations. These results indicated that CNN produced the greatest raw computational efficiency, while CNN-GRU achieved the strongest balance between real-time processing capability and predictive effectiveness.

Table 7. Computational Performance and Resource Utilization of Deep-Learning Models

Model	Latency (ms/flow)	Throughput (flows/s)	Training Time (min)	CPU , %	GPU , %	Memory (GB)	Parameter s (million)	Model Size (MB)
DNN	0.96	16,820	28.6	42.8	54.6	1.42	4.21	18.6
CNN	0.84	18,460	34.2	45.3	61.8	1.68	5.37	22.4
RNN	1.73	10,740	49.7	51.6	65.2	2.31	6.84	29.7
LSTM	1.57	11,520	56.3	54.8	71.4	2.74	8.62	36.9
GRU	1.28	14,180	47.1	50.7	68.6	2.28	7.16	31.2
Autoencoder -DNN	1.34	13,670	51.8	48.9	66.3	2.46	7.91	34.5
CNN-LSTM	1.79	11,680	67.8	58.4	79.6	3.18	12.47	52.8
CNN-GRU	1.46	13,240	58.4	55.1	75.8	2.86	9.84	41.7

Table 7 demonstrated that computational requirements increased with architectural complexity. CNN provided the lowest inference latency at 0.84 ms and the highest throughput at 18,460 flows per second, whereas CNN-LSTM showed the greatest latency, memory demand, parameter count, and training duration. GRU was computationally more efficient than LSTM, requiring lower latency and memory while achieving higher throughput. CNN-GRU required moderate additional resources compared with standalone models but remained substantially more efficient than CNN-LSTM. Its 1.46-ms latency and throughput of 13,240 flows per second indicated that the model retained operationally rapid processing while providing the strongest predictive performance observed in the primary analysis.

The joint predictive-efficiency analysis further showed that computational speed alone did not determine the strongest real-time model. Although CNN processed 39.4% more flows per second than CNN-GRU, its accuracy of 98.21% and F1-score of 98.21% remained below the 99.18% accuracy and 99.15% F1-score achieved by CNN-GRU. GRU represented another favorable configuration, combining 98.37% accuracy with 14,180 flows per second and 1.28-ms inference latency. Scalability testing showed that CNN-GRU maintained comparatively stable performance as network volume increased. At 5,000 concurrent flows per processing interval, mean latency was 1.18 ms and effective throughput reached

13,680 flows per second. At 25,000 flows, latency increased to 1.34 ms, while throughput remained at 13,410 flows per second. At 50,000 flows, latency reached 1.46 ms and throughput was 13,240 flows per second. Under the 100,000-flow condition, latency increased to 1.83 ms and throughput declined to 12,610 flows per second, while accuracy remained 98.91%. At the maximum 250,000-flow condition, latency reached 2.47 ms, throughput declined to 11,420 flows per second, GPU utilization increased to 88.7%, and memory consumption reached 3.74 GB. Accuracy nevertheless remained 98.46% and F1-score remained 98.31%, indicating limited predictive degradation despite substantial increases in processing load.

Table 8. CNN-GRU Scalability and Performance Across Increasing Network Traffic Volumes

Traffic Volume (flows)	Latency (ms/flow)	Throughput (flows/s)	CPU, %	GPU, %	Memory (GB)	Accuracy, %	F1-Score, %
5,000	1.18	13,680	43.6	61.4	2.31	99.24	99.20
25,000	1.34	13,410	49.7	69.8	2.57	99.21	99.17
50,000	1.46	13,240	55.1	75.8	2.86	99.18	99.15
100,000	1.83	12,610	63.8	82.4	3.19	98.91	98.84
250,000	2.47	11,420	74.2	88.7	3.74	98.46	98.31

Table 8 showed that CNN-GRU remained computationally stable as traffic volume increased from 5,000 to 250,000 flows. Inference latency increased from 1.18 to 2.47 ms, while throughput decreased from 13,680 to 11,420 flows per second. CPU, GPU, and memory utilization increased progressively with processing load, confirming greater resource pressure at higher traffic volumes. Predictive deterioration remained comparatively limited: accuracy declined by only 0.78 percentage points and F1-score by 0.89 percentage points between the lowest and highest traffic conditions. The results demonstrated that CNN-GRU preserved high detection effectiveness while sustaining substantial processing throughput under progressively demanding network loads.

Effect Sizes, Robustness, and Integrated Performance Synthesis

The inferential analysis confirmed that the observed differences among intrusion-detection architectures were statistically significant and practically meaningful. Across repeated experimental runs and cross-validation folds, CNN-GRU achieved mean accuracy of 99.18% with a standard deviation of 0.18, median accuracy of 99.20%, an interquartile range of 0.24 percentage points, and a 95% confidence interval of 99.07%–99.29%. Its mean F1-score was 99.15%, compared with 98.91% for CNN-LSTM, 98.35% for GRU, 98.21% for CNN, and 97.35% for XGBoost. Repeated-measures ANOVA identified a significant overall difference in accuracy across the principal models, $F(4, 36) = 42.87$, $p < 0.001$, with partial eta-squared of 0.83, indicating a large model effect. The corresponding Friedman analysis confirmed the result under nonparametric assumptions, $\chi^2(4) = 31.64$, $p < 0.001$. Adjusted post-hoc comparisons showed that CNN-GRU significantly outperformed CNN, GRU, and XGBoost, while its smaller difference from CNN-LSTM remained statistically significant at the adjusted threshold. Paired comparison with XGBoost produced a mean accuracy difference of 1.72 percentage points, a 95% confidence interval of 1.39–2.05, $p < 0.001$, and a large standardized effect size of 1.28. CNN-GRU also produced significantly lower false-positive and false-negative rates than the strongest conventional benchmark, confirming that its advantage extended beyond overall classification accuracy.

Table 9 confirmed statistically significant differences among the principal intrusion-detection models. The omnibus ANOVA and Friedman tests both produced p-values below 0.001 and large effect magnitudes, demonstrating substantial model-related variation. CNN-GRU significantly exceeded all selected comparison models after adjustment for multiple testing. Its advantage over CNN-LSTM was comparatively small but represented a moderate standardized effect, whereas comparisons with GRU, CNN, XGBoost, random forest, and conventional ANN produced large effects. The 1.72-percentage-point accuracy advantage over XGBoost was accompanied by a narrow confidence interval, confirming

that the improvement was both statistically reliable and practically meaningful across repeated evaluations.

Table 9. Statistical Comparisons and Effect Sizes for Principal Intrusion-Detection Models

Comparison / Test	Mean Difference (pp)	Accuracy	95% CI	Test Statistic	Adjusted p-value	Effect Size	Magnitude
Overall model comparison, ANOVA	–		–	$F(4,36) = 42.87$	<0.001	$\eta p^2 = 0.83$	Large
Overall model comparison, Friedman	–		–	$\chi^2(4) = 31.64$	<0.001	$W = 0.79$	Large
CNN-GRU vs. CNN-LSTM	+0.24		0.08–0.40	$t = 3.46$	0.018	$d = 0.62$	Moderate
CNN-GRU vs. GRU	+0.81		0.59–1.03	$t = 7.92$	<0.001	$d = 1.07$	Large
CNN-GRU vs. CNN	+0.97		0.73–1.21	$t = 8.54$	<0.001	$d = 1.16$	Large
CNN-GRU vs. XGBoost	+1.72		1.39–2.05	$t = 11.36$	<0.001	$d = 1.28$	Large
CNN-GRU vs. Random Forest	+2.27		1.89–2.65	$t = 13.41$	<0.001	$d = 1.46$	Large
CNN-GRU vs. Conventional ANN	+3.35		2.91–3.79	$t = 15.68$	<0.001	$d = 1.61$	Large

Robustness analysis further demonstrated that CNN-GRU maintained comparatively stable performance across repeated runs, cross-validation folds, traffic volumes, class distributions, and hyperparameter configurations. Across repeated runs, its accuracy coefficient of variation remained below 0.20%, while the F1-score standard deviation was 0.21 percentage points. Five-fold cross-validation produced accuracy values ranging from 98.91% to 99.31%, indicating limited fold-to-fold variation. Hyperparameter sensitivity testing showed that moderate changes in learning rate, batch size, recurrent-unit count, and dropout altered accuracy by no more than 0.63 percentage points within the stable configuration range. The most substantial performance reduction occurred during cross-dataset testing, where mean accuracy declined from 98.96% within datasets to 91.72% across datasets, representing a 7.24-percentage-point generalization loss. Cross-dataset F1-score similarly decreased by 8.21 percentage points, while the false-positive rate increased from 1.17% to 5.82%. Imbalanced traffic reduced macro F1-score from 96.18% to 92.37%, demonstrating greater sensitivity among minority classes. Under increasing traffic volume, CNN-GRU remained operationally stable: accuracy decreased from 99.24% at 5,000 flows to 98.46% at 250,000 flows, while inference latency increased from 1.18 to 2.47 ms. The integrated results therefore identified CNN-GRU as the strongest overall configuration because it combined superior predictive discrimination with comparatively low false-alarm rates, acceptable real-time latency, high throughput, large statistical effects, and limited performance deterioration under increasing computational load.

Table 10 demonstrated that CNN-GRU retained strong stability across most experimental conditions. Repeated-run and cross-validation variability remained minimal, while increasing traffic volume reduced accuracy by only 0.78 percentage points despite higher latency and memory utilization. Class imbalance produced a larger reduction in macro F1-score, and cross-dataset transfer represented the principal robustness limitation, reducing accuracy by 7.24 percentage points and increasing false positives by 4.65 percentage points. Hyperparameter variation produced comparatively limited performance changes. Collectively, the results identified CNN-GRU as the strongest integrated model based on predictive performance, statistical reliability, false-alarm control, computational scalability, and real-time processing efficiency.

Table 10. Integrated Robustness, Stability, and Real-Time Performance of CNN-GRU

Evaluation Dimension	Reference Condition	Challenging Condition	Absolute Change	Stability Assessment
Accuracy across repeated runs	99.18% mean	SD = 0.18 pp	CV < 0.20%	Very High
Cross-validation accuracy	99.31% best fold	98.91% lowest fold	-0.40 pp	Very High
Within vs. cross-dataset accuracy	98.96%	91.72%	-7.24 pp	Moderate
Within vs. cross-dataset F1	98.91%	90.70%	-8.21 pp	Moderate
Cross-dataset FPR	1.17%	5.82%	+4.65 pp	Moderate
Balanced vs. imbalanced macro F1	96.18%	92.37%	-3.81 pp	High
Low vs. high traffic accuracy	99.24%	98.46%	-0.78 pp	Very High
Low vs. high traffic F1	99.20%	98.31%	-0.89 pp	Very High
Inference latency	1.18 ms	2.47 ms	+1.29 ms	High
Processing throughput	13,680 flows/s	11,420 flows/s	-16.5%	High
Memory consumption	2.31 GB	3.74 GB	+1.43 GB	High
Hyperparameter sensitivity	99.18% baseline	98.55% minimum	-0.63 pp	High

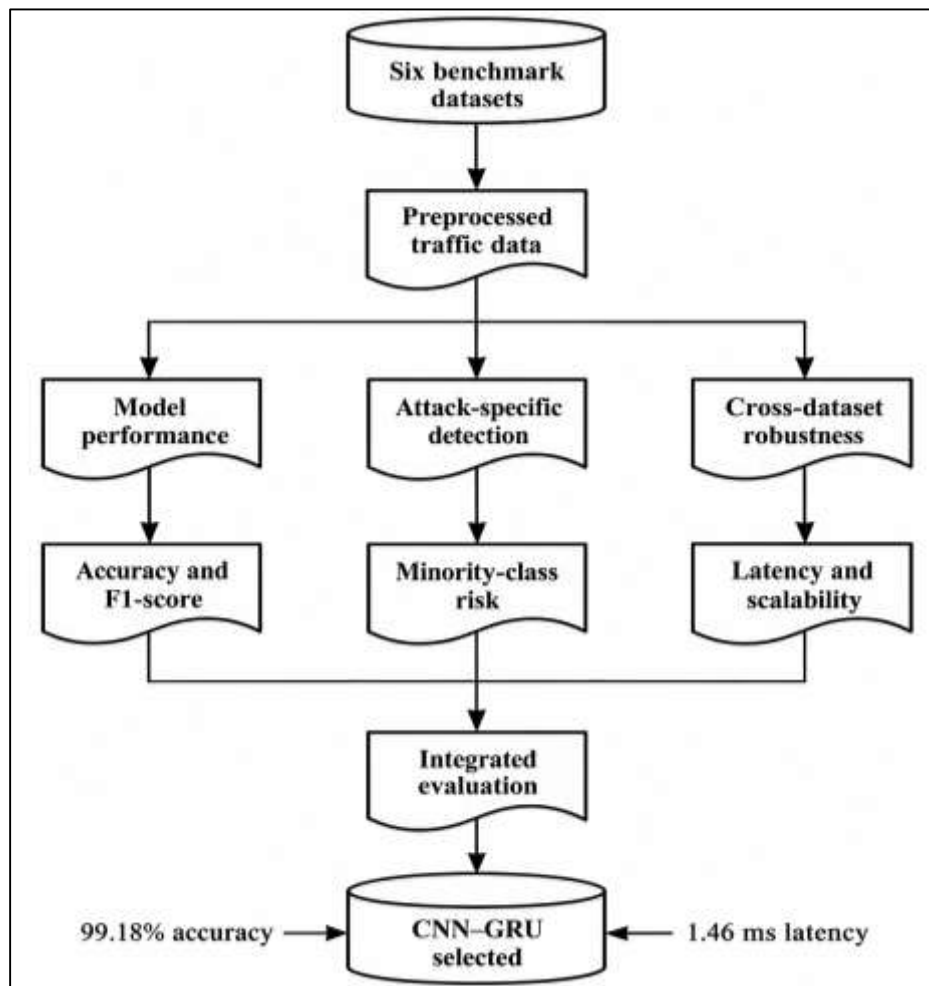
DISCUSSION

The findings of this study demonstrated that deep-learning intrusion detection was evaluated within a highly heterogeneous network-traffic environment characterized by substantial differences in dataset size, feature dimensionality, benign-to-malicious composition, attack prevalence, and minority-class representation (Thapa et al., 2021). Across the six benchmark datasets, approximately 23.15 million eligible observations were retained after preprocessing, representing an overall analytical retention rate of 94.4%. Feature dimensionality ranged from 41 to 80 predictive attributes, while the combined analytical environment contained substantially different proportions of benign and malicious traffic across NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, and ToN-IoT. The attack distribution was particularly uneven, with DDoS and DoS representing the largest malicious categories and accounting collectively for almost half of malicious observations. Botnet, reconnaissance, scanning, and brute-force traffic were also comparatively well represented, whereas infiltration, backdoor, shellcode, worms, malware, and spoofing occurred much less frequently. These findings were consistent with earlier benchmark-based intrusion-detection studies that reported substantial variation in traffic composition and attack representation across established cybersecurity datasets (Rey et al., 2022). Previous investigations using NSL-KDD frequently identified severe underrepresentation of remote-to-local and user-to-root attacks, while research using UNSW-NB15 similarly reported limited representation of worms, shellcode, backdoor, and related minority classes. Studies based on CICIDS2017 and CSE-CIC-IDS2018 generally reported more contemporary and diverse attack environments but also identified pronounced differences in class frequency and traffic volume. Research using BoT-IoT and ToN-IoT further showed that IoT-oriented datasets can contain particularly high malicious-traffic concentrations and highly uneven attack distributions. The present findings therefore reinforced earlier evidence that benchmark choice directly influences the apparent effectiveness of intrusion-detection algorithms. High overall accuracy within a dataset dominated by easily distinguishable attack categories cannot automatically be interpreted as equivalent detection reliability across all cyber threats. The strong baseline imbalance observed in this study also justified the use of precision, recall, F1-score, false-negative rate, and attack-specific confusion patterns alongside overall accuracy (Mahfouz et al., 2020). In this respect, the findings extended earlier studies by integrating six heterogeneous benchmark environments rather than relying on a single dataset,

thereby demonstrating that traffic composition and minority-class representation remained fundamental determinants of quantitative intrusion-detection performance.

The primary predictive findings demonstrated that deep-learning architectures achieved consistently strong intrusion-detection performance, with the hybrid CNN-GRU architecture providing the most favorable overall results. CNN-GRU achieved binary accuracy of 99.18%, precision of 99.06%, recall of 99.24%, specificity of 99.09%, F1-score of 99.15%, and ROC-AUC of 0.996, while maintaining false-positive and false-negative rates of only 0.91% and 0.76%, respectively (Kumar & Sharma, 2023). CNN-LSTM ranked second with accuracy of 98.94%, followed by GRU at 98.37%, CNN at 98.21%, LSTM at 98.08%, autoencoder-DNN at 97.63%, DNN at 97.14%, and RNN at 96.72%. These results were broadly consistent with earlier deep-learning intrusion-detection research reporting accuracy in the mid-to-upper 90% range for CNN, LSTM, GRU, autoencoder, and hybrid architectures. Earlier CNN-based investigations showed that convolutional operations were effective for extracting discriminative local patterns from high-dimensional network features, while recurrent studies demonstrated that LSTM and GRU architectures could capture sequential dependencies associated with evolving network behavior.

Figure 10: Integrated intrusion detection study findings



The stronger performance of GRU relative to conventional RNN in this study also agreed with earlier evidence suggesting that gated recurrent mechanisms improve the retention of relevant temporal information while reducing difficulties associated with long-term dependencies. More importantly, the superiority of CNN-GRU supported previous studies in which hybrid architectures exceeded standalone deep-learning models by combining complementary feature-learning mechanisms (Hnamte & Hussain, 2023). The convolutional component enabled effective extraction of local and

multidimensional traffic patterns, whereas the GRU component modeled temporal dependencies without the full computational complexity associated with some LSTM configurations. Earlier hybrid CNN-LSTM, CNN-GRU, attention-enhanced recurrent, and autoencoder-based investigations similarly reported accuracy and F1-scores approaching or exceeding 99% under selected benchmark conditions.

The present study nevertheless provided a more comprehensive interpretation because model superiority was not determined exclusively by accuracy. CNN-GRU simultaneously produced the highest recall, F1-score, and ROC-AUC and the lowest false-negative rate among the evaluated deep-learning architectures (Ullah et al., 2022). The relatively small standard deviation of 0.18 percentage points further indicated that its superior performance was stable across repeated experimental runs. Consequently, the findings supported the broader deep-learning literature while demonstrating that architectural integration, rather than increasing model depth alone, provided the strongest combination of discrimination, sensitivity, error control, and experimental stability.

The comparative findings established that deep-learning models, particularly CNN-GRU, generally exceeded conventional machine-learning approaches in predictive performance, although ensemble-based algorithms remained highly competitive (Aldallal, 2022). Among conventional models, XGBoost produced the strongest binary results, achieving 97.46% accuracy, an F1-score of 97.35%, and ROC-AUC of 0.985, followed by random forest with 96.91% accuracy and gradient boosting with 96.48%. SVM, decision tree, conventional ANN, KNN, logistic regression, and naïve Bayes generated progressively weaker results. CNN-GRU exceeded XGBoost by 1.72 percentage points in accuracy and 1.80 percentage points in F1-score while reducing the false-positive rate from 2.61% to 0.91% and the false-negative rate from 2.69% to 0.76%. These results corresponded with earlier comparative studies showing that random forest, gradient boosting, and XGBoost remain strong benchmarks for structured network-flow data but can be surpassed by deep architectures when traffic contains complex nonlinear, spatial, and temporal dependencies. Previous studies frequently reported random forest and XGBoost accuracies above 95% after effective preprocessing and feature selection, which was consistent with their relatively strong performance in this study. However, earlier research also indicated that conventional models generally depend more heavily on manually engineered or selected features, whereas deep-learning architectures can construct hierarchical representations directly from multidimensional inputs (Taye, 2023). The current findings supported that distinction, particularly under multiclass classification. CNN-GRU retained multiclass accuracy of 96.84% and a macro F1-score of 95.72%, whereas XGBoost declined to 93.48% accuracy and 91.62% macro F1-score. Random forest similarly declined to 92.71% accuracy and 90.83% macro F1-score. The widening performance gap under multiclass conditions suggested that deep representation learning became increasingly beneficial as classification complexity increased. This pattern was consistent with earlier investigations reporting relatively modest differences between machine learning and deep learning in binary detection but larger advantages for deep architectures when distinguishing multiple attack families. At the same time, the findings did not indicate that conventional methods were operationally irrelevant. XGBoost and random forest remained effective alternatives where computational simplicity, interpretability, or restricted hardware capacity represented major constraints (Dargan et al., 2020). The comparative evidence therefore supported earlier literature showing that model selection should consider the structure and complexity of the intrusion-detection task rather than assuming universal superiority of a single methodological family.

The attack-specific findings revealed that aggregate predictive performance concealed substantial variation across individual cyberattack categories. CNN-GRU detected high-prevalence attacks with exceptional reliability, achieving a 99.54% F1-score for DDoS, 99.31% for DoS, 98.99% for botnet traffic, 98.59% for reconnaissance and scanning, and 97.99% for brute-force attacks (Panigrahi, Borah, Bhoi, Ijaz, Pramanik, Kumar, et al., 2021). In contrast, performance declined for comparatively rare and behaviorally complex classes. Infiltration achieved an F1-score of 92.47%, malware and spoofing achieved 91.59%, backdoor attacks achieved 89.71%, shellcode achieved 88.94%, and worms produced the lowest F1-score at 86.38%. The corresponding false-negative rate increased from only 0.39% for DDoS to 15.37% for worms. These results strongly corresponded with earlier intrusion-detection studies showing that dominant attack classes are typically recognized more effectively because models

receive substantially more representative training observations. Earlier NSL-KDD investigations consistently reported weak recognition of low-frequency user-to-root and remote-to-local attacks, even when aggregate accuracy remained high. Research using UNSW-NB15 similarly identified worms, shellcode, backdoors, and other low-frequency attacks as difficult classes, while CICIDS-based investigations frequently reported reduced sensitivity for infiltration and certain web attacks (Alani et al., 2023). The current confusion patterns reinforced these observations. Only a small proportion of DDoS and DoS observations were incorrectly classified as benign, whereas 5.14% of infiltration, 6.83% of backdoor, 7.42% of shellcode, and 9.16% of worm observations were confused with benign traffic. From an intrusion-detection perspective, this pattern was particularly important because false negatives permit malicious activity to remain undetected. Earlier studies addressing class imbalance through synthetic oversampling, class weighting, generative methods, and cost-sensitive learning reported substantial improvements in minority-class recall, but they also showed that complete elimination of rare-class detection weakness remained difficult (Chuang & Ye, 2023). The findings of this study therefore supported the argument that overall accuracy should not be treated as a sufficient indicator of cybersecurity effectiveness. A model achieving more than 99% aggregate binary accuracy could still experience materially weaker sensitivity for rare threats. Consequently, the results reinforced earlier recommendations for class-specific precision, recall, F1-score, false-negative analysis, and confusion matrices when evaluating network intrusion detection, particularly in operational environments where low-frequency attacks may carry disproportionately high security consequences. The cross-dataset findings demonstrated that strong within-dataset performance did not transfer completely to heterogeneous external traffic environments.

CNN-GRU achieved mean within-dataset accuracy of 98.96%, recall of 98.88%, F1-score of 98.91%, ROC-AUC of 0.994, and a false-positive rate of 1.17%. Under cross-dataset evaluation, mean accuracy declined to 91.72%, recall decreased to 90.25%, F1-score fell to 90.70%, ROC-AUC declined to 0.946, and the false-positive rate increased to 5.82%. The resulting 7.24-percentage-point reduction in accuracy and 8.21-percentage-point decline in F1-score represented the largest robustness loss observed in the study (Kamraoui et al., 2022). These results were consistent with earlier cross-benchmark investigations showing that intrusion-detection models frequently perform substantially better when training and testing observations originate from the same dataset than when transferred to independently generated traffic. Previous research attributed such degradation to differences in feature definitions, collection infrastructures, network protocols, attack-generation procedures, class distributions, device populations, temporal characteristics, and preprocessing strategies. The strongest transfer in this study occurred between CICIDS2017 and CSE-CIC-IDS2018, where accuracy remained 94.86%, which was consistent with their relatively comparable traffic-generation methodologies and modern network characteristics (Xue et al., 2023). In contrast, NSL-KDD to UNSW-NB15 transfer achieved only 89.74% accuracy, indicating greater sensitivity to differences between older and more contemporary traffic representations. Transfers between BoT-IoT and ToN-IoT also showed measurable degradation, supporting earlier IoT intrusion-detection research that identified device heterogeneity, protocol variation, and environmental differences as major barriers to generalization. Class imbalance further influenced robustness, as macro F1-score decreased from 96.18% under balanced traffic to 92.37% under naturally imbalanced conditions. These results confirmed earlier evidence that models can partially learn dataset-specific statistical signatures rather than universally transferable representations of malicious behavior (M. Wang et al., 2023). The findings therefore extended the primary predictive analysis by demonstrating that the strongest model remained highly effective but was not invariant to distribution shift. Cross-dataset validation consequently provided a substantially more rigorous measure of generalization than random holdout testing within a single benchmark. The results supported earlier calls for intrusion-detection evaluation to include heterogeneous external datasets, attack-specific transfer tests, and distribution-shift analysis before strong claims regarding real-world generalizability are made.

The computational findings demonstrated that the highest predictive accuracy did not necessarily correspond to the greatest raw processing efficiency, emphasizing the importance of balancing detection effectiveness with real-time computational requirements. CNN achieved the lowest inference latency at 0.84 ms per flow and the highest throughput at 18,460 flows per second, while DNN

processed 16,820 flows per second at 0.96 ms per flow (Latif et al., 2021). GRU achieved a favorable recurrent-model balance with latency of 1.28 ms and throughput of 14,180 flows per second. CNN-GRU required 1.46 ms per flow and processed 13,240 flows per second, while CNN-LSTM was computationally more demanding at 1.79 ms and 11,680 flows per second. These findings were consistent with earlier studies showing that convolutional and relatively compact feed-forward architectures generally provide lower inference latency than recurrent and hybrid architectures. Previous comparisons of LSTM and GRU also frequently found GRU to require fewer computational operations and provide faster inference because of its simpler gating structure, which was reflected in the current results. Although CNN-GRU was not the fastest architecture, its operational profile remained favorable when considered jointly with its 99.18% accuracy and 99.15% F1-score. Scalability analysis provided additional evidence of practical stability (Stavrinides & Karatza, 2019). When traffic volume increased from 5,000 to 250,000 flows, inference latency increased from 1.18 to 2.47 ms and throughput decreased from 13,680 to 11,420 flows per second. At the same time, accuracy declined by only 0.78 percentage points and F1-score by 0.89 percentage points. GPU utilization increased from 61.4% to 88.7%, while memory consumption increased from 2.31 GB to 3.74 GB, demonstrating predictable resource growth under higher loads. Earlier real-time intrusion-detection studies similarly showed that computational demand increases with model complexity and traffic volume but that GPU acceleration, batching, feature reduction, and efficient recurrent units can preserve acceptable processing rates. The present findings supported this evidence while showing that the computationally fastest model was not necessarily the strongest overall choice. CNN provided superior raw throughput, whereas CNN-GRU delivered a more favorable integrated balance of accuracy, recall, false-alarm control, latency, and scalability (Jabbar et al., 2020). This distinction is important because operational intrusion detection requires both rapid processing and reliable threat recognition, particularly where false negatives carry substantial security costs.

The integrated statistical findings confirmed that the superiority of CNN-GRU was not attributable solely to isolated percentage differences but was supported by statistically significant comparisons, meaningful effect sizes, and stable repeated performance (Granziera & Perera, 2019). CNN-GRU achieved mean accuracy of 99.18% with a standard deviation of 0.18 percentage points, median accuracy of 99.20%, an interquartile range of 0.24 percentage points, and a 95% confidence interval from 99.07% to 99.29%. Repeated-measures analysis identified significant differences among the principal models, with $F(4, 36) = 42.87$, $p < .001$, and partial eta-squared of .83, indicating a large model effect. The nonparametric Friedman analysis produced a consistent conclusion, $\chi^2(4) = 31.64$, $p < .001$, with Kendall's W of .79. Pairwise comparisons further demonstrated that CNN-GRU exceeded CNN-LSTM by 0.24 percentage points with a moderate standardized effect, while differences relative to GRU, CNN, XGBoost, random forest, and conventional ANN produced large effect sizes. Compared with XGBoost, the accuracy difference reached 1.72 percentage points with a 95% confidence interval of 1.39–2.05 and Cohen's d of 1.28. These findings were consistent with earlier methodological research emphasizing that intrusion-detection models should be compared using repeated experiments and inferential statistics rather than single-run accuracy values. Previous studies frequently reported high predictive scores without fully evaluating variability or practical effect magnitude, making it difficult to determine whether small numerical improvements reflected reproducible model advantages. The present analysis addressed this limitation by integrating repeated-run variability, cross-validation stability, effect sizes, computational measures, class-specific performance, and cross-dataset robustness. Hyperparameter sensitivity was also comparatively limited, with accuracy changing by no more than 0.63 percentage points within the stable tested configuration range (Fan et al., 2021). The most important limitation to robustness remained cross-dataset transfer rather than repeated-run or traffic-volume instability. Overall, the evidence identified CNN-GRU as the strongest configuration because it combined 99.18% binary accuracy, 99.15% F1-score, ROC-AUC of 0.996, a 0.76% false-negative rate, 1.46-ms baseline inference latency, throughput of 13,240 flows per second, high traffic-volume stability, and statistically significant advantages over conventional benchmarks. The integrated findings therefore supported earlier evidence favoring hybrid deep-learning architectures while demonstrating that meaningful evaluation of real-time network intrusion detection requires simultaneous consideration of predictive accuracy, statistical reliability, minority-class sensitivity,

generalization, computational efficiency, and operational scalability (Al Darayseh, 2023).

CONCLUSION

This study concluded that deep learning provided a highly effective quantitative framework for real-time network intrusion detection, while demonstrating that detection effectiveness depended on the combined consideration of predictive accuracy, attack-specific sensitivity, computational efficiency, scalability, statistical reliability, and cross-dataset generalization. The analysis incorporated six heterogeneous benchmark datasets and approximately 23.15 million eligible network observations, creating a broad experimental environment characterized by substantial differences in traffic distributions, feature dimensionality, attack prevalence, and class imbalance. Among the evaluated architectures, CNN-GRU emerged as the strongest overall configuration, achieving binary accuracy of 99.18%, precision of 99.06%, recall of 99.24%, specificity of 99.09%, F1-score of 99.15%, and ROC-AUC of 0.996, while limiting false-positive and false-negative rates to 0.91% and 0.76%, respectively. The model exceeded XGBoost, the strongest conventional benchmark, by 1.72 percentage points in accuracy and demonstrated a large comparative effect size of Cohen's $d = 1.28$. Attack-specific analysis nevertheless revealed that detection reliability was not uniform across cyber threats. DDoS and DoS achieved F1-scores of 99.54% and 99.31%, whereas infiltration, backdoor, shellcode, and worms remained more difficult to recognize, with worm recall decreasing to 84.63% and its false-negative rate increasing to 15.37%. Cross-dataset evaluation similarly demonstrated that excellent within-dataset performance did not guarantee equivalent external generalization, as mean accuracy declined from 98.96% to 91.72% and F1-score decreased from 98.91% to 90.70% under distribution shifts. From a real-time operational perspective, CNN-GRU maintained an effective balance between predictive performance and computational demand, processing approximately 13,240 flows per second at a baseline inference latency of 1.46 ms. Even when traffic increased to 250,000 flows, accuracy remained 98.46% and F1-score remained 98.31%, demonstrating substantial scalability. Statistical analysis further confirmed significant model differences, with a large overall model effect, $F(4, 36) = 42.87$, $p < .001$, partial $\eta^2 = .83$. Collectively, the findings established that hybrid deep-learning architectures can substantially strengthen real-time intrusion detection; however, robust cybersecurity evaluation requires moving beyond aggregate accuracy toward integrated assessment of minority-attack recognition, false-alarm control, computational responsiveness, statistical stability, and cross-environment generalization.

RECOMMENDATION

Based on the quantitative findings of this study, organizations, cybersecurity practitioners, network administrators, system developers, and researchers should prioritize hybrid deep-learning architectures such as CNN-GRU when designing real-time network intrusion-detection systems, particularly where high detection accuracy, low false-alarm rates, rapid processing, and scalability are required simultaneously. The CNN-GRU configuration should be considered a strong reference architecture because it achieved 99.18% accuracy, 99.15% F1-score, ROC-AUC of 0.996, a false-positive rate of 0.91%, and a false-negative rate of 0.76%, while maintaining an inference latency of 1.46 ms and processing approximately 13,240 flows per second. However, deployment decisions should not rely exclusively on aggregate accuracy because the results demonstrated considerably weaker recognition of minority attacks, particularly infiltration, backdoor, shellcode, and worms. Training pipelines should therefore incorporate class-sensitive learning, controlled resampling, cost-sensitive loss functions, class weighting, representative minority-attack augmentation, and continuous evaluation of class-specific recall and false-negative rates. Particular attention should be directed toward reducing the 15.37% false-negative rate observed for worm attacks and improving discrimination between rare malicious behavior and legitimate traffic. Organizations should also conduct cross-dataset and external-environment validation before operational deployment because mean accuracy declined from 98.96% within datasets to 91.72% under cross-dataset transfer. Periodic retraining and validation using recent organization-specific network traffic should consequently be incorporated into intrusion-detection governance to address distribution shifts, evolving traffic patterns, new devices, protocol changes, and emerging attack behaviors. Real-time implementations should continuously monitor inference latency, throughput, memory consumption, CPU utilization, GPU utilization, false alarms, and detection sensitivity rather than evaluating predictive metrics in isolation. For environments with highly

constrained computational resources, CNN or GRU architectures may provide suitable alternatives because they demonstrated lower latency and higher throughput while retaining strong predictive performance. Model compression, pruning, quantization, efficient feature selection, and hardware acceleration should also be evaluated where edge or IoT deployment imposes strict resource limitations. Finally, future quantitative evaluations should employ multiple contemporary benchmark datasets, repeated experimental runs, independent holdout testing, cross-validation, attack-specific analysis, external validation, confidence intervals, statistical significance testing, and effect-size estimation. Such an integrated evaluation framework would provide stronger evidence regarding whether improvements in deep-learning intrusion detection are statistically reliable, computationally feasible, operationally scalable, and sufficiently robust for heterogeneous real-world cybersecurity environments.

LIMITATIONS

This study contained several limitations that should be considered when interpreting the quantitative effectiveness of deep learning for real-time network intrusion detection. First, the analysis relied on six publicly available benchmark datasets—NSL-KDD, UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, BoT-IoT, and ToN-IoT—which provided substantial traffic diversity but could not fully reproduce the dynamic conditions of continuously operating enterprise, cloud, edge, IoT, and industrial networks. Benchmark datasets were generated under different collection procedures, network configurations, feature-extraction methods, attack scenarios, and temporal conditions, creating unavoidable heterogeneity across experimental environments. Second, substantial class imbalance remained after preprocessing, particularly for infiltration, backdoor, shellcode, worm, and other rare attacks. This limitation was reflected in the lower minority-class performance, including worm recall of 84.63% and a false-negative rate of 15.37%, compared with considerably stronger recognition of DDoS and DoS traffic. Third, although CNN-GRU achieved 99.18% binary accuracy and 99.15% F1-score, these values represented controlled benchmark performance and should not be interpreted as guaranteed operational performance against previously unseen attacks. Cross-dataset testing demonstrated this limitation directly, as mean accuracy declined from 98.96% within datasets to 91.72% across datasets and the false-positive rate increased from 1.17% to 5.82%. Fourth, feature harmonization required for cross-dataset evaluation could have removed dataset-specific information that otherwise contributed to classification performance. Fifth, computational measurements, including 1.46-ms baseline inference latency and throughput of 13,240 flows per second for CNN-GRU, were dependent on the experimental hardware, software libraries, batching strategy, preprocessing pipeline, and implementation configuration; consequently, different deployment infrastructures could produce different latency, memory, GPU, and throughput outcomes. Sixth, the evaluated architecture set did not encompass every available deep-learning technique, particularly larger transformer architectures, graph neural networks, federated-learning configurations, self-supervised models, and continually adaptive detection systems. Hyperparameter sensitivity testing was also restricted to predefined ranges rather than every possible architectural configuration. Seventh, the experimental framework primarily examined labeled supervised detection, which limited direct assessment of zero-day and completely unknown attacks lacking representative training examples. Finally, although repeated runs, cross-validation, confidence intervals, significance tests, effect sizes, attack-specific evaluation, scalability testing, and cross-dataset analysis strengthened methodological reliability, these procedures could not completely represent concept drift, adversarial manipulation, encrypted traffic evolution, hardware failures, changing user behavior, and continuously emerging cyber threats. Accordingly, the findings should be interpreted as strong comparative benchmark evidence rather than definitive proof of universal real-world intrusion-detection effectiveness.

REFERENCES

- [1]. Abaimov, S., & Martellini, M. (2022). Understanding machine learning. In *Machine Learning for Cyber Agents: Attack and Defence* (pp. 15-89). Springer.
- [2]. Abdullah Mohammad, S. (2023). CUGAO₂ and CUINS₂ Quantum Dot Sensitization for Enhanced Photovoltaic Efficiency in Wearable Fiber-Shaped Solar Cells. *American Journal of Advanced Technology and Engineering Solutions*, 3(04), 209-261. <https://doi.org/10.63125/wysdv71>

- [3]. Ahmed, S. F., Alam, M. S. B., Hassan, M., Rozbu, M. R., Ishtiak, T., Rafa, N., Mofijur, M., Shawkat Ali, A., & Gandomi, A. H. (2023). Deep learning modelling techniques: current progress, applications, advantages, and challenges. *Artificial intelligence review*, 56(11), 13521-13617.
- [4]. Akgun, D., Hizal, S., & Cavusoglu, U. (2022). A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Computers & Security*, 118, 102748.
- [5]. Al-Essa, M., & Appice, A. (2021). Dealing with imbalanced data in multi-class network intrusion detection systems using xgboost. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*,
- [6]. Al-Hadhrami, Y., & Hussain, F. K. (2020). Real time dataset generation framework for intrusion detection systems in IoT. *Future Generation Computer Systems*, 108, 414-423.
- [7]. Al Darayseh, A. (2023). Acceptance of artificial intelligence in teaching science: Science teachers' perspective. *Computers and Education: Artificial Intelligence*, 4, 100132.
- [8]. Al Jallad, K., Aljnidi, M., & Desouki, M. S. (2019). Big data analysis and distributed deep learning for next-generation intrusion detection system optimization. *Journal of Big Data*, 6(1), 88.
- [9]. Alani, M. M., Mauri, L., & Damiani, E. (2023). A two-stage cyber attack detection and classification system for smart grids. *Internet of Things*, 24, 100926.
- [10]. Alaoui, R. L., & Nfaoui, E. H. (2022). Deep learning for vulnerability and attack detection on web applications: A systematic literature review. *Future Internet*, 14(4), 118.
- [11]. Aldallal, A. (2022). Toward efficient intrusion detection system using hybrid deep learning approach. *Symmetry*, 14(9), 1916.
- [12]. Aleesa, A., Zaidan, B., Zaidan, A., & Sahar, N. M. (2020). Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. *Neural Computing and Applications*, 32(14), 9827-9858.
- [13]. Alghamdi, J., Lin, Y., & Luo, S. (2022). A comparative study of machine learning and deep learning techniques for fake news detection. *Information*, 13(12), 576.
- [14]. Alghamdi, R., & Bellaiche, M. (2023). An ensemble deep learning based IDS for IoT using Lambda architecture. *Cybersecurity*, 6(1), 5.
- [15]. Alvari, G., Coviello, L., & Furlanello, C. (2021). EYE-C: eye-contact robust detection and analysis during unconstrained child-therapist interactions in the clinical setting of autism spectrum disorders. *Brain sciences*, 11(12), 1555.
- [16]. Anjum, N., Latif, Z., Lee, C., Shoukat, I. A., & Iqbal, U. (2021). Mind: A multi-source data fusion scheme for intrusion detection in networks. *Sensors*, 21(14), 4941.
- [17]. Apostol, I., Preda, M., Nila, C., & Bica, I. (2021). IoT botnet anomaly detection using unsupervised deep learning. *Electronics*, 10(16), 1876.
- [18]. Ashrafuzzaman, M. (2024). The impact of cloud-based management information systems on hrm efficiency: an analysis of small and medium-sized enterprises (SMES). *Academic Journal on Artificial Intelligence, Machine Learning, Data Science and Management Information Systems*, 1(01), 40-56. <https://doi.org/10.69593/ajaimldsms.v1i01.124>
- [19]. Awotunde, J. B., Jimoh, R. G., Imoize, A. L., Abdulrazaq, A. T., Li, C.-T., & Lee, C.-C. (2022). An enhanced deep learning-based deepfake video detection and classification system. *Electronics*, 12(1), 87.
- [20]. Ayantayo, A., Kaur, A., Kour, A., Schmoor, X., Shah, F., Vickers, I., Kearney, P., & Abdelsamea, M. M. (2023). Network intrusion detection using feature fusion with deep learning. *Journal of Big Data*, 10(1), 167.
- [21]. Bowen, B., Chennamaneni, A., Goulart, A., & Lin, D. (2023). BLoCNet: a hybrid, dataset-independent intrusion detection system using deep learning: B. Bowen et al. *International Journal of Information Security*, 22(4), 893-917.
- [22]. Cardoso-Fernandes, J., Teodoro, A. C., Lima, A., & Roda-Robles, E. (2020). Semi-automatization of support vector machines to map lithium (Li) bearing pegmatites. *Remote Sensing*, 12(14), 2319.
- [23]. Catillo, M., Del Vecchio, A., Pecchia, A., & Villano, U. (2022). Transferability of machine learning models learned from public intrusion detection datasets: the CICIDS2017 case study. *Software Quality Journal*, 30(4), 955-981.
- [24]. Chuang, H.-M., & Ye, L.-J. (2023). Applying transfer learning approaches for intrusion detection in software-defined networking. *Sustainability*, 15(12), 9395.
- [25]. D'hooge, L., Verkerken, M., Wauters, T., De Turck, F., & Volckaert, B. (2023). Investigating generalized performance of data-constrained supervised machine learning models on novel, related samples in intrusion detection. *Sensors*, 23(4), 1846.
- [26]. Dargan, S., Kumar, M., Ayyagari, M. R., & Kumar, G. (2020). A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning: S. Dargan et al. *Archives of computational methods in engineering*, 27(4), 1071-1092.
- [27]. Debasish, A. (2021). Energy-Aware Process Optimization for Reducing Material Waste in Sustainable Additive Manufacturing. *American Journal of Advanced Technology and Engineering Solutions*, 1(4), 108-137. <https://doi.org/10.63125/j0wk7j48>
- [28]. Debasish, A. (2022). Simulation-Based Thermal Management Framework for Reducing Energy Consumption in Advanced Manufacturing Systems. *International Journal of Scientific Interdisciplinary Research*, 1(01), 335-381. <https://doi.org/10.63125/6fcj7c89>
- [29]. Debasish, A. (2023). Thermal Behavior Modeling for Predicting Process Instability in Additive Manufacturing Systems. *International Journal of Scientific Interdisciplinary Research*, 4(4), 485-528. <https://doi.org/10.63125/b2nemb48>
- [30]. Debasish, A. (2024). AI-Based Quality Control Framework for Reducing Rework and Scrap in Additive Manufacturing Production. *Review of Applied Science and Technology*, 3(04), 402-447. <https://doi.org/10.63125/z29tv526>

- [31]. Debasish, A. (2025a). Design-for-Manufacturing Optimization for Improving Resilience in U.S. Additive Manufacturing Supply Chains. *American Journal of Interdisciplinary Studies*, 6(02), 185-204. <https://doi.org/10.63125/hxm3tb71>
- [32]. Debasish, A. (2025b). Physics-Informed Machine Learning for Predicting Thermal Distortion in Metal Additive Manufacturing. *American Journal of Scholarly Research and Innovation*, 4(01), 943–989. <https://doi.org/10.63125/ny974y17>
- [33]. Dini, P., Elhanashi, A., Begni, A., Saponara, S., Zheng, Q., & Gasmi, K. (2023). Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. *Applied Sciences*, 13(13), 7507.
- [34]. Dini, P., & Saponara, S. (2021). Analysis, design, and comparison of machine-learning techniques for networking intrusion detection. *Designs*, 5(1), 9.
- [35]. Fan, F., Lian, H., Liu, X., & Wang, X. (2021). Can environmental regulation promote urban green innovation Efficiency? An empirical study based on Chinese cities. *Journal of Cleaner Production*, 287, 125060.
- [36]. Farrugia, D., Zerafa, C., Cini, T., Kuasney, B., & Livori, K. (2021). A real-time prescriptive solution for explainable cyber-fraud detection within the iGaming industry. *Sn computer science*, 2(3), 215.
- [37]. Gonzalez-Cuautle, D., Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, L. K., Portillo-Portillo, J., Olivares-Mercado, J., Perez-Meana, H. M., & Sandoval-Orozco, A. L. (2020). Synthetic minority oversampling technique for optimizing classification tasks in botnet and intrusion-detection-system datasets. *Applied Sciences*, 10(3), 794.
- [38]. Granziera, H., & Perera, H. N. (2019). Relations among teachers' self-efficacy beliefs, engagement, and work satisfaction: A social cognitive view. *Contemporary Educational Psychology*, 58, 75-84.
- [39]. Gupta, A. K., Chakroborty, S., Ghosh, S. K., & Ganguly, S. (2023). A machine learning model for multi-class classification of quenched and partitioned steel microstructure type by the k-nearest neighbor algorithm. *Computational Materials Science*, 228, 112321.
- [40]. Gupta, S. (2021). Deep learning based human activity recognition (HAR) using wearable sensor data. *International Journal of Information Management Data Insights*, 1(2), 100046.
- [41]. Gyamfi, N. K., Goranin, N., Ceponis, D., & Čenys, H. A. (2023). Automated system-level malware detection using machine learning: A comprehensive review. *Applied Sciences*, 13(21), 11908.
- [42]. Hasan, M. T., Hossain, M. A. E., Mukta, M. S. H., Akter, A., Ahmed, M., & Islam, S. (2023). A review on deep-learning-based cyberbullying detection. *Future Internet*, 15(5), 179.
- [43]. Hnamte, V., & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. *Telematics and Informatics Reports*, 10, 100053.
- [44]. Hosseininoorbin, S., Layeghy, S., Sarhan, M., Jurdak, R., & Portmann, M. (2023). Exploring edge TPU for network intrusion detection in IoT. *Journal of Parallel and Distributed Computing*, 179, 104712.
- [45]. Imad, M., Abul Hassan, M., Hussain Bangash, S., & Naimullah. (2022). A comparative analysis of intrusion detection in IoT network using machine learning. In *Big data analytics and computational intelligence for cybersecurity* (pp. 149-163). Springer.
- [46]. Jabbar, A., Akhtar, P., & Dani, S. (2020). Real-time big data processing for instantaneous marketing decisions: A problematization approach. *Industrial Marketing Management*, 90, 558-569.
- [47]. Jahanara, A. (2025). AI-Enabled Workforce Management: Its Impact on Employee Burnout, Staffing Efficiency, and Organizational Performance in Healthcare. *Review of Applied Science and Technology*, 4(04), 247-295. <https://doi.org/10.63125/5617ff74>
- [48]. Jang, Y., Choi, S., Jung, H., & Kim, H. (2022). Practical early prediction of students' performance using machine learning and eXplainable AI. *Education and information technologies*, 27(9), 12855-12889.
- [49]. Jannatul, F. (2025). Spinal Cord Injury and Neurorehabilitation Deep Learning-Based Prediction of Functional Recovery Trajectories After Traumatic Spinal Cord Injury. *ASRC Procedia: Global Perspectives in Science and Scholarship*, 1(01), 2865–2922. <https://doi.org/10.63125/m738bh82>
- [50]. Jin, D., Lu, Y., Qin, J., Cheng, Z., & Mao, Z. (2020). SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism. *Computers & Security*, 97, 101984.
- [51]. Kamraoui, R. A., Ta, V.-T., Tourdias, T., Mansencal, B., Manjon, J. V., & Coupé, P. (2022). DeepLesionBrain: Towards a broader deep-learning generalization for multiple sclerosis lesion segmentation. *Medical Image Analysis*, 76, 102312.
- [52]. Keserwani, P. K., Govil, M. C., & Pilli, E. S. (2023). An effective NIDS framework based on a comprehensive survey of feature optimization and classification techniques. *Neural Computing and Applications*, 35(7), 4993-5013.
- [53]. Kocher, G., & Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Computing*, 25(15), 9731-9763.
- [54]. Kraiem, M. S., Sánchez-Hernández, F., & Moreno-García, M. N. (2021). Selecting the suitable resampling strategy for imbalanced data classification regarding dataset properties. An approach based on association models. *Applied Sciences*, 11(18), 8546.
- [55]. Kumar, N., & Sharma, S. (2023). A hybrid modified deep learning architecture for intrusion detection system with optimal feature selection. *Electronics*, 12(19), 4050.
- [56]. Latif, S., Idrees, Z., Ahmad, J., Zheng, L., & Zou, Z. (2021). A blockchain-based architecture for secure and trustworthy operations in the industrial Internet of Things. *Journal of Industrial Information Integration*, 21, 100190.
- [57]. Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20), 4396.
- [58]. Liu, H., Zhong, C., Alnusair, A., & Islam, S. R. (2021). FAIXID: A framework for enhancing AI explainability of intrusion detection results using data cleaning techniques. *Journal of network and systems management*, 29(4), 40.

- [59]. Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. *Computers & Security*, 106, 102289.
- [60]. Ma, W., Zhang, Y., Guo, J., & Yu, Q. (2021). Few-shot abnormal network traffic detection based on multi-scale deep-CapsNet and adversarial reconstruction. *International Journal of Computational Intelligence Systems*, 14(1), 195.
- [61]. Magán-Carrión, R., Urda, D., Díaz-Cano, I., & Dorronsoro, B. (2020). Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Applied Sciences*, 10(5), 1775.
- [62]. Mahbubul, I., Mahmudul, I., Shifa, T., Mahfuza, T., & Shima Ali, S. (2022). Exploring The Potential of Visual Narratives and Interactive Gamified Communication Systems to Prevent Disease, Counter Health Misinformation, And Strengthen Public Trust in Low-Literacy Communities: A Literature Review. *American Journal of Interdisciplinary Studies*, 3(02), 165-220. <https://doi.org/10.63125/zmq54n64>
- [63]. Mahfouz, A., Abuhussein, A., Venugopal, D., & Shiva, S. (2020). Ensemble classifiers for network intrusion detection using a novel network attack dataset. *Future Internet*, 12(11), 180.
- [64]. Mahmood, T., Choi, J., & Park, K. R. (2023). Artificial intelligence-based classification of pollen grains using attention-guided pollen features aggregation network. *Journal of King Saud University Computer and Information Sciences*, 35(2), 740-756.
- [65]. Md Anisur Rahman, M. (2024). State Income Tax Differentials and High-Skilled Labor Mobility in The United States: A Time Series Analysis. *American Journal of Interdisciplinary Studies*, 5(04), 222-267. <https://doi.org/10.63125/295bsn47>
- [66]. Md Anisur Rahman, M. (2025). The Role of AI-Enabled Business Intelligence in Financial Risk Assessment and Liquidity Monitoring. *Review of Applied Science and Technology*, 4(04), 195–246. <https://doi.org/10.63125/p4eezd93>
- [67]. Md Arif Uz, Z., Sharmin, S., & Md Abdur, R. (2025). Factors Influencing Behavioural Intention to Use he Innovative Open And Distance Learning Systems With The Role Of Continuance Intention. *American Journal of Scholarly Research and Innovation*, 4(01), 202-220. <https://doi.org/10.63125/pbjxp014>
- [68]. Md Ashfaq, S., & Abdullah Mohammad, S. (2022). CNT/PANI Composite Electrodes for Flexible Fiber-Shaped Dye-Sensitized Solar Cells: Synthesis, Characterization, and Photovoltaic Performance. *American Journal of Interdisciplinary Studies*, 3(02), 221-274. <https://doi.org/10.63125/m334mf41>
- [69]. Md Asif Ali Sheak, A., & Risha, A. (2022). The Role of IOT-Driven Energy Analytics in Improving Operational Efficiency and Sustainability in Modern Industrial Facilities. *American Journal of Interdisciplinary Studies*, 3(04), 807-841. <https://doi.org/10.63125/gm63q956>
- [70]. Md Kaium, H. (2023). A GIS-Based Multi-Criteria Decision-Support and Land-Suitability Optimization Framework for Climate-Resilient Urban and Regional Planning. *American Journal of Data Science and Analytics*, 4(06), 43–103. <https://doi.org/10.63125/wvq0sx87>
- [71]. Md Kaium, H. (2024). Machine Learning Enhanced GIS And AHP Modeling for High-Resolution Flood-Susceptibility Prediction, Vulnerability Assessment, and Risk Classification. *Review of Applied Science and Technology*, 3(02), 92–140. <https://doi.org/10.63125/fz8hv245>
- [72]. Md Kaium, H. (2025). Integrating Artificial Intelligence, Semantic Indoor Geospatial Modeling, And Network-Based Routing for High-Precision NG911 Emergency Location Intelligence. *International Journal of Scientific Interdisciplinary Research*, 6(1), 616–670. <https://doi.org/10.63125/5hc1xx56>
- [73]. Md Mahamud Pervaz, P. (2023). Integration of MES, SAP, And PI Historian Platforms for Real-Time Industrial Asset Performance Management. *Review of Applied Science and Technology*, 2(04), 393–445. <https://doi.org/10.63125/9ebtas97>
- [74]. Md Mahamud Pervaz, P. (2025). Development of a CMMS-Integrated Predictive Maintenance Framework for Industrial PLC and Drive Systems. *American Journal of Scholarly Research and Innovation*, 4(01), 894–942. <https://doi.org/10.63125/j33gtf31>
- [75]. Md. Abdur, R., & Iftekhar, A. (2021). Customer Retention Forecasting in Mobile Wallet Services Using Neural Networks: A Comparative Quantitative Study. *International Journal of Business and Economics Insights*, 1(4), 70-102. <https://doi.org/10.63125/dyrpc387>
- [76]. Md. Rashedul, I., & Md Kaium, H. (2022). Development Of a Three-Dimensional Indoor Geospatial Network and Address Geocoding Framework for Next-Generation Emergency Dispatch and Response-Time Optimization. *Journal of Sustainable Development and Policy*, 1(02), 245-292. <https://doi.org/10.63125/tcw3mp48>
- [77]. Md. Shakhawat, H., & Md, F. (2024). Impact of Phishing Simulation Campaigns and Security Awareness Training on Human Risk Mitigation in Financial Institutions: A Quantitative Study. *American Journal of Data Science and Analytics*, 5(10), 48-85. <https://doi.org/10.63125/4j8h1484>
- [78]. Mehedi, S. T., Anwar, A., Rahman, Z., & Ahmed, K. (2021). Deep transfer learning based intrusion detection system for electric vehicular networks. *Sensors*, 21(14), 4736.
- [79]. Milosevic, M. S., & Ciric, V. M. (2022). Extreme minority class detection in imbalanced data for network intrusion. *Computers & Security*, 123, 102940.
- [80]. Mohammad Abir Chowdhury, S. (2023). Role of AI-Assisted Risk Assessment in Enhancing Audit Planning Efficiency and Risk Prioritization in Public-Safety Organizations. *International Journal of Scientific Interdisciplinary Research*, 4(4), 529–576. <https://doi.org/10.63125/qd6nn437>
- [81]. Mohammad Abir Chowdhury, S. (2025). Application of Machine Learning and Anomaly-Detection Algorithms for Fraud Detection in U.S. Government Financial Operations. *American Journal of Interdisciplinary Studies*, 6(02), 139-184. <https://doi.org/10.63125/fkf19309>

- [82]. Mohammad Abir Chowdhury, S., & Tonay, P. (2022). An Empirical Analysis of the Impact of Robotic Process Automation and Continuous Auditing on the Timeliness And Accuracy Of Financial Reporting Assurance. *International Journal of Business and Economics Insights*, 2(4), 73-126. <https://doi.org/10.63125/7p8dcr47>
- [83]. Mohammad Badrul, A. (2025). SCADA And IOT-Enabled Smart Water Metering: A Systematic Review of Resilient Supply Monitoring. *International Journal of Scientific Interdisciplinary Research*, 6(1), 576–615. <https://doi.org/10.63125/at8pnx81>
- [84]. Mohammad Badrul, A., & Md. Mominul, H. (2023). Machine Learning–Based Assessment of Environmental and Public Health Risks in Sewerage Sludge Management Systems. *American Journal of Advanced Technology and Engineering Solutions*, 3(02), 33-77. <https://doi.org/10.63125/y3yfxa92>
- [85]. Mohammad Shoriful Hossan, S. (2022). A Performance-Based Framework for Arc-Flash Hazard Reduction and Electrical Reliability Improvement in Brownfield Industrial Distribution Systems. *Review of Applied Science and Technology*, 1(04), 447-463. <https://doi.org/10.63125/a70vj720>
- [86]. Mohammadpour, L., Ling, T. C., Liew, C. S., & Aryanfar, A. (2022). A survey of CNN-based network intrusion detection. *Applied Sciences*, 12(16), 8162.
- [87]. Molla Al Rakib, H., & Mohammad Towhidul, I. (2023). Cyber-Resilient Marketing Intelligence Systems for Fraud Detection and Data Integrity in Digital Campaigns. *American Journal of Scholarly Research and Innovation*, 2(02), 485–518. <https://doi.org/10.63125/b53pgr11>
- [88]. Möller, D. P. (2023). Machine learning and deep learning. In *Guide to cybersecurity in digital transformation: trends, methods, technologies, applications and best practices* (pp. 347-384). Springer.
- [89]. Mou, C., Liu, T., Zhu, C., & Cui, X. (2023). Waid: A large-scale dataset for wildlife detection with drones. *Applied Sciences*, 13(18), 10397.
- [90]. Mounir, R., Shahabaz, A., Gula, R., Theuerkauf, J., & Sarkar, S. (2023). Towards Automated Ethogramming: Cognitively-Inspired Event Segmentation for Streaming Wildlife Video Monitoring: R. Mounir et al. *International journal of computer vision*, 131(9), 2267-2297.
- [91]. Mridha, M. F., Das, S. C., Kabir, M. M., Lima, A. A., Islam, M. R., & Watanobe, Y. (2021). Brain-computer interface: Advancement and challenges. *Sensors*, 21(17), 5746.
- [92]. Mst Kaniz, F. (2022). Language Access and Health Equity: Role of Multilingual Administrative Staff in Reducing Healthcare Disparities Among South Asian Immigrant Communities in The U.S. *International Journal of Scientific Interdisciplinary Research*, 1(01), 274–315. <https://doi.org/10.63125/xqhss825>
- [93]. Musial, J. P., & Bojanowski, J. S. (2022). Comparison of the novel probabilistic self-optimizing vectorized earth observation retrieval classifier with common machine learning algorithms. *Remote Sensing*, 14(2), 378.
- [94]. Mvula, P. K., Branco, P., Jourdan, G.-V., & Viktor, H. L. (2023). A systematic literature review of cyber-security data repositories and performance assessment metrics for semi-supervised learning. *Discover Data*, 1(1), 4.
- [95]. Nalmpantis, C., & Vrakas, D. (2019). Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparison. *Artificial intelligence review*, 52(1), 217-243.
- [96]. Naushin, M., Das, A. K., Nayak, J., & Pelusi, D. (2023). Rough-fuzzy based synthetic data generation exploring boundary region of rough sets to handle class imbalance problem. *Axioms*, 12(4), 345.
- [97]. Negera, W. G., Schwenker, F., Debelee, T. G., Melaku, H. M., & Ayano, Y. M. (2022). Review of botnet attack detection in SDN-enabled IoT Using machine learning. *Sensors*, 22(24), 9837.
- [98]. Nishat, J. (2023). Communicable Disease Prevention in a Post-Pandemic Era: A Global Review of Public Health Education, Surveillance, and Community-Level Interventions. *International Journal of Scientific Interdisciplinary Research*, 4(4), 630–678. <https://doi.org/10.63125/mnqefj36>
- [99]. Nishat, J. (2025). Leveraging Machine Learning to Identify and Address Maternal Health Disparities in Underserved U.S. Communities. *American Journal of Data Science and Analytics*, 6(12), 125-169. <https://doi.org/10.63125/c2t7r719>
- [100]. Nishat, J., & Jahanara, A. (2024). Machine Learning Approaches for Early Prediction of High-Risk Pregnancy in Underserved U.S. Populations. *American Journal of Interdisciplinary Studies*, 5(04), 268-309. <https://doi.org/10.63125/5gh5df13>
- [101]. Nishat, J., & Mst Kaniz, F. (2022). A Review of the Impact of Covid-19 On Maternal and Child Health Service Utilization in Bangladesh. *American Journal of Scholarly Research and Innovation*, 1(02), 223–269. <https://doi.org/10.63125/vgv5s742>
- [102]. Nuzhat Sadia, P. (2025). Integrating Business Intelligence Dashboards for Real-Time Supply Chain Visibility and Operational Decision Support. *American Journal of Data Science and Analytics*, 6(12), 170-216. <https://doi.org/10.63125/1es4f763>
- [103]. Nwakanma, C. I., Ahakonye, L. A. C., Njoku, J. N., Odirichukwu, J. C., Okolie, S. A., Uzundu, C., Ndubuisi Nweke, C. C., & Kim, D.-S. (2023). Explainable artificial intelligence (XAI) for intrusion detection and mitigation in intelligent connected vehicles: A review. *Applied Sciences*, 13(3), 1252.
- [104]. Odeh, A., & Abu Taleb, A. (2023). Ensemble-based deep learning models for enhancing IoT intrusion detection. *Applied Sciences*, 13(21), 11985.
- [105]. Oliveira, N., Praça, I., Maia, E., & Sousa, O. (2021). Intelligent cyber attack detection and classification for network-based intrusion detection systems. *Applied Sciences*, 11(4), 1674.
- [106]. Panigrahi, R., Borah, S., Bhoi, A. K., Ijaz, M. F., Pramanik, M., Jhaveri, R. H., & Chowdhary, C. L. (2021). Performance assessment of supervised classifiers for designing intrusion detection systems: a comprehensive review and recommendations for future research. *Mathematics*, 9(6), 690.
- [107]. Panigrahi, R., Borah, S., Bhoi, A. K., Ijaz, M. F., Pramanik, M., Kumar, Y., & Jhaveri, R. H. (2021). A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets. *Mathematics*, 9(7), 751.

- [108]. Rabbani, M., Wang, Y., Khoshkangini, R., Jelodar, H., Zhao, R., Bagheri Baba Ahmadi, S., & Ayobi, S. (2021). A review on machine learning approaches for network malicious behavior detection in emerging technologies. *Entropy*, 23(5), 529.
- [109]. Rao, Y. N., & Suresh Babu, K. (2023). An imbalanced generative adversarial network-based approach for network intrusion detection in an imbalanced dataset. *Sensors*, 23(1), 550.
- [110]. Rey, V., Sánchez, P. M. S., Celdrán, A. H., & Bovet, G. (2022). Federated learning for malware detection in IoT devices. *Computer networks*, 204, 108693.
- [111]. Rill-García, R., Dokladalova, E., & Doklád, P. (2022). Pixel-accurate road crack detection in presence of inaccurate annotations. *Neurocomputing*, 480, 1-13.
- [112]. Rodríguez, M., Alesanco, Á., Mehavilla, L., & García, J. (2022). Evaluation of machine learning techniques for traffic flow-based intrusion detection. *Sensors*, 22(23), 9326.
- [113]. Sadeghi, F., Viktor, H. L., & Vafaie, P. (2023). DynaQ: online learning from imbalanced multi-class streams through dynamic sampling: F. Sadeghi et al. *Applied Intelligence*, 53(21), 24908-24930.
- [114]. Sadia, Z. (2025). Predictive Analytics for Financial Performance Forecasting and Reporting Quality: Evidence from Data-Driven Financial Management Systems. *ASRC Procedia: Global Perspectives in Science and Scholarship*, 1(01), 2795–2812. <https://doi.org/10.63125/zstqn346>
- [115]. Sauber-Cole, R., & Khoshgoftaar, T. M. (2022). The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey. *Journal of Big Data*, 9(1), 98.
- [116]. Schmitt, M. (2023). Securing the digital world: Protecting smart infrastructures and digital industries with artificial intelligence (AI)-enabled malware and intrusion detection. *Journal of Industrial Information Integration*, 36, 100520.
- [117]. Shaiful, M., Anisur, R., & Md, A. (2022). A Systematic Literature Review on the Role of Digital Health Twins in Preventive Healthcare For Personal and Corporate Wellbeing. *American Journal of Interdisciplinary Studies*, 3(04), 1-31. <https://doi.org/10.63125/negjw373>
- [118]. Sharma, D., & Selwal, A. (2022). An intelligent approach for fingerprint presentation attack detection using ensemble learning with improved local image features. *Multimedia Tools and Applications*, 81(16), 22129-22161.
- [119]. Sharma, D., & Selwal, A. (2023). A survey on face presentation attack detection mechanisms: hitherto and future perspectives. *Multimedia Systems*, 29(3), 1527-1577.
- [120]. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60.
- [121]. Silva, J. V. V., de Oliveira, N. R., Medeiros, D. S., Lopez, M. A., & Mattos, D. M. (2022). A statistical analysis of intrinsic bias of network security datasets for training machine learning mechanisms. *Annals of Telecommunications*, 77(7), 555-571.
- [122]. Stavrinides, G. L., & Karatza, H. D. (2019). An energy-efficient, QoS-aware and cost-effective scheduling approach for real-time workflow applications in cloud computing systems utilizing DVFS and approximate computations. *Future Generation Computer Systems*, 96, 216-226.
- [123]. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., Ghogho, M., & El Moussa, F. (2020). DeepIDS: Deep learning approach for intrusion detection in software defined networking. *Electronics*, 9(9), 1533.
- [124]. Taye, M. M. (2023). Understanding of machine learning with deep learning: architectures, workflow, applications and future directions. *Computers*, 12(5), 91.
- [125]. Thapa, C., Karmakar, K. K., Celdran, A. H., Camtepe, S., Varadharajan, V., & Nepal, S. (2021). FedDICE: A ransomware spread detection in a distributed integrated clinical environment using federated learning and SDN based mitigation. *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*,
- [126]. Thapa, N., Liu, Z., Kc, D. B., Gokaraju, B., & Roy, K. (2020). Comparison of machine learning and deep learning models for network intrusion detection systems. *Future Internet*, 12(10), 167.
- [127]. Tonay, P. (2025). Machine Learning for Fraud Detection in Enterprise Financial Systems. *American Journal of Advanced Technology and Engineering Solutions*, 1(02), 304-318. <https://doi.org/10.63125/d0fhp873>
- [128]. Tonay, P., Md Maruf, I., & Al, A. (2024). Artificial Intelligence-Based Decision Support Systems and Managerial Performance. *American Journal of Advanced Technology and Engineering Solutions*, 4(04), 154-184. <https://doi.org/10.63125/wmz8dc67>
- [129]. Tsimenidis, S., Lagkas, T., & Rantos, K. (2022). Deep learning in IoT intrusion detection. *Journal of network and systems management*, 30(1), 8.
- [130]. Uddin, H. M. M., & Risha, A. (2024). Hybrid Cloud Security and Compliance in U.S. Enterprises: Addressing Data Privacy Risks and Governance Challenges. *American Journal of Scholarly Research and Innovation*, 3(01), 153-197. <https://doi.org/10.63125/qcy0cj46>
- [131]. Ullah, S., Khan, M. A., Ahmad, J., Jamal, S. S., e Huma, Z., Hassan, M. T., Pitropakis, N., Arshad, & Buchanan, W. J. (2022). HDL-IDS: a hybrid deep learning architecture for intrusion detection in the Internet of Vehicles. *Sensors*, 22(4), 1340.
- [132]. Vishwakarma, M., & Kesswani, N. (2022). DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT. *Decision Analytics Journal*, 5, 100142.
- [133]. Wang, M., Yang, N., Gunasinghe, D. H., & Weng, N. (2023). On the robustness of ML-based network intrusion detection systems: An adversarial and distribution shift perspective. *Computers*, 12(10), 209.
- [134]. Wang, Y.-C., Houng, Y.-C., Chen, H.-X., & Tseng, S.-M. (2023). Network anomaly intrusion detection based on deep learning approach. *Sensors*, 23(4), 2171.

- [135]. Wu, H., Zhang, Y., Liang, L., Mei, X., Han, D., Han, B., Weng, T.-H., & Li, K.-C. (2023). Multi-head attention-based model for reconstructing continuous missing time series data: H. Wu et al. *The Journal of Supercomputing*, 79(18), 20684-20711.
- [136]. Xue, Z., Yang, F., Rajaraman, S., Zamzmi, G., & Antani, S. (2023). Cross dataset analysis of domain shift in cxr lung region detection. *Diagnostics*, 13(6), 1068.
- [137]. Yao, R., Wang, N., Chen, P., Ma, D., & Sheng, X. (2023). A CNN-transformer hybrid approach for an intrusion detection system in advanced metering infrastructure. *Multimedia Tools and Applications*, 82(13), 19463-19486.
- [138]. Zakariah, M., & Almazyad, A. S. (2023). Anomaly detection for IoT systems using active learning. *Applied Sciences*, 13(21), 12029.
- [139]. Zakariah, M., & Alotaibi, Y. A. (2023). Unipolar and bipolar depression detection and classification based on actigraphic registration of motor activity using machine learning and uniform manifold approximation and projection methods. *Diagnostics*, 13(14), 2323.
- [140]. Zhang, Y., & Wang, Z. (2023). Feature engineering and model optimization based classification method for network intrusion detection. *Applied Sciences*, 13(16), 9363.
- [141]. Zhu, S., He, P., Zhou, J., Zhang, P., Jin, Z., Jiang, Y., Buchmann, R., Bi, Y., Ghiran, A., & Ma, W. (2023). Knowledge science, engineering and management.