International Journal of Emerging Engineering and Technology (IJEET)
ISSN (e): 2958-3764
Volume: 4, Number: 2, Pages: 104 - 110, Year: 2025

AI-Based Network Intrusion Detection System for Enterprise and Cloud-based Infrastructures

Malik Adnan[1,*], Abdul Sattar [2], Hamza Azeem [1], and Irshad Ahmed Sumra [1]

[1] School of Systems and Technology, University of Management and Technology Lahore, 54770, Pakistan.
[2]Department of Computer Science, Lahore Garrison University, Lahore. Pakistan
[*] Corresponding author: Malik Adnan (Email: f2021408049@umt.edu.pk)

Received: 02/03/2025, Revised: 19/05/2025, Accepted: 13/06/2025

***Abstract*— The proliferation of enterprise networks, distributed cloud structures, and the Internet of Things (IoT) has increased the dangers and vulnerabilities of digital infrastructure to sophisticated cyberattacks. Traditional Network Intrusion Detection Systems (NIDS) rely solely on static signature matching, which can't detect new zero-day attacks, and standalone systems that use anomaly detection techniques suffer from high false-positive rates that can overload Security Operations Centres (SOCs). One of the main problems is accurately identifying threats while maintaining the detection system's adaptability and scalability, which challenge is increasingly addressed by leveraging Artificial Intelligence (AI) and deep learning. The literature review and synthesis are systematic and conducted using peer-reviewed literature on the topic from 2018 to 2026, following an explicit methodology aligned with the PRISMA guidelines. The takeaway messages are that cost-sensitive and deep hybrid architectures outperform both the cost-sensitive and cost-ignorant benchmarks, but their effectiveness critically depends on model explainability, evaluation rigour, and threshold tuning. Lastly, this survey summarizes some of the biggest challenges and future research directions in explainable AI (XAI), federated learning, as well as lightweight and resource-aware modeling for network security.**

***Index Terms*—Cybersecurity, Artificial Intelligence, Deep Learning, Network Intrusion Detection System, Anomaly Detection, Technology Acceptance Model.**

I. INTRODUCTION

As networks grow in number and communication environments become distributed, networked, the threats to cybersecurity in modern infrastructures have become much more sophisticated [1]. With cloud computing services, remote mobile access pipelines, and enterprise Internet of Things (IoT) deployments, the number of logical layers in the standard network stack has grown [2],[3]. Ransomware propagation, Distributed Denial of Service (DDoS), stealthy data exfiltration, Distributed Memory Denial of Service (DMDoS), and zero-day exploits are just a few of the multi-stage attacks adversarial actors often use to threaten catastrophic compliance, economic and operational risks for public enterprise systems and critical infrastructure [4]. Such dynamic techniques are beyond the, requiring a much broader scope of traditional defence techniques like firewalls and access control policies, and Network Intrusion Detection Systems (NIDS), therefore have become very important in contemporary defensive security architectures [5].

The conventional method for IDS is to compare events occurring in the network against a static database of known malicious events and patterns [6],[7]. The rule-based engines work very well at detecting known threats but cannot detect changes in threat signatures (zero-day attacks) or new threats (fast-changing threat variants) [8]. Anomaly-based systems, on the other hand, maintain a model of normal background traffic in the network and can detect statistical deviations that, in theory, can indicate new attacks [9], [5]. However, these anomaly detectors often suffer from high false-positive rates due to class imbalance, diverse traffic sources and environmental variations, and the overload of analysts' workflows and diminished trust in security operations centres [10], [11].

To overcome the limitations of the signature and anomaly paradigms, researchers have investigated hybrid architectures that combine the best features of the two paradigms: a signature-based approach and machine learning (ML), including deep learning (DL) [4],[12]. AI can be used to learn from complex representations of features from a high-dimensional network telemetry space, allowing the model to adapt to new infrastructure footprints and generalise to subtle changes in the way a threat is executed, as advanced NIDS does [1],[3]. But as this literature has increased, there has also been a gap between the idealised laboratory performance and enterprise use of the technology that has remained constant [10]. Models trained on benchmark datasets may not be general enough to handle artefacts seen in real production traffic, may not be easily explainable for automated containment decisions, and can consume significant compute resources, making them impractical to deploy online in production [13],[14]. This work fills several gaps in previous surveys on this topic and is presented with a focus on them. Buczak and Guven give a useful early review of intrusion detection techniques using data mining and machine learning, but the techniques are not based

on the deep learning architectures now generally used in NIDS research. While [15-17] compares a variety of deep-learning models on intrusion-detection datasets and provides useful performance data, it only compares deep architectures and does not consider deployment aspects of shallow or hybrid architectures, or the organisational and human-trust-related aspects of deployment. A survey of detection methods and architectures is complementary to, but distinct from, a survey of network-based intrusion-detection datasets done by Ring et al. This work builds on these contributions, and extends those of the previously mentioned surveys in three ways: (1) it covers shallow, deep, and hybrid architectures within a single, unified comparative framework, rather than treating them as discrete areas, as most previous surveys did; (2) it was created to cover developments since most previous surveys, including explainable AI, federated learning, and adversarial robustness, up to 2026; and (3) it explicitly introduces organizational adoption factors, mediated by the Technology Acceptance Model, which connects technical detection performance to the practical question if and when security operations teams will trust and act on AI generated alerts.

The following are the major contributions of this survey. The analysis of performance frameworks, the effects of false alarms in the operational environment and the importance of using a balanced set of performance measures are discussed [18], [19]. Second, user trust is explored using the Technology Acceptance Model (TAM) and a discussion on open research questions and future trends like Explainable AI (XAI) and Federated Learning for enterprise deployment at scale is presented [20],[13] [21-34].

II. RESEARCH METHODOLOGY: PRISMA-BASED LITERATURE REVIEW

To ensure the comprehensiveness, objectivity and repeatability of surveying, this survey follows the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) guidelines, in line with the principles of findable, accessible, interoperable and reproducible (FAIR) research data management [35]. The selection process was in four stages: . Identification, Screening, Eligibility Assessment, and Inclusion. The automated keyword-based queries were performed during the identification stage across the main digital repositories and citation databases: IEEE Xplore, SpringerLink, ACM Digital Library, Elsevier ScienceDirect, and Google Scholar. The initial search results that were returned for this search (2018-2026 only) were [authors: insert total number of records retrieved].

In the screening stage, 318 duplicate records were removed, leaving 806 unique records for title and abstract screening. Records were excluded at this stage if they did not propose, evaluate, or survey an AI-, ML-, or DL-based intrusion- or anomaly-detection method; were not written in English; were not peer-reviewed; or fell outside the 2018–2026 publication window. This stage excluded 694 records, leaving 112 for full-text eligibility assessment. In the eligibility stage, the remaining records were subjected to full-text review, where data were systematically extracted and categorised into author/year identifiers, specific algorithmic methods, datasets used, types of attacks evaluated, performance metrics, and identified limitations [12]. This review focused on the rigour of the validation reported, the presence of possible data-leakage artefacts, the cross-validation strategies used, and the specific treatment of severe class imbalance [22],[23]. In the inclusion stage, 36 records meeting all eligibility criteria were retained. The final core collection comprises the 36 studies that form the technical foundation of the taxonomy and comparative analysis presented in this survey.

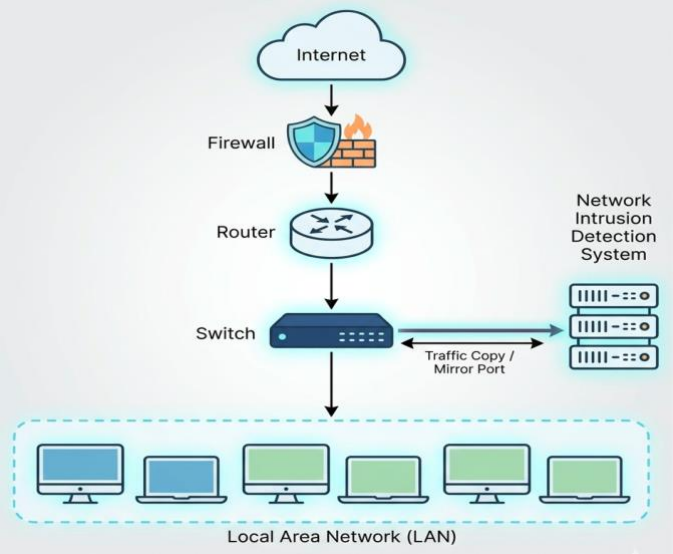

Fig. 1 NIDS real time traffic mirror placement via architecture.

III. BACKGROUND

The enterprise network is now a more and more massive and intricate space that includes on-premises LANs, private data centers, multi-cloud platforms, remote end user endpoints and industrial IoT environments [2],[24]. This type of configuration is referred to as a continuous passive/inline monitoring configuration where several different visibility vectors are collected [1]. This monitoring has two types of visibility vectors: host-based or network-based. Host-based intrusion detection systems (HIDS) monitor the internal activities of the machine, application life cycles, application logs and system calls. In contrast, a network-based system (NIDS) (Fig. 1) stores line-rate data packets, physical interface logs, and transport flow records, and can monitor traffic patterns between network segments without any impact on any endpoint hosts [5],[8].

The AI models process collected raw network telemetry through a complex data processing pipeline to enable the models to respond to the raw data. Here, the first part of the packet capture is to capture raw packets or the flow summary of the network interface. Secondly, to prevent the model from becoming unstable in the training process, these values are used in a normalization engine. Finally, the AI detection engine compares the prepped features to create binary or multi-class alerts that are then passed on to the security operation center for analyst triage, enrichment and incident response (Fig. 2).

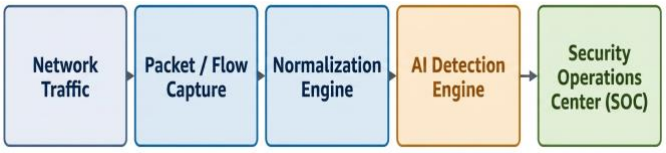

Fig. 2. Overall structure of an AI-based network intrusion detection system.

In enterprise environments, there is no such thing as a normal background traffic; it's always moving depending on the corporate patching cycle, business cycle, rollouts of networks and the evolving mobility patterns of the users [10]. This background noise leads to concept drift in the sense that the distribution of data from the past differs from the distribution of data in the current data traffic through the system and thus causes false alarms and model inaccuracies [25],[6]. This variation is important because it means that for each traffic regime an enterprise traffic regime may be different, which means the configuration that is effective for one traffic regime may not be effective for a different one [14]. In this regard, the contemporary systems need to be updated periodically with security operations and continuously re-optimized to provide an adequate level of visibility of the threats taking place, not overloading the security operations [25].

IV. CLASSIFICATION

In the literature, there are several existing systems based on artificial intelligence in Network Intrusion Detection that can be classified to a multi-layer taxonomy based on algorithmic structure, learning type and the way network data is processed. This taxonomy [4],[12] can be used to classify the different strategies that have been devised to balance detection sensitivity with false alarm rates. Shallow machine learning algorithms that are explicitly engineered with features to define decision boundaries is the first branch [1]. The classifiers are Support Vector Machines (SVM), Random Forests (RF), K-Nearest Neighbors (KNN) and Naive Bayes classifiers. Because of their low computational cost, high inference speed and lower number of hyperparameters, these kinds of models are widely used in the edge and resource-constrained applications. Shallow models are not always able to model non-linear relationships, however, and they tend to perform poorly when the number of features is high or when they are presented with a zero-day variant in enterprise traffic with high volume [12].

The second branch is focused on deep learning architectures, which learn the features and do the classification in a single pipeline, without having to pick the features by hand, as in the first one [4]. In this category are categorized networks, including Convolutional Neural Networks (CNN) for spatial feature maps, Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) blocks for temporal dependencies and generative networks like Generative Adversarial Networks (GAN) [26],[3]. In addition, unsupervised deep models (such as Deep Autoencoders (AE) and Stacked Autoencoders) are generally used for anomaly detection [27],[34]. The architectures map high dimensional telemetry to a low dimensional "bottleneck" layer, and the difference between the original and reconstructed layers are used as an anomaly indicator for discovering previously unseen attack signatures [9].

The third branch is hybrid intelligent models, which use multiple intelligent detection methods to help alleviate their respective limitations [12]. A popular hybrid model is a combination of an unsupervised deep autoencoder and a shallow downstream classifier such as a Random Forest or an SVM [27]. Here, the AE is employed as a non-symmetric feature extraction algorithm to train structural representations that are captured in the AE latent space and the shallow classifier is used on extracted embeddings to perform the classification task in low latency. The integration allows the system to rapidly filter out known threats from the traffic stream and the remaining traffic to deep learning engine for detection of new combinations or low prevalence anomalies [6].

V. COMPARATIVE ANALYSIS

To evaluate AI-based intrusion detection objectively, performance metrics should be evaluated with common datasets and the performance of different approaches on various enterprise and cloud performance benchmarks [4],[21],[15] should be studied (Table I).

Shallow machine-learning approaches usually have a clear advantage in terms of the ease with which prediction can be made and in terms of computation efficiency [1]. To the best of our knowledge, the optimised Random Forest configuration has been proven to achieve high accuracy in the high-prevalence attack classes, such as Generic (DoS) attacks, on the datasets used in this study, including UNSW-NB15 [16]. However, shallow classifiers are not very sensitive to multi-stage attack classes with low prevalence, such as Backdoors, Worms, or targeted Analysis exploits, most often because standard training loops are better suited to the majority background traffic [23].

The goal of deep learning research has been to address these vulnerabilities by learning complex temporal relationships [4]. The LSTM and Gated Recurrent Unit (GRU) architectures have been used to analyse the sequential connection traces to identify the scan-to-exploit sequences that take place over extended time periods [3].

Four studies were chosen as representative, and in-depth side by side comparison was conducted in Table 1, from the larger number of studies that passed the screening process described in Section II. Three explicit criteria were used to guide the selection beyond the number of citations: (i) architectural diversity, to ensure that studies included shallow-deep hybrid, pure benchmarking, anomaly-based and class-imbalance-focused architectures; (ii) methodological completeness, including studies that mentioned reproducible evaluation protocols and quantitative results, while excluding studies with incomplete experimental detail; and (iii) relevance to enterprise

and cloud-facing deployment scenarios, the operationally-focused domain of this survey. It is not intended to assert that these four studies were necessarily the best of the other studies reviewed, but rather to show the spectrum of design trade-offs found in the literature.

VI. CHALLENGES

There are a number of technical and operational hurdles for implementing an AI-powered network intrusion detection system:

- In production enterprise networks, the amount of benign background traffic is extremely large and malicious activities are very few anomalies [11]. Classification by using the traditional loss functions may lead to over-classification and under-detection of low prevalence or stealth attacks [23].
- However, concept drift and baseline shift are non-static in enterprise scenarios due to frequent software changes, moving to cloud computing and evolving business cycles [10]. This continuous evolution changes traffic distributions from the baseline and causes the performance to degrade with time, if the models are trained on a static traffic with a fixed distribution [25].
- Data leakage and data validation bias are common in many research validation designs, such as random splits across records from duplicated datasets [22], [32]. If the exact same packet occurs in the training set and in the test set, this practice can result in over-optimistic benchmark scores which will not be reflective of true generalization to real-world situations [14].
- Multiple layers in deep learning models greatly require memory and compute resources [4]. This computation load causes latency problems and limits application to line-rate speeds in enterprise cores [31].
- The models trained on a limited public dataset have the disadvantage of performing poorly when confronted with new attack vectors not seen during training, or when deployed in other network topologies than those the model was trained on [10].
- Anomaly alerts can be raised during routine internal vulnerability scanning and automated data backups, even if they are legitimate and thus a source of alert fatigue and elevated false positives [6]. This surge of alerts may cause severe alert fatigue that would consume analyst time and resources, and may cause analysts to miss critical security events [11].
- Advanced deep learning architectures can be opaque: Sometimes these architectures do not give enough transparency on how they make a specific classification decision [13]. Without an explanatory context, security analysts will not be able to easily confirm alerts and this will extend the triage process and hinder it in the environment of risk [20].
- There are no standardized testing frameworks to measure the ability of a trained model to perform well on a completely new enterprise network or different sets of encryptions [10],[14].
- AI models that only use network flow records without considering the broader context of the system are available [25]. A clear need for frameworks that actively correlate network flow anomalies to host logs, authentication events, endpoint states and threat intelligence feeds to validate and improve on the accuracy of the correlation [10].
- No Standardized Workload-Aware Metrics: Many metrics used to evaluate the performance of a model are purely mathematical, focusing on the accuracy or ROC-AUC score, without considering operational aspects like analyst triage time, or the cost of a false positive or false negative to the business [28].
- The lack of Non-Disruptive Update Mechanisms: The problem of concept drift is known but current approaches usually involve retraining models from scratch [6]. Studies on continuous learning methods that allow to update models without interfering with running security monitoring flows are still scarce [25].
- Advanced adversary can either intentionally change the timing interval or the pattern of packets passed between them so that no attack can be detected, but the pattern of packets is still changed to deliver the attack payload [26]. Adversarial stress tests and red team testing are usually not included in the development of traditional NIDS [14].
- Future work should involve incorporating interpretability frameworks with them directly into deep learning pipelines [13]. Clear explanations provided to analysts regarding which specific feature anomalies raised an alert can greatly speed up triage time and increase the level of trust in the system [20] [29].
- Distributed Federated Learning Models: Many organizations can share an intrusion detection model, but without centralizing any network logs as described in Privacy-Preserving Federated Learning [3]. This allows for an identification of the global threat vector without compromising compliance and privacy of local data, but rather in a joint effort.
- Optimization and lightening up the architecture including model pruning, quantization, and knowledge distillation are also crucial. These optimizations are all aimed at reducing the latency impact of deep neural networks to be deployed directly onto the edge devices or fast core routers [2],[30].

- Future NIDS will be enhanced by leveraging Graph Neural Networks (GNNs) and being able to incorporate a model of semantic relationships between network entities, asset criticality, and user accounts using context-enrichment engines [25]. This structural approach can be used to eliminate irrelevant anomalies and improve the quality of alerts, if they correspond to multi-stage attack scenarios [10].
- Continuous Red-Teaming: Adversarial Hardening: Embed adversarial training loops directly into the training procedure to enhance the system's resistance to evasion attacks [26]. Having a standard procedure for validation, including continuous red team simulation testing, would be helpful to more closely ensure the claims of performance against adaptive real-world adversaries [14].
- Applications such as intrusion detection and analyst support, including the ability to classify flow records as structured text, summarization of alerts in natural language, and conversational threat-hunting interface support are increasingly being addressed with the use of large language models and transformer-based language models [36]. These models can provide a significant compute advantage, have the potential to render more explainable models, and demonstrate the ability to detect, but the latency of inference, the risk of hallucination and significant compute requirement at present makes adoption of these models more feasible in post-detection triage and reporting than in the detection path itself[37].
- Beyond inference latency, there are even more practical factors to consider when implementing AI based NIDS as an on-line line rate solution in enterprise or cloud environments, such as how to stream extract features under packet-rate pressure, how to update and roll back models without disrupting active monitoring, and how to gracefully degrade when compute resources are saturated during traffic surges[38]. The level of detection accuracy is not considered in the current literature with these operational requirements, and they provide a clear line of research for future empirical studies.

VII. Conclusion

The authors suggest that it is better to view AI-driven network intrusion detection as a socio technical pipeline of decision support than a classification exercise based on this survey. While deep and hybrid machine learning architectures show great promise for identifying hidden threats in vast amounts of data in enterprise telemetry, they must be based on representativeness, validation and the tuning of thresholds. It is a systematic review of recent literature from 2018-2026 which establishes a direct link between technical metrics and operational workloads and the explainability of the models and user trust. The field must now shift from individual benchmark performance to more complete, well-documented and effective deployment systems. By emphasizing explainable models, context-awareness, and robust validation procedures, next-generation AI-NIDS can offer lasting security for today's enterprise and cloud environments.

Funding Statement

The authors received no specific funding for this study.

Conflicts of Interest

The authors declare no conflicts of interest to report regarding the present study.

Author Contributions

All authors contributed to the conception, literature review, drafting, and critical revision of this manuscript and approved the final version for submission.

Data Availability Statement

Data is available on reasonable request.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

References

[1] E. G. A. L. Buczak, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE, Piscataway, NJ, USA, 2016.

[2] T. A. N. e. al., "A comparative evaluation of intrusion detection systems on the edge-IIoT-2022 dataset," Elsevier, Amsterdam, Netherlands, 2023.

[3] N. C. A. A. Diro, "Distributed attack detection scheme using deep learning approach for Internet of Things," Elsevier, Amsterdam, Netherlands, 2018.

[4] E. S. P. V. V. U. T. P. Mishra, "Intrusion detection techniques in cloud environment: A survey," Elsevier, Amsterdam, Netherlands, 2017.

[5] L. M. S. M. H. J. M. A. Ferrag, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Elsevier, Amsterdam, Netherlands, 2020.

[6] J. D.-V. G. M.-F. E. V. P. García-Teodoro, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Elsevier, Amsterdam, Netherlands, 2009.

[7] V. S. N. Hubballi, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," Elsevier, Amsterdam, Netherlands, 2014.

[8] B. I. S. A. R. Shah, "Performance comparison of intrusion detection systems and application of machine learning to Snort system," Elsevier, Amsterdam, Netherlands, 2018.

[9] J.-M. P. A. Patcha, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Elsevier, Amsterdam, Netherlands, 2007.

[10] A. B. V. K. V. Chandola, "Anomaly detection: A survey," ACM, New York, NY, USA, 2009.

[11] V. P. R. Sommer, "Outside the closed world: On using machine learning for network intrusion detection," IEEE Computer Society, Oakland, CA, USA, 2010.

[12] A. T. T. Pietraszek, "Data mining and machine learning — Towards reducing false positives in intrusion detection," Elsevier, Amsterdam, Netherlands, 2005.

[13] J. S. S. Gamage, "Deep learning methods in network intrusion detection: A survey and an objective comparison," Elsevier, Amsterdam, Netherlands, 2020.

[14] J. B. O. A. W. R. M. A. S. R. C. H. O. A. M. G. Rjoub, "A survey on explainable artificial intelligence for cybersecurity," IEEE, Piscataway, NJ, USA, 2023.

[15] M. V. S. K. A. A. B. D. P. A. Milenkoski, "Evaluating computer intrusion detection systems: A survey of common practices," ACM, New York, NY, USA, 2015.

[16] G. J. D. Chicco, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BioMed Central (Springer Nature), London, UK, 2020.

[17] M. G. J. Davis, "The relationship between Precision-Recall and ROC curves," ACM, Pittsburgh, PA, USA, 2006.

[18] R. P. Bagozzi, "The legacy of the technology acceptance model and a proposal for a paradigm shift," Association for Information Systems, Atlanta, GA, USA, 2007.

[19] M. D. W. e. al., "The FAIR guiding principles for scientific data management and stewardship," Nature Portfolio, London, UK, 2016.

[20] J. H. J. S. N. Moustafa, "A holistic review of network anomaly detection systems: A comprehensive survey," Elsevier, Amsterdam, Netherlands, 2019.

[21] N. B. N. Chergui, "Contextual-based approach to reduce false positives," Institution of Engineering and Technology (IET), London, UK, 2020.

[22] F. R. B. Biggio, "Wild patterns: Ten years after the rise of adversarial machine learning," Elsevier, Amsterdam, Netherlands, 2018.

[23] T. N. N. V. D. P. Q. S. N. Shone, "A deep learning approach to network intrusion detection," IEEE, Piscataway, NJ, USA, 2018.

[24] J. H. F. Farahnakian, "A deep auto-encoder based approach for intrusion detection system," Global IT Research Institute (GIRI) / IEEE, PyeongChang, South Korea, 2018.

[25] S. W. D. S. D. L. A. H. M. Ring, "A survey of network-based intrusion detection data sets," Elsevier, Amsterdam, Netherlands, 2019.

[26] A. H. L. A. A. G. I. Sharafaldin, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," SCITEPRESS, Funchal, Madeira, Portugal, 2018.

[27] J. S. N. Moustafa, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," IEEE, Canberra, Australia, 2015.

[28] A. H. G. A. Telikani, "Cost-sensitive stacked autoencoders for intrusion detection in the Internet of Things," Elsevier, Amsterdam, Netherlands, 2019.

[29] G. Varoquaux, "Cross-validation failure: Small sample sizes lead to large error bars," Elsevier, Amsterdam, Netherlands, 2018.

[30] P.-F. G. Y. H. F. M. L. M. É. T. M. Lanvin, "Errors in the CICIDS2017 dataset and the significant differences in detection performances it makes," Springer, Cham, Switzerland, 2023.

[31] P. L. E. M. d. O. W. M. L. B. R. A. V. G. F. D. F. G. Apruzzese, "The role of machine learning in cybersecurity," ACM, New York, NY, USA, 2023.

[32] T. Fawcett, "An introduction to ROC analysis," Elsevier, Amsterdam, Netherlands, 2006.

[33] R. L. A. Thakkar, "A review of the advancement in intrusion detection datasets," Elsevier, Amsterdam, Netherlands, 2020.

[34] C.-H. R. L. Y.-C. L. K.-Y. T. H.-J. Liao, "Intrusion detection system: A comprehensive review," Elsevier, Amsterdam, Netherlands, 2013.

[35] H. Kheddar, "Transformers and large language models for efficient intrusion detection systems: A comprehensive survey," Elsevier, Amsterdam, Netherlands, 2025.

[36] E. B. W. L. A. A. G. M. Tavallaee, "A detailed analysis of the KDD CUP 99 data set," IEEE, Ottawa, ON, Canada, 2009.

[37] H. Hasbullah, I. A. Soomro, and J. L. Ab Manan, “Denial of service (DOS) attack and its possible solutions in VANET,” International Journal of Electronics and Communication Engineering, vol. 4, no. 5, pp. 813–817, 2010.

[38] [38]Sumra, I.A., Hasbullah, H.B., AbManan, Jl.B. (2015). Attacks on Security Goals (Confidentiality, Integrity, Availability) in VANET: A Survey. In: Laouiti, A., Qayyum, A., Mohamad Saad, M. (eds) Vehicular Ad-hoc Networks for Smart Cities. Advances in Intelligent Systems and Computing, vol 306. Springer.

TABLE I: COMPARATIVE SUMMARY OF THE FOUR MOST RELEVANT AI-BASED NIDS STUDIES

Author Name / Year	Proposed Method	Working of Model	Limitations
Shone et al. (2018)	They used a Non-symmetric Deep Autoencoder (NDAE) in combination with a Random Forest (RF) classifier in a two-stage system. First, the autoencoder learns a non-linear latent representation of the high dimensional network traffic features in an unsupervised manner. Second, the Random Forest is trained using this subset of features to classify the traffic as benign or malicious, thus reducing computation costs without compromising classification accuracy.	• The AE does a compression of the network features. • The Random Forest is used to classify traffic that has a low latency. • Provides the highest survey accuracy-speed balance.	• Used old/redundant data (KDD Cup 99) [17]. • Unable to extrapolate from one set of data to another. •The incorporation of the validation of real enterprise traffic.
Ferrag et al. (2020)	The authors are not proposing a single model, but they did perform a comparative benchmarking study across a broad spectrum of deep learning architectures such as CNNs, RNNs, LSTMs, GANs, and autoencoders, all with the same network traffic datasets. This is an overall, systematic and side-by-side comparison that identifies the architectural families that are performing best during the same experimental conditions for detection.	• Most complete benchmark in the survey. • Tested the largest array of deep learning architectures and datasets in one shot.	•No competition between the MIM and the NCP. •Did not consider extreme network class imbalance. • The risk of loss of data is high because the data is split randomly.
Moustafa et al. (2019)	The authors present a comprehensive analysis of network anomaly-detection systems and combine the findings from the literature to examine the capabilities of signature-based, anomaly-based, and hybrid NIDS in dynamic environments like cloud and IoT networks. They note that approaches using a static notion of "normal traffic" are especially prone to high false-positive rates when the notion of "normal traffic" changes, which is often the case in elastic cloud workloads, and cite two recurring open challenges that they found in the literature: dataset quality and evaluation in real-time.	• Targeted and specific in terms of cloud infrastructure security. • Supports various traffic patterns, such as dynamic baselines and multi-tenant traffic.	•Low attack traffic (<0.01%). Does not adapt to concept drift in the long term.
Telikani and Gandomi (2019)	They suggested an intrusion detection system based on a cost-sensitive stacked autoencoder in IoT traffic. Network traffic is very skewed, with many more benign records than attack records, and with standard loss functions being able to attain a high overall accuracy while still erring on the side of the minority class attacks. This is overcome by the proposed method by giving more of a penalty to minority attack classes during training, thus the model learns to correctly identify rare but high-impact attack classes instead of optimizing for overall accuracy.	•Directly addresses class imbalance with cost-weighting loss. • Dramatically enhances recall of the rare or minority class attack.	•Threshold tuning needs to be done manually and is dependent on the dataset. •Not well understood for its generality over varying traffic distributions.