# Network Intrusion Detection: Evolution from Conventional Approaches to LLM Collaboration and Emerging Risks

Yaokai Feng
Faculty of Information Science and Electrical
Engineering, Kyushu University, Japan
Email: fengyk@ait.kyushu-u.ac.jp

Kouichi Sakurai
Faculty of Information Science and Electrical
Engineering, Kyushu University, Japan
Email: sakurai@inf.kyushu-u.ac.jp

*Abstract*—**This survey systematizes the evolution of network intrusion detection systems (NIDS), from conventional methods such as signature-based and neural network (NN)-based approaches to recent integrations with large language models (LLMs). It clearly and concisely summarizes the current status, strengths, and limitations of conventional techniques, and explores the practical benefits of integrating LLMs into NIDS. Recent research on the application of LLMs to NIDS in diverse environments is reviewed, including conventional network infrastructures, autonomous vehicle environments and IoT environments.**

**From this survey, readers will learn that: 1) the earliest methods, signature-based IDSs, continue to make significant contributions to modern systems, despite their well-known weaknesses; 2) NN-based detection, although considered promising and under development for more than two decades, and despite numerous related approaches, still faces significant challenges in practical deployment; 3) LLMs are useful for NIDS in many cases, and a number of related approaches have been proposed; however, they still face significant challenges in practical applications. Moreover, they can even be exploited as offensive tools, such as for generating malware, crafting phishing messages, or launching cyberattacks. Recently, several studies have been proposed to address these challenges, which are also reviewed in this survey; and 4) strategies for constructing domain-specific LLMs have been proposed and are outlined in this survey, as it is nearly impossible to train a NIDS-specific LLM from scratch.**

## I. Structure of This Survey Paper

This survey focuses on existing studies of Network Intrusion Detection Systems (NIDS), tracing its evolution from the earliest signature-based methods to neural network (NN)-based approaches, and finally to the latest developments involving cooperation with large language models (LLMs). The strengths, limitations, and current status of each type of method are clearly and concisely summarized, with references to representative studies. This survey also reviews recent studies that aim to mitigate the challenges of LLMs when utilized in NIDS. Additionally, it examines cases where LLMs have been exploited as offensive tools. Since the scope of this survey is broad, a structural outline of the contents is provided in Figure 1 to help readers grasp the overall organization.

Fig. 1. Structure of This Survey Paper.

## II. Signature-Based Network Intrusion Detection: Current Trends and Limitations

Signature-based NIDSs remain one of the most fundamental and widely deployed approaches in network security. They rely on predefined rules or signatures that describe known attack patterns, allowing high-accuracy detection of previously observed threats. One of the most representative tools in this category is Snort [1] (1999), developed by Martin Roesch in 1998 and currently being maintained by Cisco. Its role has evolved from being a standalone IDS tool in the early 2000s to becoming a component within larger, integrated security solutions. As of 2025, it continues to be actively developed and maintained by Cisco Systems. The latest version, Snort 3 released around 2021, is now a standard feature of Cisco Secure Firewall Threat Defense (FTD) and serves as the engine for commercial firewall [2]. In other words, Snort remains not only a standalone tool but also a core technology within Cisco's commercial security offerings.

Two other well-known systems are Suricata [3] and Zeek (formerly Bro) [4]. Suricata was developed by OISF (the Open Information Security Foundation) in 2009 and its latest version 8.0.1 was released on September 16, 2025. It offers advantages such as multithreaded processing, automatic protocol detection, trying to enable higher throughput and scalability for modern high-speed networks. In contrast to the strictly signature-based operation of Snort and Suricata, Zeek was originally developed by Vern Paxson [5] (1999) and its latest version is 8.0.3 [4], released on October 15, 2025. It adopts an event-driven and behavior-based approach, providing richer contextual information beyond simple signature matching.

In addition, integrated frameworks such as Security Onion [6] combine multiple open-source components including Snort, Suricata, Zeek, and OSSEC (Open Source Security Event Correlator) to provide both network-based and host-based intrusion detection capabilities within a unified monitoring and incident response platform, in which OSSEC functions as a host-based IDS (HIDS), focusing on log analysis, file integrity monitoring, and rootkit detection.

The primary strength of the signature-based IDS is their high accuracy and low false-positive rate when dealing with known threats. Although it has had a long history and still used in modern systems as mentioned above, many studies have mentioned its limitations [7] (2025), [8] (2022), [9] (2025), [10] (2025), [11] (2015) and [12] (2025). The main limitations are summarized as follows.

*1)* ***Inability to detect unknown (zero-day) attacks*** *:* This is because it tries to find intrusions by simply matching.

*2)* ***Vulnerability to evasion and obfuscation techniques*** *:* Attackers can evade signature matching by encrypting or encoding payloads;

*3)* ***High maintenance and operational costs (delayed signature updates)****:* To remain effective, signature databases must be continuously updated and tested using threat intelligence. Delays or errors in updates can lead to security gaps or false detections [12] (2025);

*4)* ***High risk of false positives and false negatives****:* Signature-based systems may generate false alerts by matching benign traffic with similar patterns (false positives), while failing to detect modified or variant attacks (false negatives);

*5)* ***Scalability and performance limitations****:* Matching a large number of signatures in real time across large-scale networks imposes a heavy computational burden. In high-throughput environments, this can lead to delays or missed detections;

*6)* ***Difficulty in capturing attack context****:* Attack chains or multi-phase attacks often form attack context. Because signature matching is based on isolated patterns, it is difficult to correlate and detect the full life-cycle of an attack, such as $reconnaissance \rightarrow intrusion \rightarrow lateralmovement$. Each stage may look benign in isolation [11] (2015) and [12] (2025).

To overcome these limitations, hybrid and anomaly-based IDSs incorporating machine learning and deep learning models have emerged since the mid-2010s, combining the interpretability of rule-based systems with the adaptability of data-driven detection. Nonetheless, signature-based approaches remain a critical foundation for network defense infrastructures due to their transparency, stability, and operational maturity.

## III. Neural Network-based Network Intrusion Detection: Capabilities and Limitations

Due to the above-mentioned limitations of signature-based IDSs, NN-based detection has attracted significant research interest, with numerous models proposed over the past two decades. There also have been many survey papers on NN-based IDS technologies. Here, we provide a brief overview organized by categories such as single-model approaches, ensemble methods, temporal sequential models, and explainable models.

### A. Single NN Model-Based Approaches

In the early days of using neural network (NN)-based models, researchers attempted to implement intrusion detection systems (IDS) using single NN models, starting with a single MLP (Multi-Layer Perceptron) model. [13] (2001) trained a backpropagation neural network (of the MLP type) to detect novel attacks (anomaly detection)—they showed how to train neural networks to generalize to previously unseen attack patterns and discussed the issues of feature design and generalization in intrusion detection systems. [14] (2002) empirically compared NN models (MLP/backpropagation) and SVMs (Support Vector Machines) to achieve high accuracy on the KDD/DARPA benchmark; they also investigated reduced feature models (using feature subsets) and reported practical performance trade-offs. This is an early and widely cited IEEE conference paper demonstrating the applicability of MLPs to intrusion detection system (IDS) benchmarks. [15] (2004) implemented a multilayer perceptron (MLP) for multiclass classification of attack types (not just binary abnormal/normal) and compared several hidden layer configurations and regularization settings. The paper represents an early 2000s effort to use a single MLP for multiclass intrusion detection (IDS) tasks. Several other models have also been used. [16] (2014) introduced the deep belief network (DBN) to the field of intrusion detection, and proposed an intrusion identification model based on it. [17] (2015) proposed a stacked autoencoder (SAE) deep network for network traffic identification. Experimental results using a real-world dataset collected from the authors' corporate network show that the proposed scheme performs well. [18] (2015) used a deep belief network (DBN) trained with a restricted Boltzmann machine (RBM) to classify traffic data. The study claims that their method performs significantly better than the DBN-SVM method [19] (2011), where DBN is used for feature extraction and SVM is used for classification.

Many DNN (deep neural network) models have also been explored. The method in [20] (2018) is based on a DNN and uses four hidden layers for classification. The output layer consists of a fully connected network and a softmax classifier. Rectified linear units are used as the activation function for the hidden layers of this model. [21] (2018) proposed an asymmetric DAE (deep autoencoder) and a stacked NDAE-based DNN classification model for unsupervised feature learning (FL). Only the encoder portion of the autoencoder (AE) is used, operating in an asymmetric manner, aiming to improve the model's efficiency in terms of computation and processing time. Two asymmetric DAEs, each with three hidden layers, are stacked. RF (random forest) is used for classification. Experimental results show that this method achieves better detection performance than [16] (2014) and [22] (2016). [23] (2019) also proposed a model based on a deep neural network (DNN), specifically consisting of multiple stacked fully connected layers. The goal of the study was to implement a flow-based multi-class classification anomaly intrusion detection system (IDS). Oversampling and downsampling techniques were used to improve detection performance for minority classes.

Many approaches using sequence models (LSTM, RNN, Transformer) have also been proposed. This is in the context of many attacks unfolding over time (e.g., $scanning \rightarrow exploitation \rightarrow lateral movement$), and sequence models capture temporal dependencies that static approaches miss. [22] (2016) used an LSTM architecture to construct an IDS model with deep learning methods. [24] (2016) was one of the earliest LSTM-based IDS studies, using KDD99 data to show that packet/flow sequences have temporal dependencies that can be exploited by RNNs (recurrent neural networks). [25] (2017) utilized system logs as natural language sequences and used LSTM to detect sequence anomalies. This was a highly influential work that demonstrated the effectiveness of sequence modeling for multi-stage and log-based attack detection. [26] (2017) applied RNNs (LSTMs) to packet/flow sequences, demonstrating higher detection accuracy than traditional methods. [26] (2017) explored how to model IDS based on DNNs and proposed a DNN approach for intrusion detection using RNNs. Experimental results showed that this approach outperformed traditional machine learning classification techniques such as J48, Naive Bayes, MLP, SVM, and Random Forest (RF) in both binary and multi-classification tasks. Furthermore, the study investigated the impact of the number of neurons and different learning rates on model performance. [27] (2018) proposed an IDS architecture consisting of a recurrent neural network (RNN) with a gated recurrent unit (GRU), a multilayer perceptron (MLP), and a softmax module. Experiments showed that the GRU was more suitable as the memory unit of the IDS than the LSTM, demonstrating its effectiveness as a simplification and improvement. Furthermore, the bidirectional GRU performed better.

### *B. Multiple NN Models-Based Approaches*

Currently, many detection approaches using multiple models (including sequential and ensemble models) have been proposed. [28] (2019) proposed a deep hierarchical network based on CNN and LSTM, where CNN is used to extract spatial features of the flow and LSTM is used to extract temporal features. Finally, the features are fed into a fully connected network to classify the flow. The two networks are trained simultaneously to automatically extract spatial and temporal features of the flow. Experiments show that this method performs better than methods using CNN or LSTM alone. [29] (2020) proposed an intrusion detection system that combines convolutional neural networks (CNN) and bidirectional long short-term memory (BiLSTM) models in a deep hierarchical architecture. In this system, CNN is used to extract spatial features, while BiLSTM is used to capture temporal features. The authors claim that the multi-class attack detection performance of this scheme is significantly better than that of using only a single model. [30] (2021), [31] (2022) combine LSTM with CNN or feature preprocessing (e.g., PCA) for multi-class detection. [32] (2023) introduces a Transformer NN-based intrusion detection system (TNN-IDS) designed for IoT networks supporting MQTT. By leveraging the parallel processing capabilities of the Transformer architecture, the model accelerates learning and improves detection of malicious attacks. [33] (2023) provides a clear comparison between the Transformer-based intrusion detection system and the RNN/LSTM-based intrusion detection system, demonstrating better learning and training efficiency for long sequences. [34] (2024) proposes a method for learning network feature representations and detecting feature interactions in imbalanced data using Transformer-based transfer learning.

Ensemble learning has emerged as a powerful intrusion detection strategy, motivated by the need to capture diverse attack patterns that a single classifier cannot typically effectively model. By combining multiple base learners (whether through bagging, boosting, stacking, or hybrid voting methods), researchers have demonstrated that it can improve the robustness, adaptability, and generalization of IDS performance on benchmark datasets. One prominent direction is the use of stacked ensembles with feature selection. [35] (2016) implemented a DNN-based approach using restricted Boltzmann machines (RBMs) and deep belief networks (DBNs). A single-hidden-layer RBM is used for unsupervised feature reduction, and the resulting weights are passed to another RBM to construct the DBN. The pre-trained weights are then fed into a fine-tuning layer consisting of a logistic regression (LR) classifier and a softmax function for multi-class classification. [36] (2019) proposed an adaptive ensemble model that uses multiple base classifiers, such as KNN, DT, RF, and DNN, and selects the best classifier using an adaptive voting algorithm. Experiments show that this approach outperforms each method individually. [37] (2019) proposed a two-stage deep neural network (DNN) model based on stacked autoencoders (AEs) and a softmax classifier. The model consists of two decision stages: the first stage classifies network traffic as normal or abnormal, and the results are then used as additional features in the second stage to classify normal traffic and various attack types. This approach enables the model to learn meaningful feature representations from large amounts of unlabeled data and perform automatic classification.

[38] (2020) also proposed a multi-layer approach to develop flexible and effective intrusion detection models. This architecture combines an unsupervised stage for multi-channel feature learning with a supervised stage that exploits cross-channel feature dependencies. In the unsupervised stage, two autoencoders are trained on normal and attack streams to reconstruct samples. These reconstructed samples are then used to create an augmented dataset, which is input into a one-dimensional convolutional neural network (1D-CNN). The output is flattened and passed through a fully connected layer before entering a softmax layer for final classification. [39] (2019), [40] (2020), [41] (2020), and [42] (2021) implemented sequential detection methods. That is, by using multiple detection models connected sequentially to obtain the final detection result, their detection performance is better than that of a single model. [43] (2022) proposed a hybrid intrusion detection system that combines correlation-based feature selection with a weighted stacked classifier, achieving enhanced multi-class detection performance on the NSL-KDD and CIC-IDS2018 datasets. [44] (2021), [43] (2022), [45] (2023), [46] (2024). These works emphasize the importance of combining feature optimization and ensemble modeling for effective detection. [47] (2024) proposed a trigger-based two-stage detection method. For the first time, a Riemannian manifold metric was used as a trigger feature, and a mechanism was proposed to update the trigger threshold based on feedback from the second-stage detection results. Experimental results show that this scheme requires significantly fewer second-stage detection calls than previous trigger-based two-stage

detection systems. [48] (2023).

### C. Explainable IDS Approaches

Research on explainable intrusion detection systems (IDS) has also attracted considerable attention in recent years. Several studies have aimed to introduce AI-based intrusion detection systems with some degree of explainability. [49] (2022) proposed a compact two-stage pipeline that uses high-performance ensembles (e.g., XGBoost) for detection and incorporates explainability to validate and reinforce decisions. [50] (2023) proposed an explainable AI-based intrusion detection system for IoT scenarios that combines machine learning detection with SHAP-based explanations to reveal which flow-level and device-level features drive a given alert. [51] (2023) introduced an explainable intrusion detection system (X-IDS) that enables the system to explain its decisions. [52] (2024) proposed E-XAI, an end-to-end evaluation framework that systematically measures the effectiveness of black-box XAI techniques (particularly SHAP and LIME) in explaining the decisions of ML/DL models used in network intrusion detection. A review of related research was presented in [53] (2022), which was one of the first comprehensive reviews to define explainable intrusion detection systems (X-IDS) as an independent research field, listing black-box and white-box explainability methods, stakeholder requirements (analysts vs. managers), and evaluation challenges.

### D. Transformers-Based Approaches

[54] (2022) proposed RTIDS, a Transformer encoder-based intrusion detection system (IDS) that uses positional embeddings and a stacked encoder-decoder structure to reconstruct compact feature representations to handle imbalanced high-dimensional streaming data. They demonstrate strong detection performance on common benchmarks. [55] (2024) applied Transformer-based spatiotemporal mechanisms to CAN (Vehicle Controller Network) messages, focusing on detecting anomalies in CAN protocol traffic. [56] (2024) proposed a Transformer model-based approach specifically for cloud environments. This approach combines the core principles of NIDS with the inherent attention mechanism of the Transformer architecture, enabling deeper analysis of the relationship between input features and various intrusion types, thereby improving detection accuracy. [57] (2025) proposed a self-supervised approach using contrastive learning between Transformer encoders and raw packet sequences. It aims to better generalize to unseen traffic/zero-day anomalies.

### E. Limitations of NN-Based Detection

As mentioned above, signature-based intrusion detection systems (IDS) have many limitations, while neural network (NN)-based detection methods have received widespread attention. Since over 20 years ago, a large number of NN-based detection models have been proposed. As mentioned above, many traditional neural network models have been tried: basic models such as MLP and Deep MLP; sequence and time series models such as RNN, LSTM, and GRU; spatial feature extraction models such as CNN; anomaly detection models such as AE and VAE (Variational Autoencoder); deep generative and feature learning models such as DBN (Deep Belief Network) and RDM (Restricted Boltzmann Machine); regularization and stability-oriented models such as self-normalizing neural network (SNN); and hybrid models such as CNN + LSTM or AE + DNN, which can simultaneously capture spatial and temporal features. However, even in today's cybersecurity environment, signature-based intrusion detection systems (IDS) still play a vital role. Indeed, signature-based intrusion detection systems (IDS) are now commonly combined with anomaly detection and machine learning methods to create hybrid solutions that can adapt to evolving threats ([58] (2010), [59] (2016), [60] (2021), and [7] (2025). This is primarily due to various weaknesses and challenges inherent in neural network-based IDS, such as the stability of detection performance and lack of interpretability. This subsection will address the key specific challenges facing neural network-based IDS.

*1) **The lack of Labeled Data in the Training Dataset**:* In practice, providing labeled data is extremely difficult. To label traffic data in training datasets as "attack-like" or "benign," security experts must examine logs, payloads, and system behavior. Even experts struggle to determine whether certain anomalous traffic is an attack or simply anomalous but benign. This process is labor-intensive, time-consuming, and requires specialized knowledge. This means manual labeling is costly. Furthermore, new attack types emerge frequently (zero-day vulnerabilities, new malware, adversarial attacks). Since there are no labeled samples when they first appear, datasets quickly become outdated. Privacy and confidentiality are also significant factors. Real-world network traces often contain sensitive or personal data (emails, financial transactions), and organizations are reluctant to share raw data, limiting the availability of publicly labeled datasets. All of this makes building large, clean, labeled datasets often impractical.

However, the classifier must learn the statistical distribution of benign and attack traffic. If there are insufficient labeled samples, the learned decision boundary will be incomplete or biased, that is, the classifier will not be able to capture the diverse patterns of normal and malicious traffic, resulting in misclassification. More labeled data ensures that the classifier captures real patterns rather than random artifacts [61] (2006), [62] (2016), [63] (2019). In addition, insufficient labeled data in the training dataset can lead to overfitting (high accuracy on the training data but poor performance on unseen traffic), especially when the detection model becomes very large and complex, with a large number of parameters [61] (2006), [62] (2016), [63] (2019).

*2)* ***Data Imbalance in Training Datasets:*** The datasets used in training are often highly imbalanced between different attack classes and between attack samples and benign samples. Due to the limited number of labeled samples for the minority class, the classifier is biased towards the majority traffic. This reduces the recall rate/true positive rate of the attack. Data imbalance is a common problem, particularly evident in IDS [64] (2020), [65] (2023), and [66] (2023). The imbalance in sample distribution can cause the model to favor samples from the majority class during training and ignore samples from the minority class [67] (2012) and [63] (2019). Data imbalance can lead to model bias. Specifically, it makes the model easily biased towards the majority class data during training, making it ineffective in detecting minority class attacks [63] (2019).

Although some oversampling methods (such as SMOTE) have been used to solve this problem, such methods obviously introduce noise (unrealistic data) and have at least the following side effects. Although some alternatives and improvements to SMOTE have emerged, the core challenges still exist.

- **Overfitting problem**. This is because oversampling simply appends duplicate data to the original dataset, which can cause multiple instances of certain examples to become ”tied,” leading to overfitting [68] (2009).
- **Class overlap problem**. This is because if the minority and majority classes are close in feature space, oversampling may generate synthetic samples that fall into the majority region [68] (2009).
- **Damage to the data distribution of the minority class**, especially when there are different attack subclasses within the minority class. In this case, SMOTE may incorrectly interpolate across subclasses, generating unrealistic attack samples. This can also negatively impact detection performance [69] (2004).
- **Poor performance on high-dimensional data**. Features used in network intrusion detection are often high-dimensional, meaning the data in the feature space is very sparse. The distance-based interpolation used in SMOTE can create unrealistic synthetic points in the sparse space, which significantly degrades detection quality [70] (2019).

On the other hand, undersampling algorithms can reduce the number of majority class samples, thereby improving the problem of data imbalance, but there are also some negative effects.

- **Loss of potentially useful information**. Randomly removing benign samples may discard informative patterns. This can cause the classifier to miss important variations in normal traffic, leading to more false positives [71] (1997).
- **Risk of underfitting**. If a large number of critical samples are removed, the classifier may fail to correctly model the majority class [72] (2003).
- **Risk of misrepresenting true traffic**. In fact, artificially balancing the dataset by removing samples may reduce its realism, thereby harming practical deployment [68] (2009).

*3)* ***Complexity of Feature Decision:*** The core of network attack detection lies in isolating attacks within the feature space. Therefore, the separability of different data categories in the feature space is crucial, and feature selection has a decisive impact on detection performance. Feature selection in high-dimensional data remains a significant challenge. Ignoring key features significantly impacts detection performance. However, irrelevant or redundant features can also severely degrade model performance. Consequently, considerable work has been devoted to efficient feature selection, and numerous different feature selection methods (filtering, wrapping, and embedding) have been proposed. However, each of these methods has its own advantages and disadvantages, and no single approach guarantees optimal results in all scenarios. Even for experts, choosing the right method remains a challenge. [73] (2024).

The Internet environment has become increasingly complex, especially with large-scale distributed networks and the Internet of Things (IoT). Many complex attacks are often highly dynamic and exhibit unique spatiotemporal characteristics. [58] (2010), [74] (2016), [63] (2019), and [75] (2020). That is, they evolve rapidly (dynamic), often propagate across hosts or subnets (spatial), and often unfold over time (multi-step, multi-stage, slow-to-slow behavior). This makes it difficult for static machine learning models to adapt and capture the spatiotemporal characteristics of real-world attacks, thus limiting the effectiveness of machine learning.

The complexity and dynamic nature of the spatiotemporal characteristics of network attack traffic make it difficult to build models that effectively detect attacks. Specifically, the complexity of the data includes the following aspects [76] (2025):

- **High dimensionality of network traffic**. It typically contains network layer information, source and destination information, and time series data. Furthermore, it requires interactive consideration, making it more difficult to extract spatiotemporal patterns;
- **Heterogeneous network environment**. Especially in the IoT environment, different subnetworks, nodes, and devices make it difficult to determine a set of features that are suitable for attack detection in different network environments [77] (2021);
- **Temporal dependency**: Many attack behaviors are temporally dependent, meaning that malicious activities often unfold as a sequence of related events rather than isolated anomalies [63] (2019). For example, reconnaissance, exploitation, and data exfiltration often proceed in stages over time, and their detection requires capturing these sequential patterns. Previous research has shown that recurrent models such as LSTM are effective for intrusion detection precisely because they can learn temporal correlations in attack traffic [78] (2017) and [79] (2016). Similarly, system log analysis frameworks like DeepLog have also highlighted that anomaly detection performance improves significantly when temporal structure is explicitly modeled [80](2017). These findings suggest that accounting for temporal dependencies is crucial for accurately detecting and classifying evolving cyberattacks. Consequently, effectively detecting such multi-stage, complex attacks is practically difficult and, in many practical cases, nearly impossible.

*4) **Robustness of Detection Models:*** Robustness refers to a model's ability to maintain good performance as conditions change from training to deployment. This includes both unintentional changes, such as distribution shifts (different network environments, new traffic patterns, new attack variants), and deliberate attempts to evade the model (adversarial attacks)[58] (2010) and [81] (2023). That is, the performance of a detection model can be unstable (unrobust) in two main cases: 1) An NN-based NIDS that appears to perform very well in lab evaluations may miss real attacks or trigger many false positives in production[81] (2023). 2) Deliberate adversarial attacks aim to intentionally cause misclassifications in the detection model. An attacker can craft minimal feature changes that push the input beyond the learned decision boundary without changing the observable malicious intent (e.g., packet timing adjustments, small payload changes, or flow-level feature perturbations). Machine learning models, especially deep networks, learn complex boundaries that adversaries can exploit[82] (2023). The robustness issue has led to the aforementioned current situation: feature-based detection remains the mainstream in network intrusion detection, while the practical application of neural network-based models is very limited, despite the former's many weaknesses mentioned above.

In NIDS research, robustness encompasses the ability to generalize under data or environmental changes and the ability to resist adversarial attacks. Addressing both aspects is crucial for the effectiveness of any machine learning-based intrusion detector. Although many studies have attempted to address the adversarial attack problem ([83] (2024)), no fundamental solution has yet been found. Furthermore, the first type of unintentional changes, such as distribution shift, stems from problems in the training data, such as data imbalance and a shortage of labeled samples. Therefore, this remains a challenging problem.

*5) **Non-interpretability of Detection Models:*** NN models, especially DNN models, often operate like ”black boxes,” making their decision-making process difficult to understand. Explaining why certain traffic is classified as an attack is crucial for enabling human intervention after an alert is triggered ([64] (2020)). While some research has attempted to mitigate this issue, as discussed in the previous section ([50] (2023), [51] (2023), and [52] (2024), XAI tools have limited adoption among incident responders and struggle to meet the decision-making needs of analysts and model maintainers ([84] (2023).

While NN-based IDSs offer adaptability compared to signature-based systems, they still suffer from data scarcity, robustness, and interpretability issues. The next section introduces LLMs as a promising complement, explaining how their contextual reasoning and generative capabilities can address these gaps and enable more adaptive intrusion detection.

## IV. Utilizing LLMs for Network Intrusion Detection: Capabilities and Limitations

Given the persistent limitations of NN-based approaches, researchers have explored LLMs to overcome feature engineering bottlenecks and improve explainability. We now examine the advantages of LLMs for NIDS and practical strategies for domain adaptation.

### *A. Benefits of LLMs for NIDS*

Pretrained language models (PLMs) have demonstrated impressive performance across a variety of natural language processing (NLP) tasks [85] (2023), [86] (2023), [87] (2023), and [88] (2024). Studies have shown that performance tends to improve with increasing model size, especially above certain parameter thresholds [89] (2022) and [87] (2023). Models with large parameter sizes are often referred to as LLMs [88] (2024). LLMs are complex models trained on massive datasets consisting of publicly available text and corpora. They typically contain billions to trillions of parameters. For example, Mistral 7B has 7 billion parameters, considered relatively small, while LLaMA-4 has 2 trillion parameters, making it one of the largest models. Prominent examples include GPT, Claude, Falcon, LLaMA, BloombergGPT, and TigerBot. These models have demonstrated outstanding performance in tasks such as chatbot interaction, text generation, and programming. LLMs can be general or domain-specific. For example, Llama-Fin is tailored for financial applications, supporting tasks such as risk assessment and compliance evaluation through post-training on financial corpora, order tracking data, and preference extraction. Med-PaLM is designed for medical and clinical tasks and fine-tuned using medical literature, case studies, and QA datasets.

The core mechanism behind LLMs is tokenization, which breaks text into smaller units called tokens. LLMs predict the next token based on context, generating coherent and task-specific language. This makes their output both interpretable and effective. LLMs have had a profound societal impact, with applications in areas such as image generation, text composition, and music composition. Their influence spans nearly every field, including cybersecurity.

Traditional machine learning models, such as rule-based systems or neural network-based classifiers, often rely on hand-crafted features and struggle to capture long-range dependencies. [90] (2021) LLMs offer a promising solution to these limitations. According to [91] (2023), [92] (2023), [84] (2023), [93] (2023), [94] (2023), [91] (2023), and [95] (2023), through fine-tuning, LLMs can: 1) identify patterns in massive datasets, 2) learn the characteristics of malicious traffic, 3) detect anomalies, 4) characterize the intent behind intrusions, and 5) provide actionable recommendations for security responses.

As the number and complexity of cyber threats continue to grow, the demand for intelligent systems that can automatically detect vulnerabilities, analyze malware, and respond to attacks is becoming increasingly urgent. In recent years, the application of LLM in intrusion detection systems (IDS) has attracted much attention, opening up new avenues for AI-driven network security.

Many studies, including [96] (2024), have evaluated the potential of LLM in network intrusion detection systems (NIDS). These studies focus on how LLM can process and understand massive amounts of network log data, autonomously learn, adapt to changing network behavior, and effectively distinguish between normal activities and potential threats. The results show that integrating LLM into intrusion detection systems (IDS) has many practical advantages and can significantly enhance their capabilities:

*1) **Continuous Adaptation**:* LLM is highly adaptive, able to learn and update its knowledge as new threats emerge. This ensures that the IDS can effectively detect new and sophisticated attacks that may bypass traditional rule-based detection mechanisms.

*2) **Automated Policy Implementation**:* LLM enables streamlined automation, enabling complex security policies to be implemented with minimal human intervention. This reduces the likelihood of manual configuration errors and helps prevent misconfigurations that can lead to security vulnerabilities.

*3) **Deep Behavioral Insight**:* LLM provides a comprehensive understanding of network traffic behavior, enabling intrusion detection systems (IDS) to identify subtle anomalies that may be difficult to detect. This capability helps identify behaviors that may indicate potential threats or misconfigurations in network devices, allowing proactive preventive measures before an attack occurs.

### *B. Construction of Domain-Specific LLMs for NIDSs*

Generally speaking, developing specialized LLMs from scratch for network security applications, such as NIDS, is often impractical due to the extensive computational resources required. Fortunately, existing general-purpose LLMs have accumulated rich linguistic and semantic knowledge and exhibit strong generalization capabilities. Instead of building new models from scratch, we can enhance their effectiveness in the network security domain by fine-tuning them using domain-specific datasets. This approach allows us to leverage the rich knowledge embedded in pre-trained LLMs while adapting them to the unique characteristics of network traffic and intrusion patterns.

By combining pre-trained LLMs with target network security data, we can achieve efficient and scalable NIDS implementations that fully utilize existing resources while improving detection accuracy in complex and evolving threat environments. To apply general-purpose LLMs to NIDS, two main approaches are commonly used: continuous pre-training (CPT) and supervised fine-tuning (SFT) [97] (2025).

*1) **Continual Pre-Training (CPT)**:* Continuous pre-training involves further training a pre-trained LLM using a large amount of unlabeled domain-specific data, such as [98] (2018), [99] (2020), [100] (2022), [101] (2024), [102] (2024), [103] (2024), [104] (2024), [105] (2024), and [106] (2025). It extends the general pre-training phase of the LLM on a new domain-specific corpus, adapting its language representation to specialized knowledge such as network security logs, network traffic data, or vulnerability reports. Instead of training a complete model from scratch, CPT reuses the general language and reasoning capabilities of a base model (e.g., GPT, BERT, or LLaMA) and applies it to large amounts of network security text or structured event data using the same self-supervised learning objective [99] (2020). This process enables the model to internalize cybersecurity vocabulary (e.g., CVE identifiers, protocol names, log patterns) and improve contextual reasoning capabilities for security-related tasks without losing general language capabilities. CPT can be performed in a variety of ways:

- **Domain-adaptive pre-training (DAPT)** retrains the model on unlabeled text from a single specialized field (e.g., threat reports or network logs) to shift its distribution toward that domain.
- **Task-adaptive pre-training (TAPT)** continues pre-training on unlabeled data drawn from the specific downstream task (for example, IDS alert logs) to reduce the gap between pre-training and fine-tuning distributions.
- More recent work also proposes **data-efficient or parameter-efficient CPT**, where only adapter layers or LoRA modules are updated to inject cybersecurity-specific knowledge while keeping the backbone frozen. In intrusion detection, such CPT has been shown to improve anomaly interpretation and threat context extraction from raw IoT or system log data.

*2) **Supervised Fine-Tuning (SFT)**:* Supervised fine-tuning leverages labeled domain-specific data to train a model, directly optimizing its performance on a specific cybersecurity task ([107] (2022),[100] (2022),[108] (2023), and[109] (2023). It aligns a pre-trained or CPT-adapted model with labeled data and a specific objective related to cybersecurity detection, classification, or inference. Compared to CPT's self-supervised adaptation, SFT uses explicit input label pairs, such as "network flow → benign/DDoS/port scan" or "log snippet → ransomware infection." The fine-tuning objective is typically to minimize the cross-entropy loss of the correct class label or command response, making the model specialized for tasks such as intrusion detection, threat classification, and incident summarization. Because full-parameter fine-tuning updates all parameters of the model, it is computationally expensive, especially for large models. Consequently, parameter-efficient fine-tuning (PEFT) techniques, such as low-rank adaptation (LoRA), have attracted considerable attention. PEFT methods fine-tune only a small number of parameters or introduce additional trainable parameters, while keeping the majority of the pre-trained LLM parameters unchanged. This approach significantly reduces computational cost while maintaining performance. Several PEFT techniques have been proposed, including adapter fine-tuning, prefix fine-tuning, hint fine-tuning, LoRA, and QLoRA. These methods offer flexible and efficient alternatives for adapting large models to specific tasks without requiring complete retraining.

**Adapter tuning** Small neural modules (called adapters) are inserted after the multi-head attention and feed-forward layers of the Transformer architecture. During fine-tuning, only the parameters within these adapters are updated, while the rest of the pre-trained model remains unchanged. This approach significantly reduces computational cost while enabling efficient task-specific adaptation [110] (2021).

**P-tuning** By introducing trainable cues, optimal cue embeddings for specific tasks are automatically learned. This eliminates the need for manually designed cues and improves performance. This approach can be further enhanced by introducing anchor tags, which help stabilize and guide the learning process, allowing it to be more effectively adapted to specific tasks. [111] (2021)

**Prefix tuning** The model parameters are kept fixed, and a small set of continuous, task-specific vectors (called prefixes) are optimized. These prefixes are added to the input sequence and guide the model's behavior during inference, allowing efficient adaptation to specific tasks without modifying the core model. [111] (2021)

**Prompt tuning** Fine-tuning a language model for a specific task by learning soft hints via backpropagation and using labeled examples to guide the process [112] (2021). LoRA [100] (2022) introduces a small, trainable sub-module in the Transformer architecture. It freezes the pre-trained model weights and inserts a trainable low-rank factorization matrix at each layer, significantly reducing the number of trainable parameters required for downstream tasks. After training, the learned matrix parameters are merged with the original model. QLoRA [113] (2023) builds on LoRA by introducing further optimizations, such as quantization techniques, to reduce memory usage and improve fine-tuning efficiency.

In the field of intrusion detection, SFT is often combined with labeled datasets such as CICIoT2023 or TON-IoT, enabling models to classify new or unseen network events with high accuracy [114] (2024) and [115] (2025).

In essence, CPT focuses on transferring domain knowledge, while SFT aligns the model with a clear detection goal. CPT enriches the model's internal representation space with cybersecurity semantics, while SFT operationalizes this knowledge to achieve actionable intrusion detection system (IDS) performance. Modern LLM-based intrusion detection system (IDS) frameworks often combine the two—first performing domain-adaptive CPT on large amounts of unlabeled security data, followed by SFT on smaller labeled attack datasets—to achieve superior detection of both known and novel threats [116] (2024) and [117] (2024).

*3) **Prompt engineering:*** :

Recent research in natural language processing has highlighted the importance of hint engineering as an emerging fine-tuning approach [118] (2021) and [119] (2023). By designing effective hints to guide LLMs toward desired outputs, it can alleviate the training data and resource bottlenecks required for building cybersecurity models. In hint engineering, inserting task-specific hints is particularly beneficial for security tasks involving limited data features. This approach enables LLMs to learn directly from stream-level features in a zero-shot learning manner [120] (2020). In this way, LLMs can extract structured cyber threat intelligence from unstructured data, providing standardized threat descriptions and formalized classifications [121] (2024).

Other emerging techniques also provide valuable insights for building cybersecurity-focused LLMs. Model editing techniques [122] (2023) and [123] (2024) enable direct modification of LLMs to incorporate cybersecurity knowledge without negatively impacting the model's performance in unrelated domains. These methods allow for targeted updates to the model's internal representations, which makes them particularly suitable for adapting LLMs to the evolving threat landscape while retaining general language understanding capabilities.

### *C. Existing Approaches Applying LLMs in NIDSs*

This topic has attracted widespread attention in the field of network intrusion detection, and a large amount of related research has been carried out. The following sections will introduce some representative research results and classify them according to different research fields.

*1) **Approaches Employing BERT-Based Encoders and Classification Heads:*** BERT (Bidirectional Encoder Representations from Transformers) [124] (2019) has demonstrated impressive performance in enhancing various natural language processing (NLP) tasks. In the field of network intrusion detection systems (NIDS), BERT is primarily used as an encoder backbone—a feature extractor that converts sequences of network events (e.g., packets, flows, CAN messages) into contextual embeddings. These embeddings are then passed to a classification head, such as a multilayer perceptron (MLP), a softmax layer, or an anomaly scoring module, for attack detection. Several studies have explored BERT-based NIDS approaches:

[125] (2022) applied BERT to detect attacks against CAN (Controller Area Network). By treating arbitration IDs as tokens and using a masked language model objective, BERT can learn periodic protocol sequences and detect multiple attack types in resource-constrained automotive environments. In [126] (2023), network flow sequences are treated as sentences, and a BERT-style encoder is used to model contextual relationships across flows. Compared to traditional machine learning methods, fine-tuning improves intrusion classification and domain adaptation. Furthermore, hybrid models have been proposed for IoT security. In [126] (2023), a framework integrates GPT (for packet prediction), BERT (for verification prediction), and LSTM (for packet classification). This combination enhances the predictive capabilities of IoT network security.

[127] (2023) proposed a lightweight semantic fusion intrusion detection model. BERT captures semantic features, while BiLSTM learns to fuse features through knowledge distillation, enabling efficient classification. [128] (2024) proposed a BERT-based architecture for

IoT network threat detection. This architecture incorporates privacy-preserving fixed-length encoding (PPFLE) during training, and outperforms traditional machine learning/deep learning methods.

[129] (2024) proposed a flexible Transformer-based NIDS architecture. This architecture captures long-term network behavior and allows modular replacement of components such as input encoding (including BERT), Transformer layers, and classification heads. The study also analyzed how encoding and classification choices affect performance. [34] (2024) used the BERT-large model with a multi-head attention mechanism to mine network data and extract training features, thereby improving the performance of wireless devices.

[130] (2025) transforms network traffic into natural language-like sequences, enabling BERT to extract high-quality features. Its bidirectional encoder captures complex contextual information, improving detection accuracy and identifying complex attack patterns. The model efficiently processes sequential data, captures temporal dependencies, and reduces computational complexity, making it suitable for real-time or resource-constrained environments. In [131] (2025), BERT is used for anomaly detection in resource-constrained environments. Its efficient architecture enables low-cost evaluation, making it suitable for IoT applications. [132] (2024) combines federated learning with BERT to build a powerful intrusion detection system (IDS). It supports centralized and federated contexts and uses linear quantization to compute the model for edge deployment, demonstrating the feasibility of LLM in IoT ecosystems.

When LLMs like BERT are used solely as encoders in intrusion detection systems (IDS), the classification head becomes crucial. It determines the model's efficiency in converting latent embeddings into accurate predictions of attack types or anomaly scores. Typically, the classification head is a lightweight neural layer or subnetwork that takes contextual embeddings (typically [CLS] tokens) from BERT and outputs a probability distribution over attack categories. The design and complexity of this classification head significantly impact detection performance and computational efficiency. Several studies have explored different approaches for the classification head in BERT-based IDS frameworks: a) Linear Softmax Head: In [125] (2022), a simple linear softmax layer is used as the classification head. This lightweight design is optimized for embedded deployment, balancing performance and resource constraints; b) CNN-LSTM Hybrid Head: [34] (2024) proposed a combined CNN and LSTM architecture for classifying various types of network intrusion attacks. This hybrid approach is able to capture both spatial and temporal features, and its effectiveness has been verified by comparison with other deep learning methods. These diverse implementations highlight the importance of tailoring the classification head to the specific requirements of intrusion detection systems (IDS), such as accuracy, interpretability, and deployment constraints.

*2) **Approaches based on LLMs for NIDSs in Conventional Network Environments**:* Recent research has explored various approaches to enhance the performance of intrusion detection systems (IDSs) using LLMs and Transformer-based architectures. These studies focus on improving anomaly detection capabilities, interpretability, adaptability, and scalability to adapt to various network environments.

[133] (2022) proposed a scalable multi-anomaly detection model (called AnomalyAdapters) that encodes log sequences using a series of pre-trained Transformers. It uses adapter modules to efficiently learn log structures and anomaly types. This adapter-based approach preserves contextual information, reduces parameter overhead, and can learn across diverse log sources without sacrificing performance. [134] (2023) proposed a novel framework (called ChatIDS) that uses LLMs to explain IDS alerts in an intuitive language. Designed to assist non-experts, ChatIDS can interpret alerts and suggest actionable security measures. ChatGPT demonstrated its feasibility, showcasing the potential of conversational AI in cybersecurity. [135] (2024) proposes a method for detecting cyberattack behaviors by leveraging the combined strengths of large language models and a synchronized attention mechanism. The effectiveness of the proposed approach is evaluated across diverse datasets, including server logs, financial transaction behaviors, and user comment data. [136] (2024) enhances DDoS detection by converting non-contextual network flows into structured sequences that can be processed by LLMs. The proposal is named DoLLM. DoLLM uses the Llama2-7B model to label flow data and leverages semantic understanding to detect complex attacks such as carpet bombing. This approach significantly improves detection accuracy by capturing subtle flow patterns. [137] (2024) proposes an adaptive detection framework, a scalable, real-time intrusion detection system (IDS) that evolves with emerging threats. It uses a BERT encoder to distinguish between benign and malicious traffic and applies a Gaussian mixture model (GMM) to cluster high-dimensional embeddings. This enables dynamic identification of unknown attack types while

maintaining high detection accuracy. [138] (2025) proposes an LLM-based detection architecture that comprehensively explains how to use LLMs in network attack detection. The paper explores the three roles of LLMs in pre-training, fine-tuning, and detection: classifier, encoder, and predictor. A DDoS detection case study demonstrated that LLM's contextual mining capabilities excelled in identifying carpet bombing attacks. Together, these studies highlight the versatility of LLM in network security, from improving detection accuracy and explainability to providing adaptive and scalable IDS solutions for complex and evolving threats.

*3) **Approaches Based on LLMs for NIDS in Autonomous Vehicles and IoT Environments**:* Transformers and LLMs are increasingly being used in autonomous vehicle (AV) and Internet of Things (IoT) environments to develop robust and efficient intrusion detection systems (IDS). These models excel at capturing contextual patterns, detecting anomalies, and adapting to diverse network conditions.

[90] (2021) proposed a novel intrusion detection system (IDS) framework that applies LLMs to CAN network security. The authors introduced a Transformer-based approach that leverages the GPT architecture and enhances it with a bidirectional modeling mechanism. In a standard GPT model, the network predicts future tokens based solely on past context—a unidirectional, left-to-right process. However, in CAN communications, both preceding and following message patterns can provide valuable contextual information for anomaly detection. That is, using only forward predictions can miss anomalies that become apparent when checking backward consistency. To address this issue, the authors designed a bidirectional general prediction model (Bi-GPT) that employs two GPT models: one trained on the normal time series (forward GPT) and the other trained on the reversed series (backward GPT). During detection, both models calculate a prediction likelihood for each message. These forward and backward prediction losses are then combined to form a bidirectional likelihood score, which serves as an indicator of whether a given CAN message sequence deviates from normal patterns. The system is trained in an unsupervised manner using only normal CAN data, without the need for labeled attack samples. Each CAN frame (consisting of identifier and data fields) is labeled as a sequence of words similar to natural language. During inference, the bidirectional loss (the average of the forward and backward prediction losses) is used as an anomaly score. The threshold of this score determines whether a message sequence is classified as normal or an intrusion. Compared to baseline models such as LSTM, GRU, and standard unidirectional GPT, Bi-GPT achieves the highest detection performance. Multi-class classification can be achieved by training with labeled attack patterns, and the output probabilities of the Bi-GPT model can be used as features for downstream classifiers (such as softmax-based or ensemble models).

The aforementioned work [125](2022) applied BERT to detect attacks against CAN. [139](2023) proposed a Transformer-based intrusion detection system, CAN-Former IDS, which predicts abnormal behavior by simultaneously analyzing CAN ID sequences and corresponding message payload values. The model uses fully self-supervised training and token interaction, eliminating the need for manual feature design. [140](2024) designed a federated learning-edge-cloud communication architecture for the Internet of Vehicles (IoV) and introduced a feature selection Transformer, FSFormer, combined with a feature attention mechanism to dynamically identify and enhance important features, thereby improving the model's ability to extract key information. In this architecture, mobile users collect and encrypt data, then upload it to edge devices for training. [141](2025) proposed a hybrid network intrusion detection system (NIDS) called IoV-BERT-IDS, which is designed specifically for on-board and off-board networks in the Internet of Vehicles (IoV) field. The system includes a semantic extractor that converts network traffic data, including CAN packets, into a format suitable for BERT processing. The model is pre-trained and fine-tuned to effectively detect intrusions. BERT plays a key role in capturing bidirectional contextual features, significantly improving the model's generalization and detection accuracy in IoV environments.

[142] (2023) proposed a Transformer-based IoT NIDS that uses a self-attention mechanism to learn intrusion behaviors from diverse data in heterogeneous IoT environments. [143] (2023) introduced a Transformer neural network-based intrusion detection system (IDS) (referred to as TNN-IDS) for IoT networks supporting MQTT. TNN-IDS addresses the limitations of imbalanced training data by introducing parallel processing and multi-head attention (MHA) layers into the Transformer architecture, thereby enhancing the learning and detection capabilities of malicious activities. [144] (2023) proposed a multi-Transformer fusion intrusion detection system (IDS) model designed specifically for the Industrial Internet of Things (IIoT) environment. The research cited in [128] (2024) introduced a novel lightweight

architecture based on BERT for network threat detection in IoT and Industrial Internet of Things (IIoT) networks. Furthermore, [126] (2023) proposed an IoT network intrusion prediction framework that combines the advantages of GPT, BERT, and LSTM models. [145] (2024) performed anomaly-based intrusion detection using a Transformer/BERT-style encoder with BBPE labeling and evaluated the results on an IoT dataset. This is a concrete example of applying LLM technology to IoT intrusion detection. [132] (2024) proposes an intrusion detection system (IDS) using federated learning and LLM to address key constraints of the Internet of Things (IoT) (edge resources/privacy), while also leveraging Transformer/LLM techniques for network intrusion detection systems (NIDS). BERT is modified to optimize its performance on resource-constrained edge devices. Furthermore, linear quantization is used to compress the model for deployment on edge devices. In other words, the study applies a BERT-style Transformer to the edge/federated environment of IoT/5G and uses quantization/model optimization for edge deployment. [106] (2025) integrates LLM tokens/embeddings into a dual-path, payload-centric intrusion detection system (IDS) (token-aware ensemble) to detect payload-level attacks. The focus is on reducing false positives while using LLM embeddings as enriched features for the classifier. It demonstrates how LLM tokenizers/embeddings can transform raw packet payloads into semantically richer representations, improving detection of novel/obfuscated payload attacks. [117] (2024) uses LLMs as autonomous agents for threat detection and contextual interpretation in IoT networks; LLM reasoning is combined with a network feature extractor for detection and human-readable explanations. It demonstrates the role of LLMs not only in detection but also in supporting operators with interpretable outputs and suggested mitigation steps. [146] (2024) proposes a network intrusion prediction framework for IoT security that combines LLMs with LSTM networks. The framework integrates two LLMs in a feedback loop: a fine-tuned Generative Pretrained Transformer (GPT) model for predicting network traffic and a fine-tuned BERT model for evaluating the predicted traffic. An LSTM classifier then identifies malicious packets in the predicted results. [147] (2025) proposes an intrusion prediction framework for IoT security that is driven by two LLMs: a fine-tuned BART (Bidirectional Autoregressive Transformer) model for network traffic prediction and a fine-tuned BERT model for traffic evaluation. The framework leverages the bidirectional capabilities of BART to identify malicious packets among these predictions.

*4) **Approaches Based on Multiple Models**:* While Transformers and LLMs show promise for building robust intrusion detection systems (IDS) for autonomous vehicles and IoT environments, LLM models alone often fail to deliver satisfactory detection performance. This limitation stems from the fact that LLMs were originally designed for natural language tasks, not network intrusion detection. Deploying effective NIDS is inherently challenging, especially when trying to accurately identify anomalies in increasingly sophisticated and evasive cyberthreats. Most NIDS research relies on structured features such as network logs, with some exploring text-based features such as payload content. However, traditional machine learning and deep learning models often struggle to effectively learn from both tabular and textual data.

The aforementioned paper [126] (2023) is a prime example, in which GPT is used to predict upcoming network packets based on current traffic patterns, BERT evaluates the effectiveness of these predictions, and LSTM classifies packets as normal or malicious. The integration of these models significantly enhances the framework's predictive capabilities, making it highly effective in the field of IoT network security. The aforementioned papers [146] (2024) and [147] (2025) are also examples of combining multiple models. [146] (2024) combines an LLM with an LSTM network, while [147] (2025) uses two LLMs for IoT security: a fine-tuned BART model and a fine-tuned BERT model. [84] (2023) proposes a specialized network intrusion detection system (NIDS) called HuntGPT, designed to present detected threats in an easily interpretable form. This system integrates the GPT-3.5 Turbo conversational agent. The results of the study demonstrate that conversational agents based on LLM technology, combined with explainable artificial intelligence (XAI), can provide a powerful mechanism for generating explainable and actionable insights within the NIDS framework. Although XAI technologies have been introduced to help cybersecurity operations teams better assess AI-generated alerts, their adoption by incident responders has been limited. These tools often fail to meet the decision-making needs of analysts and model maintainers. Meanwhile, LLMs are recognized for their unique approach to addressing these challenges. Through fine-tuning, LLMs can identify patterns in massive datasets and adapt to diverse functional requirements. [148] (2024) discusses the strengths of machine learning models and LLMs in different tasks and explores how they can be effectively combined to address challenges associated with mobile networks. Combining LLMs with neural network-based models allows each model to leverage its unique strengths, resulting in higher performance than either model alone. To support this concept, the study analyzes the performance of LLMs compared to traditional machine learning (ML) algorithms and explores potential applications of LLMs. Furthermore, the study explores the integration of ML and LLMs, demonstrating how they can be used synergistically in a mobile network environment. The study emphasizes the integration of LLMs with traditional neural network (NN) models to enhance their functionality and adaptability. The study demonstrates the advantages of this approach through a case study, leveraging the generative capabilities of LLMs to improve the performance of NN-based intrusion detection. Specifically, this approach uses synthetic data generated by LLMs to enhance NN-based intrusion detection systems (IDS). The study proposes the concept of "generative AI-in-the-loop," leveraging the semantic understanding, contextual awareness, and reasoning capabilities of LLMs to assist humans in handling complex or unforeseen situations in mobile communication networks. [149] (2024) combines a multilayer perceptron (MLP) and a character architecture, without tokenization, in a neural encoder (CANINE) architecture for processing numerical, categorical, and textual data, respectively. It leverages the advantages of the MLP in capturing complex relationships in numerical and categorical data and the advantages of CANINE's character-based encoding for detailed text analysis. In other words, the study proposes a method for integrating tabular and textual features to enhance the performance of network intrusion detection systems (NIDS). By overcoming the limitations of traditional NIDS, this approach contributes to the development of more effective and accurate anomaly detection methods, leading to more reliable and efficient network security solutions. The results show that combining deep learning and pre-trained Transformer models, combining the two feature types, can significantly improve detection accuracy. The aforementioned [128] (2024) proposed a lightweight BERT-based architecture tailored for NIDS in IoT and Industrial IoT (IIoT) environments. Similarly, [126] (2023) introduced a NIDS framework for IoT that leverages the strengths of GPT, BERT, and LSTM models. In this framework, GPT predicts upcoming network packets based on current traffic, BERT evaluates the effectiveness of these predictions, and LSTM classifies packets as normal or malicious. The integration of these models enhances predictive power, making this framework highly effective for IoT network security. [150] (2025)presents an IDS approach that leverages a pretrained encoder–decoder LLM (T5), fine-tuned to adapt its classification scheme for attack detection. This anomaly-based method uses statistical features from historical network flows as input.

[151] (2025) introduces eX-NIDS, a framework designed to enhance the explainability of flow-based NIDSs by using LLM. A key component of the framework, Prompt Augmenter, extracts contextual information and cyber threat intelligence (CTI)-related knowledge from flows labeled as malicious by NIDS. This rich, context-specific data is incorporated into the input prompts of the LLM, enabling it to generate detailed explanations and interpretations of why a flow was classified as malicious. The approach is shown to outperform a baseline method, Basic-Prompt Explainer, which does not include contextual information in the LLM prompts. The proposed framework is evaluated using Llama-3 and GPT-4, and the results show that the augmented LLM can generate consistent and informative explanations. These findings suggest that LLM augmented with contextual data can serve as a valuable complementary tool in NIDSs to improve the explainability of malicious flow classification.

### D. Limitations of LLMs-based Detection

Although LLMs offer clear benefits, integrating them into IDS introduces critical challenges that affect accuracy, scalability, and trust. As previously mentioned, the complexity of LLMs makes their decision-making processes difficult to interpret. This lack of transparency can lead to trust issues among cybersecurity professionals, who rely on clear explanations to verify the effectiveness of security measures. Another major challenge is the computational cost required to fine-tune LLMs for specific network environments. This process requires significant resources and time, making it impractical for organizations with limited infrastructure. Furthermore, the high cost of model development and deployment can hinder the adoption of LLM-based IDS solutions, especially in resource-constrained environments. Overfitting is also a concern. When LLMs are fine-tuned on narrow datasets, they may struggle to generalize to unknown threats or network configurations. This reduces their effectiveness in identifying new or evolving attack patterns, which is crucial for maintaining a strong cybersecurity defense.

While LLMs are often valuable, they can sometimes contain incorrect or misleading information.[152] (2023) This section explains the possible causes of these issues and proposes potential solutions to address them in the next section.

*1) **Data representation mismatch and domain specificity**:* Network data (flows, packets, binary payloads, IP addresses, timing information) is very different from natural language. Converting it into tokenized or hinted embeddings so that LLMs can understand it is nontrivial. This mismatch reduces the detection accuracy. [153] (2024) showed that LLMs struggle to accurately detect malicious NetFlow, in part due to these representation issues.

*2) **Lack of network intrusion data**:* Many studies have pointed out the lack of large-scale, high-quality, and up-to-date network security datasets for training or fine-tuning LLMs in the field of intrusion detection. This limits the ability of LLMs to generalize to new threats or real network environments [153](2024), [154] (2025), and [155](2025)

*3) **High computational cost, latency, and scalability issues**:* LLMs are large in size, require a large amount of inference computation, and are generally memory/GPU-intensive. Real-time LLM inference in high-throughput networks often incurs prohibitive latency unless supported by costly hardware. [153] (2024) notes that while LLMs may help improve interpretability, their ”high computational requirements” make them less suitable for pure detection in many deployment environments. While some methods can reduce model size (distillation, quantization, pruning), they may reduce performance or robustness. [156] (2025) shows that optimization/compression increases vulnerability to adversarial attacks or novel attacks.

*4) **Adversarial robustness, prompt attacks, and data leakage**:* LLMs are vulnerable to input perturbations (noisy inputs, spelling errors), adversarial examples, and hint-based attacks (hint injection/jailbreaking), which can mislead detection models or lead to false positives/false negatives. These vulnerabilities are particularly concerning in security settings. [157] (2024) showed how carefully crafted hints can force LLMs to produce incorrect outputs even when the semantic input has not changed significantly. Furthermore, research on continuous embedding space attacks (adversarial training in embedding space) shows that LLMs can still be vulnerable even with adversarial defenses[158] (2024).

*5) **Explainability, interpretability, and trust**:* Reliable alarm reasoning/explanation is crucial for human analysts to understand the alarms, verify their correctness, and take remedial actions. While LLMs are considered to have great potential to provide some explanation in many cases, they currently struggle with accurate detection, making interpretability more of a supplement than a replacement for traditional methods [153] (2024). Without reliable explanations, there is a risk that false positives and false negatives will be easily accepted, or worse, malicious activity may go unnoticed due to misinterpretation of model confidence.

*6) **Evolving threats and model maintenance**:* Cybersecurity is a rapidly evolving field: new attack strategies, zero-day vulnerabilities, protocol changes, and more are emerging. If an LLM is trained or fine-tuned based on historical data, it may not be able to detect or adapt to new attacks. Model drift (schema changes) is a significant problem. [154] (2025) warns that the lack of up-to-date threat knowledge is critical. Furthermore, maintaining an LLM (retraining or continuous learning) is costly and prone to risks (e.g., catastrophic forgetting, introduction of bias).

*7) **Overconfidence and misleading**:* Because LLMs can be overconfident or misled (through prompt structure, biased training), false alarms can proliferate, leading to alert fatigue in real operations.

*8) **Privacy, legal, and ethical concerns**:* When using LLM on internal logs, payloads, or private network data, there is a risk of exposing sensitive data (e.g., IP, user behavior, credentials) in prompts, logs, or model outputs, which can raise privacy concerns (e.g., data leakage) as well as legal/ethical issues. Data leakage through memorization (where the model remembers parts of the training data) is not trivial ([154] (2025) and [159] (2025). Regulatory compliance (e.g., GDPR: General Data Protection Regulation) may require ensuring that data used for fine-tuning or inference is properly handled; auditing and tracing decision-making processes is even more difficult for large models.

For these reasons, LLMs hold great promise for improving detection, explainability, and automation, but they cannot yet fully replace traditional purpose-built IDS architectures, especially for real-time detection in high-throughput networks. Addressing these issues requires a combination of domain-specific engineering (feature/token design, fine-tuning, continuous learning), robust evaluation (attack awareness, adversarial testing), and operational considerations (deployment constraints, validation, privacy).

### *E. Recent Progress in Mitigating LLM Limitations for NIDS*

The following are some specific studies that aim to mitigate the known limitations of using LLM for NIDS.

*1) Use hybrid architectures: LLMs as augmenting modules, not sole detectors:* Don't simply replace traditional feature- and statistics-based detectors with LLMs. Instead, combine fast, lightweight front-line detectors (process/rule/anomaly engines) with LLMs for costly tasks such as context analysis, labeling, interpretation, and classification. This reduces latency and cost while leveraging the strengths of LLMs (semantic understanding/interpretation). Recent NIDS surveys and experimental frameworks recommend a hybrid stack as the most practical deployment approach [116] (2024) and [160] (2025).

*2) Domain-adapt LLMs via continual pre-training or task-adaptive pre-training before fine-tuning:* Close the representation gap between natural language pre-training and network data by performing domain-adaptive CPT (continuous pre-training) or TAPT (task-adaptive pre-training) on a corpus of security text, protocol traces, and sanitized payloads before any fine-tuning. LLM-NIDS papers [137] (2024) and [160] (2025) show that CPT can improve token representations of protocol names, CVE IDs, and log patterns, and reduce phantom reads when interpreting network artifacts. Use a tokenizer and input format tailored to the stream/payload (e.g., hexadecimal/byte-level tokenization plus semantic tags).

*3) Parameter-efficient adaptation for frequent updates:* To adapt models to current conditions without expensive retraining, PEFT (Parameter Efficient Fine-Tuning) methods (LoRA, Adapter, Prefix/Hint Tuning, P-Tuning) can be used. This allows only a small set of parameters to be updated during continuous learning or the infusion of new threat intelligence. This reduces computational effort and the risk of catastrophic forgetting while enabling rapid deployment of updates tailored to new threats. Multiple LLM security studies explicitly recommend using LoRA/Adapter for edge IDS adaptation, [137] (2024) and [116] (2024).

*4) RAG architectures for freshness and transparency:* RAG (Retrieval-Augmented Generation): Combines LLM with a retrieval store containing the latest threat intelligence, signatures, and network context. Retrieval aligns answers with current indicators, reduces hallucinations, and provides auditability (you can correlate model outputs with retrieved evidence). [160] (2025) and [161] (2025).

*5) Harden inputs: sanitization, canonicalization, and adversarial filtering:* Treat network inputs as adversarial channels. Before feeding data into the LLM, normalize and sanitize the payload (normalize encoding, remove aliasing artifacts), and run adversarial example detectors or perturbation-resistant preprocessing. Recent LLM security guidelines and NIST guidance emphasize input sanitization and dedicated adversarial filtering modules as effective first-line defenses against evasion and just-in-time injection. [162] (2024) and [163] (2025).

*6) Adversarial training and red-teaming in the development loop:* Active and persistent adversarial training and red team LLM components using reality evasion techniques (obfuscation, byte-level payload mutation, carefully crafted prompts) are reported to be highly effective [160] (2025) and [163] (2025).

*7) Efficiency engineering: distillation, quantization, and tiered inference:* Use model distillation and aggressive quantization for low-latency inference; retain a small distilled model for real-time filtering and upgrade to a larger LLM for deep forensics or analyst queries. A paper on deploying Transformers for NIDS claims this layered approach offers the best trade-off between throughput and performance. Careful verification that compression does not reduce adversarial robustness is provided in [116] (2024) and [137] (2024).

Understanding these limitations is critical because LLM weaknesses can be exploited offensively. The following section explores how attackers leverage LLMs for penetration testing, attack traffic generation, malware generation, and phishing, and why defensive design must anticipate these threats.

## V. Utilizing LLMs for Offensive Tools

The rapid adoption of LLM agents and multi-agent systems has enabled remarkable capabilities in natural language processing and generation. However, they can also be exploited as offensive tools, leading to unprecedented security challenges. LLMs have been shown by [164] (2023) to be particularly beneficial during the reconnaissance phase. Using a case study approach, the study explores how LLMs can assist in collecting valuable reconnaissance data, including IP address ranges, domain names, vendor technologies, network topology, SSL/TLS ciphers, ports, and services, and the operating system used by the target. This information helps in the planning phase of a penetration test or network attack, guiding the selection of appropriate strategies, tools, and techniques to uncover risks such as unpatched software and misconfigurations. Penetration testing (or pen testing) is a form of controlled cyberattack used to assess the security of computer systems. Moreover,LLMs can also be used to generate attack traffic and malware, and even directly carry out attacks[165] (2024). The following subsections provide a comprehensive overview of related work.

### A. Utilizing LLMs for Penetration Testing

Penetration testing, typically performed by testers or red teams, aims to identify weaknesses in systems, networks, or applications and provide comprehensive reports and recommended improvements before malicious attackers can exploit them. By simulating real-world intrusions, penetration testing can help organizations discover vulnerabilities and strengthen their defenses against potential threats [166](2025). Traditional penetration testing is expert-driven and resource-intensive, requiring specialized knowledge, tools, and careful coordination. However, as systems become increasingly complex and the need for frequent testing grows, there is growing interest in automating or enhancing penetration testing with machine assistance. Recent developments have shown that LLMs can support automated penetration testing, offering new possibilities for improving efficiency and scalability.[167] (2023) evaluated the ability of LLMs (specifically Google's Bard and ChatGPT) to generate malicious payloads for penetration testing. The results showed that ChatGPT can generate more targeted and complex payloads, potentially helping attackers build sophisticated exploits. Additionally, [168] (2023) investigated the intersection of LLMs and privilege escalation. The study introduced a fully automated privilege escalation tool designed to benchmark the effectiveness of various LLMs in exploiting vulnerabilities. The results showed that GPT-4 Turbo outperformed GPT-3.5 Turbo and Llama-3 in identifying and exploiting privilege escalation opportunities. Furthermore, [169] (2023) proposed PentestGPT, an automated penetration testing tool based on LLMs. PentestGPT uses LLMs to manage and automate various parts of the penetration testing workflow. It divides tasks into multiple modules (such as reasoning, generation, and parsing) to compensate for the context loss inherent in LLM when handling long or complex tasks. Evaluations show that PentestGPT achieves significantly higher task completion rates than the baseline LLM on real-world targets. Specifically, by combining three self-interacting modules (reasoning, generation, and parsing), the research demonstrates strong performance on a penetration testing benchmark consisting of 13 scenarios and 182 subtasks.

[152] (2023) evaluated the potential of ChatGPT in penetration testing. The GPT model was applied to various stages of penetration testing, including pre-engagement interaction, threat modeling, intelligence gathering, vulnerability analysis, vulnerability exploitation, and post-exploitation. [170] (2023) introduced an LLM guidance tool designed to automate and prototype penetration testing tasks by using prompts to guide the LLM in discovering and exploiting vulnerabilities. Furthermore, challenges such as maintaining focus during testing, handling errors, and handling multi-step exploitation paths were explained. For example, LLM often repeated enumeration commands, resulting in ineffective exploitation of discovered vulnerabilities. [171] (2023) explored how LLM (e.g., GPT-3.5) can assist penetration testers at both a high-level (planning) and a low-level (identifying or executing specific exploits). They implemented a closed-loop system between LLM-generated commands and vulnerable virtual machines via SSH, where the state of the virtual machine is fed back to the LLM to guide subsequent actions. While preliminary, the work demonstrates the feasibility and challenges of enabling LLM to not only plan but also take action (execute) during certain parts of a penetration test. [172] (2023) discusses the use of ChatGPT to generate detailed attack scenarios. It demonstrates the effectiveness of a system that feeds asset management data (OS type, version, device usage, accounts) and vulnerability information published by CISA into ChatGPT, searches for high-threat attack paths, and then outputs attack paths that may be useful for penetration testing and red teaming.

[173] (2024) introduces an LLM-based agent called HackSynth, designed for autonomous penetration testing. The agent employs a dual-module architecture consisting of a Planner and a Summarizer, enabling iterative command generation and feedback processing. The authors also examine the safety and predictability of the agent's actions. The work highlights the potential of LLM-based agents in advancing autonomous penetration testing and underscores the importance of robust safeguards. [174] (2025) presents the VulnBot framework, which uses multiple specialized agents to simulate human-like collaboration during penetration testing. Its architecture splits tasks into multiple phases (reconnaissance, scanning, exploitation), uses a task graph to plan actions, and allows inter-agent communication. Compared to simpler approaches, the framework achieves improved efficiency and accuracy in real-world target experiments. [175] (2025) aims to improve the reasoning behavior of LLM agents during penetration testing by using structured attack trees derived from the MITRE ATT-CK framework. By using deterministic task tree constraint reasoning, agents can avoid hallucinatory or wasteful behavior and

more efficiently and successfully complete penetration testing subtasks in benchmarks such as HackTheBox. [176] (2025) proposed a multi-agent penetration testing system called xOffense, which uses an open-source LLM (Qwen3-32B) fine-tuned with data from the Thinking Chain. Agents are assigned reconnaissance, scanning, and exploitation phases. The system coordinates these phases and improves performance on benchmarks designed for automated penetration testing. [177] (2025) implements a multi-agent penetration testing framework (called CurriculumPT) that combines curriculum learning with LLM-based agents. The system enables agents to progressively acquire and apply exploitation skills across CVE-based tasks. It uses a structured progression from simple to complex vulnerabilities, allowing agents to build an experience knowledge base. Moreover, it supports generalization to new attack surfaces without fine-tuning, leveraging prior learned strategies. [178] (2025) introduces RapidPen, a fully automated penetration testing framework designed to achieve an initial foothold (IP-to-Shell) without human intervention. RapidPen leverages LLMs to autonomously discover and exploit vulnerabilities starting from a single IP address. It integrates advanced ReAct-style task planning with retrieval-augmented exploit knowledge bases and employs a command-generation loop with direct execution feedback. Through this architecture, RapidPen systematically scans services, identifies viable attack vectors, and executes targeted exploits in a fully automated manner.

### *B. Utilizing LLMs for Generating Attack Traffic and Malware*

Obviously, the technologies reviewed above that are used in penetration testing can also be used to carry out malicious network attacks. [179] (2023) explores how criminals are using generative AI to plan and execute ransomware attacks. The research shows that these AI tools significantly lower the barrier to entry for non-technical attackers. The research also found that individuals with IT expertise but lacking other skills can use generative AI to craft more convincing phishing emails. The widespread adoption of generative AI could lead to an increase in the number and sophistication of ransomware attacks. [180] (2023) examines the transformative role of generative AI in social engineering (SE) attacks. The research aims to deepen understanding of the risks associated with this emerging paradigm, its impact on humans, and potential countermeasures. The research demonstrates how AI capabilities in realistic content creation, advanced targeting, and automated attack infrastructure can significantly enhance the effectiveness of these attacks. [181] (2023) also demonstrates the application of generative AI models in SE attacks. These tools can generate highly persuasive, personalized content to enhance the effectiveness of phishing campaigns, including deepfake scams and voice cloning for voice phishing attacks. WormGPT is designed specifically for malicious campaigns, increasing the success rate of business email compromise (BEC) attacks through personalized, convincing emails.

[182] (2024) introduces AUTOATTACKER, a system that automatically generates attacks using Large Language Models (LLMs) for complex tasks such as lateral movement, credential acquisition, and other stages of the attack lifecycle. The work focuses on the post-compromise or operational phase of cyberattacks, exploring how LLMs can automate or assist attackers after the initial breach. The proposed framework leverages LLMs for planning, command generation, decision-making, and iterative progression through attack phases. The system simulates or emulates attacker behavior with LLM support. The research demonstrates how LLMs can lower the barrier to entry for sophisticated attacks and provides valuable insights for defense teams to anticipate emerging attack vectors. [183] (2025) investigates the use of fine-tuned Large Language Models (LLMs) for the automated generation of synthetic attacks, including Cross-Site Scripting (XSS), SQL injection, and command injection. A dedicated web application has been developed to enable penetration testers to quickly generate high-quality payloads without requiring in-depth knowledge of artificial intelligence. The fine-tuned model demonstrates the ability to produce synthetic payloads that closely mimic real-world attacks. This approach not only enhances the model's precision and reliability but also provides a practical resource for cybersecurity professionals to strengthen the security of web applications.

[184] (2023) discusses the emergence of GPT-based malware and highlights that as traditional malware detection systems evolve to address a wide range of sophisticated attacks, threat actors are now leveraging LLM to develop advanced strategies for infecting new malware. This shift poses significant challenges to traditional detection methods. [185] (2023) demonstrates how publicly available plugins can be combined with LLM, which acts as a proxy between attackers and victims. The study presents a proof-of-concept in which ChatGPT is used to spread malware while evading detection and establishes communication with a command and control (C2) server to receive instructions for interacting with the victim's

system. It also outlines the general approach and key elements required to remain undetected and successfully execute an attack. [186] (2023) demonstrates that current large text models can be used by attackers to generate malware. Furthermore, the model's ability to rewrite malware code in various ways is tested. It also highlights the difficulty of GPT-3 in generating complex malware from simple prompts. [187] (2023) demonstrates the powerful capabilities of LLM in generating complex and diverse malware. The ability to automate these processes and increase the sophistication of attacks poses a significant challenge to traditional cybersecurity defenses. It presents 13 examples of cybersecurity-related tasks that ChatGPT can attempt. [188] (2023) also demonstrates the power of LLM in malware creation, where the authors used LLM to create malware such as WannaCry, NotPetya, Ryuk, REvil, and Locky. The paper shows that ChatGPT can generate ransomware code snippets, including encryption procedures and ransom note generation. [189] (2023) investigates the potential for abusing advances in artificial intelligence by using ChatGPT to develop seven malware programs and two attack tools. The authors demonstrate that these models can generate functional malicious code in minutes.

### *C. Utilizing LLMs for Generating Phishing-Messages*

[190] (2018) involved using word vector representations of social media posts to train a model to generate spear-phishing messages. [191] (2023) demonstrated the potential of LLM to enhance and scale spear-phishing campaigns. By using GPT-3.5 and GPT-4 to create spear-phishing messages for over 600 UK Members of Parliament, the authors showed that these models significantly lowered the barrier to entry for cybercriminals. [171] (2023) demonstrated that ChatGPT can not only generate malware code and phishing emails but is also highly effective in SQL injection attacks. The authors also demonstrated that AI can generate polymorphic malware and craft convincing phishing emails. The article [192] (2023) describes a convincing fake email exchange created using generative AI in which company executives appear to discuss how to cover up financial deficits. With the help of an army of social media bots, the "leaked" information quickly spread, causing the company's stock price to plummet and causing permanent reputational damage. [193] (2024) showed that lateral phishing emails generated by LLM were as effective as those crafted by communications professionals, highlighting the key threat posed by LLM in leading phishing campaigns. It revealed that AI-crafted emails, particularly those leveraging internal organizational information, had a high success rate in persuading employees to respond. [194] (2023) is also an article on phishing emails. The paper examined the effectiveness of human-crafted phishing emails compared to those crafted by GPT-3.

### *D. Utilizing LLMs for Generating Malicious Network Traffic Capable of Evading IDS*

Since LLMs have the capability to generate flexible and natural-looking traffic, it is understandable that they are powerful tools for creating attack traffic capable of evading pattern-based IDS. In fact, they can generate multi-stage attacks, benign-looking malicious traffic, and code that makes detection difficult for traditional IDS. Moreover, they may also produce noise to degrade the detection performance of IDS.

Multi-stage attacks can hide their intent and make detection more difficult. [195] (2025) investigates whether LLM can act autonomously in multi-stage cyberattacks (reconnaissance, initial access, lateral movement, exfiltration) across multiple hosts. The paper introduces a system called Incalmo, a high-level abstraction layer that allows LLM to specify tasks (e.g., "infect a host," "scan a network"), which are then translated into realistic, low-level commands by agent modules. It also includes services such as environment state tracking and attack graph support. LLM using Incalmo was able to successfully execute multi-stage attacks in 9 out of 10 evaluated network environments. Even the smaller LLM (using Incalmo) was able to successfully execute in 5 out of 10 environments. [196] (2025) provides an overview of the threat capabilities of LLMs when used as autonomous agents in cyberattacks, covering reconnaissance, malicious content/code generation, coordinated or multi-stage attacks, and threats in infrastructure-less mobile or IoT networks. Existing defense approaches and their limitations are also discussed.

Recent studies indicate that LLMs can be integrated into frameworks that automate reconnaissance, exploit selection, fuzzing, and post-exploitation activities—capabilities that generate traffic or behaviors designed to evade ML-based or signature-based IDS [197] (2025). LLM-assisted generators can craft payloads that mimic benign application traffic (e.g., HTTP headers, legitimate cookies, or common IoT telemetry patterns) or reformulate exploit code to appear syntactically similar to legitimate samples while preserving semantic validity. Such emulation can defeat detectors that rely on surface-level labeled distributions. Prior work has shown that adversarial examples—small, constrained perturbations to inputs—can cause ML-based NIDS to misclassify malicious traffic as benign [198] (2022). LLMs and generator models can explore the constrained space of protocol-valid perturbations (e.g., minor payload bit/byte changes, timing jitter, header field tweaks) that preserve attack semantics while shifting features outside the detector's decision boundary. Practical evaluations indicate that such attacks are feasible in many ML-NIDS designs [199] (2025).

[200] (2024) designs an automated system, called EaTVul, to attack DNN-based software vulnerability detection systems. A surrogate model is first trained based on BiLSTM with an attention mechanism. The averaged attention scores from the attention layer are retrieved to identify the key features that contribute significantly to the prediction. These important features serve as inputs to ChatGPT, which generates adversarial data. The generated adversarial data will then be further reviewed and optimized, and ChatGPT is used again to regenerate the adversarial data. In the above-explained study [182] (2024), LLM acts as a controller that plans and executes attack sequences, integrating with real tools. [197] (2025) proposes an adversarial traffic generation framework, called AdvTG, to deceive DL-based malicious traffic based on the LLM. And a specialized prompt is designed for traffic generation tasks, where non-functional fields are generated to produce the mutated traffic, while functional fields and target types are supplied as input. This fine-tuning allows it to generate traffic that remains compliant and functional. Furthermore, reinforcement learning (RL) is utilized to make AdvTG automatically selects traffic fields that exhibit more robust adversarial properties. Experimental results show that AdvTG achieves over 40% attack success rate (ASR) across six detection models on six datasets. [201] (2025) provides an overview of both defensive and offensive uses of LLMs in cybersecurity: detection, red team task automation, and attack assistance (e.g., prompt-based reconnaissance, social engineering). The paper incorporates case studies and practical problems.

[202] (2025) shows that LLM-generated noise (text, code, images) can reduce the detection performance and Code generation tasks were found particularly effective in evading hardware-based detection. Furthermore, in [203] (2024), the authors introduce that their group generated JavaScript code utilizing LLMs that appears benign but is malicious, in order to bypass IDS. They produced benign-like and harder-to-detect code. [204] (2025) explains that LLM agents can be manipulated to install and execute malware while mimicking normal communication. Techniques like "RAG Backdoor" and "Inter-Agent Trust Exploitation" are used to evade detection. It demonstrate that adversaries can effectively coerce popular LLMs into autonomously installing and executing malware on victims.

## VI. Summary

This survey traces the evolution of Network Intrusion Detection Systems (NIDS) from traditional signature-based methods to neural network (NN)-based approaches and, more recently, to frameworks incorporating Large Language Models (LLMs). Signature-based systems remain foundational due to their transparency and operational maturity but suffer from poor adaptability to zero-day attacks and high maintenance costs. NN-based models introduced powerful learning capabilities and improved anomaly detection but continue to face challenges such as data imbalance, robustness, and interpretability. LLMs offer promising solutions through contextual reasoning and generative capabilities. These benefits are amplified when combined with domain-specific adaptation strategies, including continual pre-training, supervised fine-tuning, and prompt engineering. Studies on the use of LLMs in both defensive and offensive cybersecurity contexts are outlined. Additionally, the challenges associated with using LLMs for NIDS, along with recent research addressing these challenges, are reviewed."

## References

[1] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proc. the 13th USENIX conference on System administration*, 1999, pp. 229–238, accessed on 2025-10-19). [Online]. Available: https://www.usenix.org/legacy/event/lisa99/full_papers/roesch/roesch.pdf

[2] "Snort3," accessed on 2025-10-21. [Online]. Available: https://www.snort.org/snort3

[3] "Suricata-8.0.1," accessed on 2025-10-21. [Online]. Available: https://docs.suricata.io/en/suricata-8.0.1/

[4] "Zeek 8.0.3," accessed on 2025-10-21. [Online]. Available: https://zeek.org/get-zeek/

[5] M. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 31, no. 23, pp. 2435–2463, 1999. [Online]. Available: https://doi.org/10.1016/S1389-1286(99)00112-7

[6] "Security onion solutions," accessed on 2025-10-21. [Online]. Available: https://securityonionsolutions.com/

[7] O. Alnasser, J. A. Muhtadi, K. Saleem, and S. Shrestha, "Signature and anomaly-based intrusion detection system for secure iots and v2g communication," *Alexandria Engineering Journal*, vol. 125, pp. 424–440, 2025. [Online]. Available: https://doi.org/0.1016/j.aej.2025.03.068

[8] P. R. Kothamali and S. Banik, “Limitations of signature-based threat detection,” *AI for Medicine Health*, vol. 13, no. 01, pp. 381–391, 2022, accessed on 2025-10-1. [Online]. Available: https://redcrevistas.com/index.php/Revista/article/view/131

[9] “Sentinel overwatch services: What are the limitations of signature-based intrusion detection?” 2025, accessed on 2025-10-1. [Online]. Available: https://sentinel-overwatch.com/what-are-the-limitations-of-signature-based-intrusion-detection/

[10] Blog, “General international group: Advantages and disadvantages of intrusion detection system (ids) types,” 2025, accessed on 2025-10-11. [Online]. Available: https://generalintlgroup.com/en/blog/advantages-and-disadvantages-of-intrusion-detection-system-ids-types

[11] B. J. Kwon, J. Mondal, J. Jang, L. Bilge, and T. Dumitra, “The dropper effect: Insights into malware distribution with downloader graph analytics,” in *Proc. the ACM Conference on Computer and Communications Security (CCS)*, 2015, p. 1118–1129. [Online]. Available: https://doi.org/10.1145/2810103.2813724

[12] “Opswat threat landscape report — opswat,” 2025, accessed on 2025-10-19. [Online]. Available: https://www.opswat.com/resources/reports/2025-threat-landscape-report

[13] S. C. Lee and D. Heinbuch, “Training a neural-network based intrusion detector to recognize novel attacks,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 4, pp. 294–299, 2001. [Online]. Available: https://doi.org/10.1109/3468.935046

[14] S. Mukkamala, G. Janoski, and A. Sung, “Intrusion detection using neural networks and support vector machines,” in *Proc. the International Joint Conference on Neural Networks. IJCNN’02 (Cat. No.02CH37290)*, vol. 2, 2002, pp. 1702–1707 vol.2. [Online]. Available: https://doi.org/10.1109/IJCNN.2002.1007774

[15] M. Mehdi and Z. Mohammad, “A neural network based system for intrusion detection and classification of attacks,” in *Proc. IEEE International Conference on Advances in Intelligent Systems*, 2004, accessed 2025-10-20. [Online]. Available: https://api.semanticscholar.org/CorpusID:11962542

[16] N. Gao, L. Gao, Q. Gao, and H. Wang, “An intrusion detection model based on deep belief networks,” in *Proc. the 2nd International Conference on Advanced Cloud and Big Data*, 2014, pp. 247–252. [Online]. Available: https://doi.org/10.1109/CBD.2014.41

[17] Z. Wang, in *Blackhat 2015*, 2015. [Online]. Available: https://www.blackhat.com/docs/us-15/materials/us-15-Wang-The-Applications-Of-Deep-Learning-On-Traffic-Identification-wp.pdf

[18] M. Z. Alom, V. R. Bontupalli, and T. M. Taha, “Intrusion detection using deep belief networks,” in *Proc.National Aerospace and Electronics Conference (NAECON)*, 2015, pp. 339–344. [Online]. Available: https://doi.org/10.1109/NAECON.2015.7443094

[19] M. A. Salama, H. F. Eid, A. R. Rabie, D. Ashraf, and E. H. Aboul, “Hybrid intelligent intrusion detection scheme,” *Soft Computing in Industrial Applications*, vol. 96, pp. 293–303, 2011. [Online]. Available: https://doi.org/10.1007/978-3-642-20505-7_26

[20] Y. Jia, M. Wang, and Y. Wang, “Network intrusion detection algorithm based on deep neural network,” *IET Information Securrity*, vol. 13, no. 1, pp. 48–53, 2018. [Online]. Available: https://doi.org/10.1049/iet-ifs.2018.5258

[21] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018. [Online]. Available: https://doi.org/10.1109/TETCI.2017.2772792

[22] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, “Long short term memory recurrent neural network classifier for intrusion detection,” in *Proc. International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–5. [Online]. Available: https://doi.org/10.1109/PlatCon.2016.7456805

[23] P. Toupas, D. Chamou, K. M. Giannoutakis, A. Drosou, and D. Tzovaras, “An intrusion detection system for multi-class classification based on deep neural networks,” in *Poc. the 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2019, pp. 1253–1258. [Online]. Available: https://doi.org/10.1109/ICMLA.2019.00206

[24] J. Kim, J. Kim, H. L. T. Thu, , and H. Kim, “Long short term memory recurrent neural network classifier for intrusion detection,” in *Proc. International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–5. [Online]. Available: https://doi.org/10.1109/PlatCon.2016.7456805

[25] M. Du, F. Li, G. Zheng, and V. Srikumar, “Deeplog: Anomaly detection and diagnosis from system logs through deep learning,” in *Proc. the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1285–1298. [Online]. Available: https://doi.org/10.1145/3133956.3134015

[26] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017. [Online]. Available: https://doi.org/10.1109/ACCESS.2017.2762418

[27] C. Xu, J. Shen, X. Du, and F. Zhang, “An intrusion detection system using a deep neural network with gated recurrent units,” *IEEE Access*, vol. 6, pp. 48 697–48 707, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2867564

[28] Y. Zhang, X. Chen, L. Jin, X. Wang, and D. Guo, “Network intrusion detection: Based on deep hierarchical network and original flow data,” *IEEE Access*, vol. 7, pp. 37 004–37 016, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2905041

[29] K. Jiang, W. Wang, A. Wang, and H. Wu, “Network intrusion detection combined hybrid sampling with deep hierarchical network,” *IEEE Access*, vol. 8, pp. 32 464–32 476, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2973730

[30] F. Laghrissi, S. D. andK. Douzi, and B. Hssina, “Intrusion detection systems using long short-term memory (lstm),” *Jounal of Big Data*, vol. 8, no. 65, 2021. [Online]. Available: https://doi.org/10.1186/s40537-021-00448-4

[31] S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo, Q. Pham, and N. Dao, “A lstm-fcnn based multi-class intrusion detection using scalable framework,” *Computers and Electrical Engineering*, vol. 99, p. 107720, 2022. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2022.107720

[32] S. Ullah, J. Ahmad, M. A. Khan, M. S. Alshehri, W. Boulila, A. Koubaa, S. U. Jan, and M. M. Ch, “Tnn-ids: Transformer neural network-based intrusion detection system for mqtt-enabled iot networks,” *Computer Networks*, vol. 237, no. 110072, 2023. [Online]. Available: https://doi.org/10.1016/j.comnet.2023.110072

[33] E. M. Mercha, E. M. CHakir, , and M. Erradi, “Trans-ids: A transformer-based intrusion detection system,” in *Proc. the 20th International Conference on Security and Cryptography*, 2023, pp. 402–409. [Online]. Available: https://doi.org/10.5220/0012085800003555

[34] F. Ullah, S. Ullah, G. Srivastava, and J. C. Lin, "Ids-int: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digital Communications and Networks*, vol. 10, no. 1, pp. 190–204, 2024. [Online]. Available: https://doi.org/10.1016/j.dcan.2023.03.008

[35] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proc. the 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 195–200. [Online]. Available: https://doi.org/10.1109/ICMLA.2016.0040

[36] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82 512–82 521, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2923640

[37] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30 373–30 385, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2899721

[38] G. Andresini, A. Appice, N. D. Mauro, C. Loglisci, and D. Malerba, "Multi-channel deep feature learning for intrusion detection," *IEEE Access*, vol. 8, pp. 53 346–53 359, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2980937

[39] S. PHETLASY, S. Ohzahata, C. Wu, and T. Kato, "A sequential classifiers combination method to reduce false negative for intrusion detection system," *IEICE Transactions on Information and Systems*, vol. E102, no. 5, p. 888–897, 2019. [Online]. Available: https://doi.org/10.1587/transinf.2018NTP0019

[40] H. Z. Hao, Y. Feng, H. Koide, and K. Sakurai, "A sequential detection method for intrusion detection system based on artificial neural networks," *Int. J. Netw. Comput.*, vol. 10, no. 2, pp. 213–226, 2020. [Online]. Available: https://doi.org/10.15803/ijnc.10.2_213

[41] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine learning-based iot-botnet attack detection with sequential architecture," *Sensors*, vol. 20, no. 16, p. 4372 2020. [Online]. Available: https://doi.org/10.3390/s20164372

[42] X. Cai, Y. Feng, and K. Sakurai, "Sequential detection of cyber-attacks using a classification filter," in *Proc. The 6th IEEE Cyber Science and Technology Congress*, 2021, pp. 659–666. [Online]. Available: https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00111

[43] R. Zhao, Y. Mu, L. Zou, and X. Wen, "A hybrid intrusion detection system based on feature selection and weighted stacking classifier," *IEEE Access*, vol. 10, pp. 71 414–71 426, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3186975

[44] A. Z. Kiflay, A. Tsokanos, and R. Kirner, "A network intrusion detection system using ensemble machine learning," in *Proc. International Carnahan Conference on Security Technology (ICCST)*, 2021, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICCST49569.2021.9717397

[45] O. Ammar and T. Anas Abu, "Ensemble-based deep learning models for enhancing iot intrusion detection," *Appl. Sci.*, vol. 13, no. 21, p. 11985, 2023. [Online]. Available: https://doi.org/10.3390/app132111985

[46] A. Aldaej, I. Ullah, and M. Atiquzzaman, "Ensemble technique of intrusion detection for iot-edge platform," *Scientific Reports*, vol. 14, no. 11703, 2024. [Online]. Available: https://doi.org/10.1038/s41598-024-62435-y

[47] Y. Lyu, Y.Feng, and K. Sakurai, "Design and performance evaluation of a two-stage detection of ddos attacks using a trigger with a feature on riemannian manifolds," in *Proc. the 38th International Conference on Advanced Information Networking and Applications (4)*, 2024, pp. 133–144. [Online]. Available: https://doi.org/10.1007/978-3-031-57916-5_12

[48] M. Niu, Y. Feng, and K. Sakurai, "A two-stage detection system of ddos attacks in sdn using a trigger with multiple features and self-adaptive thresholds," in *Proc. the 17th International Conference on Ubiquitous Information Management and Communication*, 2023, pp. 1–8. [Online]. Available: https://doi.org/10.1109/IMCOM56909.2023.10035661

[49] P. Barnard, N. Marchetti, and L. A. DaSilva, "Robust network intrusion detection through explainable artificial intelligence (xai)," *IEEE Networking Letters*, vol. 4, no. 3, pp. 167–171, 2022. [Online]. Available: https://doi.org/10.1109/LNET.2022.3186589

[50] M. Siganos, P. R. Grammatikis, I. Kotsiuba, E. Markakis, I. Moscholios, S. Goudos, and P. Sarigiannidis, "Explainable ai-based intrusion detection in the internet of things," in *Proc. the 18th ACM Int. Conf. on Availability, Reliability and Security (ARES)*, 2023, pp. 1–10. [Online]. Available: https://doi.org/10.1145/3600160.3605162

[51] H. C. Tanuwidjaja, T. Takahashi, T.-N. Lin, B. Lee, and T. Ban, "Hybrid explainable intrusion detection system: Global vs. local approach," in *Proc. the ACM workshop on Recent Advances in Resilient and Trustworthy ML Systems in Autonomous Network*, 2023, p. 37 – 42. [Online]. Available: https://doi.org/10.1145/3605772.3624004

[52] O. Arreche, T. R. Guntur, J. W. Roberts, and M. Abdallah, "E-xai: Evaluating black-box explainable ai frameworks for network intrusion detection," *IEEE Access*, vol. 12, pp. 23 954–23 988, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3365140

[53] S. Neupane, J. Ables, W. Anderson, S. Mittal, S. Rahimi, and I. Banicescu, "Explainable intrusion detection systems (x-ids): A survey of current methods, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 112 392–112 415, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3216617

[54] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "Rtids: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64 375–64 387, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3182333

[55] H. Jo and D. H. Kim, "Intrusion detection using transformer in controller area network," *IEEE Access*, vol. 12, pp. 121 932–121 946, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3452634

[56] Z. Long, H. Yan, G. Shen, X. Zhang, H. He, and L. Cheng, "A transformer-based network intrusion detection approach for cloud security," *Journal of Cloud Computing*, vol. 13, no. 5, 2024. [Online]. Available: https://doi.org/10.1186/s13677-023-00574-9

[57] I. Koukoulis, I. Syrigos, and T. Korakis, "Self-supervised transformer-based contrastive learning for intrusion detection systems," *arXiv:2505.08816*, 2025. [Online]. Available: https://arxiv.org/abs/2505.08816

[58] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symposium on Security and Privacy*, 2010, pp. 305–316. [Online]. Available: https://doi.org/10.1109/SP.2010.25

[59] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of

network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, no. C, pp. 19–31, 2016. [Online]. Available: https://doi.org/10.1016/j.jnca.2015.11.01

[60] Y. Otoum and A. Nayak, "As-ids: Anomaly and signature based ids for the internet of things," *Journal of Network and Systems Management*, vol. 29, no. 3, p. 23, 2021. [Online]. Available: https://doi.org/10.1007/s10922-021-09589-6

[61] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[62] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[63] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers and Security*, vol. 86, pp. 147–167, 2019. [Online]. Available: https://doi.org/10.1016/j.cose.2019.06.005

[64] P. R. Kothamali, S. Banik, and S. V. Nadimpalli, "Challenges in applying ml to cybersecurity," *AI for Medicine Health*, vol. 11, no. 01, pp. 214–256, 2020, accessed on 2025-10-1. [Online]. Available: https://redcrevistas.com/index.php/Revista/article/view/133

[65] J. Sun, Y. Tang, and S. Wang, "Model robustness optimization method using gan and feature pyramid," *Journal of Computer Engineering and Applications*, vol. 17, no. 5, p. 1139–1146, 2023. [Online]. Available: https://doi.org/10.3778/j.issn.1673-9418.2106063

[66] T. Wongvorachan, S. He, and O. Bulut, "A comparison of undersampling, oversampling, and smote methods for dealing with imbalanced classification in educational data mining," *Information*, vol. 14, no. 1, p. 54, 2023. [Online]. Available: https://doi.org/10.3390/info14010054

[67] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers and Security*, vol. 31, no. 3, p. 357–374, 2012. [Online]. Available: https://doi.org/10.1016/j.cose.2011.12.012

[68] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 21, no. 9, pp. 1263–1284, 2009. [Online]. Available: https://doi.org/10.1109/TKDE.2008.239

[69] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004. [Online]. Available: https://doi.org/10.1145/1007730.1007735

[70] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Springer, 2019.

[71] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. the 14th International Conference on Machine Learning (ICML)*, 1997, pp. 179–186, accessed on 2025-10-19. [Online]. Available: https://api.semanticscholar.org/CorpusID:18370956

[72] C. Drummond and R. Holte, "C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. workshop on learning from imbalanced datasets," in *Proc. the 20th International Conference on Machine Learning (ICML)*, 2003, accessed on 2025-10-19. [Online]. Available: https://api.semanticscholar.org/CorpusID:204083391

[73] D. Theng and K. K. Bhoyar, "Feature selection techniques for machine learning: a survey of more than two decades of research," *Knowl Inf Syst*, vol. 66, p. 1575–1637, 2024. [Online]. Available: https://doi.org/10.1007/s10115-023-02010-5

[74] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016. [Online]. Available: https://doi.org/10.1109/COMST.2015.2494502

[75] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 179, no. 107247, 2020. [Online]. Available: https://doi.org//10.1016/j.comnet.2020.107247

[76] Y. Zhang, R. C. Muniyandi, and F. Qamar, "A review of deep learning applications in intrusion detection systems: Overcoming challenges in spatiotemporal feature extraction and data imbalance," *Appl. Sci.*, vol. 15, no. 3, p. 1552, 2025. [Online]. Available: https://doi.org/10.3390/app15031552

[77] S. Sharipuddin and E. A. Winanto, "Enhanced deep learning intrusion detection in iot heterogeneous network with feature extraction," *Indonesian Journal of Electrical and Engineering and Informatics*, vol. 9, no. 3, pp. 747–755, 2021, accessed on 2025-10-20. [Online]. Available: https://section.iaesonline.com/index.php/IJEEI/article/view/3134/643

[78] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017. [Online]. Available: https://doi.org/10.1109/ACCESS.2017.2762418

[79] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proc. the International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–5. [Online]. Available: https://doi.org/10.1109/PlatCon.2016.7456805

[80] M. Du, F. Li, G. Zheng, , and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. the ACM Conference on Computer and Communications Security (CCS)*, 2017, p. 1285–1298. [Online]. Available: https://doi.org/10.1145/3133956.3134015

[81] M. Wang, N. Yang, D. H. Gunasinghe, and N. Weng, "On the robustness of ml-based network intrusion detection systems: An adversarial and distribution shift perspective," *Computers*, vol. 12, no. 10, p. 209, 2023. [Online]. Available: https://doi.org/10.3390/computers12100209

[82] H. Mohammadian, A. A. Ghorbani, and A. H. Lashkari, "A gradient-based approach for adversarial attack on deep learning-based network intrusion detection systems," *Applied Soft Computing*, vol. 137, no. 110173, 2023. [Online]. Available: https://doi.org/10.1016/j.asoc.2023.110173

[83] P. Antonio, A. Sergio, G. D. Vicente, and G. Alberto, "Apollon: A robust defense system against adversarial machine learning attacks in intrusion detection systems," *Computers and Security*, vol. 136, no. 103546, 2024. [Online]. Available: https://doi.org/10.1016/j.cose.2023.103546

[84] T. Ali and P. Kostakos, "Huntgpt: Integrating machine learning-based anomaly detection and explainable ai with large language models (llms)," *arXiv:2309.16021*, 2023. [Online]. Available: https://arxiv.org/abs/2309.16021

[85] W. X. Zhao, W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," *arXiv:2303.18223*, 2023. [Online]. Available: https://arxiv.org/abs/2303.18223

[86] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting

elicits reasoning in large language models," *arXiv:2201.11903*, 2023. [Online]. Available: https://arxiv.org/abs/2201.11903

[87] M. Shanahan, "Talking about large language models," *arXiv:2212.03551*, 2023. [Online]. Available: https://arxiv.org/abs/2212.03551

[88] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," *arXiv:2402.06196*, 2024. [Online]. Available: https://arxiv.org/abs/2402.06196

[89] J. Hoffmann, J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training compute-optimal large language models," *arXiv:2203.15556*, 2022. [Online]. Available: https://arxiv.org/abs/2203.15556

[90] M. Nam, S. Park, and D. S. Kim, "Intrusion detection method using bi-directional gpt for in-vehicle controller area networks," *IEEE Access*, vol. 9, pp. 124 931–124 944, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3110524

[91] R. Fayyazi and S. Yang, "On the uses of large language models to interpret ambiguous cyberattack descriptions," *arXiv:2306.14062*, 2023. [Online]. Available: https://arxiv.org/abs/2306.14062

[92] Y. Chen, A. Arunasalam, and Z. B. Celi, "Can large language models provide security and privacy advice? measuring the ability of llms to refute misconceptions," *arXiv:2310.02431*, 2023. [Online]. Available: https://arxiv.org/abs/2310.02431

[93] B. Breve, G. Cimino, G. Desolda, V. Deufemia, and A. Elefante, "On the user perception of security risks of tap rules: A user study," in *Proc. International Symposium on End User Development*, 2023, pp. 162–167. [Online]. Available: https://doi.org/10.1007/978-3-031-34433-6_10

[94] A. Ehsan and A. S. Ehabr, "Cve-driven attack technique prediction with semantic information extraction and a domain-specific language model," *arXiv:2309.02785*. [Online]. Available: https://arxiv.org/abs/2309.02785

[95] Y. Chen, A. Arunasalam, and Z. Celik, "Can large language models provide security and privacy advice? measuring the ability of llms to refute misconceptions," in *Proc. the 39th Annual Computer Security Applications Conference*, 2023, p. 366–378. [Online]. Available: https://doi.org/10.1145/3627106.3627196

[96] O. G. Lira, A. Marroquin, and M. A. To, "Harnessing the advanced capabilities of llm for adaptive intrusion detection systems," in *Proc. the 38th International Conference on Advanced Information Networking and Applications (AINA)*, 2024. [Online]. Available: https://doi.org/10.1007/978-3-031-57942-4_44

[97] J. Zhang, H. Bu, H. Wen, Y. Liu, H. Fei, R. Xi, L. Li, Y. Yang, H. Zhu, and D. Meng, "When llms meet cybersecurity: a systematic literature review," *Cybersecurity*, vol. 8, no. 55, 2025. [Online]. Available: https://doi.org/10.1186/s42400-025-00361-w

[98] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, p. 328–339, accessed on 2025-10-19. [Online]. Available: https://aclanthology.org/P18-1031.pdf

[99] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proc. the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, p. 8342–8360. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.740

[100] E. Hu, Y. Shen, P. Wallis, Z. A. Zhu, Y. Li, S. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *Proc. the 10th International Conference on Learning Representations (ICLR)*, 2022, https://arxiv.org/pdf/2106.09685v1/1000.

[101] C. Yıldız, N. K. Ravichandran, N. Sharma, M. Bethge, and B. Ermis, "Investigating continual pretraining in large language models: Insights and implications," *arXiv:2402.17400*, 2024. [Online]. Available: https://arxiv.org/abs/2402.17400

[102] T. Zhang, X. Chen, C. Qu, A. Yuille, and Z. Zhou, "Leveraging ai predicted and expert revised annotations in interactive segmentation: Continual tuning or full training?" *arXiv:2402.19423*, 2024. [Online]. Available: https://arxiv.org/abs/2402.19423

[103] T. Wu, L. Luo, Y.-F. Li, S. Pan, T.-T. Vu, and G. Haffari, "Continual learning for large language models: a survey," *arXiv:2402.01364*, 2024. [Online]. Available: https://arxiv.org/abs/2402.01364

[104] A. Ibrahim, B. Thérien, K. Gupta, M. L. Richter, Q. Anthony, T. Lesort, E. Belilovsky, and I. Rish, *arXiv:2403.08763*, 2024. [Online]. Available: https://arxiv.org/abs/2403.08763

[105] Z. Maasaoui, M. Merzouki, A. Battou, and A. Lbath, "Anomaly based intrusion detection using large language models," in *Proc. the 21st IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, 2024, pp. 1–8. [Online]. Available: https://doi.org/10.1109/AICCSA63423.2024.10912623

[106] Y. C. Kim, C. J. Lee, and Y. Yoon, "Payload-aware intrusion detection with cmae and large language models," *ACM Transactions on Privacy and Security*, 2025. [Online]. Available: https://doi.org/https://doi.org/10.1145/3769682

[107] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," in *Proc. the 10th International Conference on Learning Representations (ICLR)*, 2022, accessed on 2025-10-19. [Online]. Available: https://api.semanticscholar.org/CorpusID:237416585

[108] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang, "Instruction tuning for large language models:a survey," *arXiv: 2308.10792*, 2023. [Online]. Available: https://arxiv.org/abs/2308.10792

[109] G. Dong, H. Yuan, K. Lu, C. Li, M. Xue, D. Liu, W. Wang, Z. Yuan, C. Zhou, and J. Zhou, "How abilities in large language models are affected by supervised fine-tuning data composition," *arXiv:2310.05492*, 2023. [Online]. Available: https://arxiv.org/abs/2310.05492

[110] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J. Low, L. Bing, and L. Si, "On the effectiveness of adapter-based tuning for pretrained language model adaptation," in *Proc. the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP)*, 2021, p. 2208–2222. [Online]. Available: https://doi.org/10.18653/v1/2021.acl-long.172

[111] X. Liu, K. Ji, Y. Fu, Z. Du, Z. Yang, and J. Tang, "P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks," *arXiv:2110.07602*, 2021. [Online]. Available: https://arxiv.org/abs/2110.07602v2

[112] B. Lester, R. Al-Rfou, and N. Constant, "The power

of scale for parameter efficient prompt tuning," in *Proc. the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021, p. 3045–3059. [Online]. Available: https://doi.org/10.18653/V1/2021

[113] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: efficient finetuning of quantized llms," in *Proc. the 37th International Conference on Neural Information Processing System (NIPS)*, 2023, pp. 10 088–10 115. [Online]. Available: https://arxiv.org/abs/2305.14314

[114] E. Nwafor, U. Baskota, M. S. Parwez, J. Blackstone, and H. Olufowobi, "Evaluating large language models for enhanced intrusion detection in internet of things networks," in *Proc. IEEE Global Communications Conference*, 2024, pp. 3358–3363. [Online]. Available: https://doi.org/10.1109/GLOBECOM52923.2024.10901300

[115] Y. A. Farrukh, S. Wali, I. Khan, and N. D. Bastian, "Xg-nid: Dual-modality network intrusion detection using a heterogeneous graph neural network and large language model," *Expert Systems with Applications*, vol. 287, no. 128089, 2025. [Online]. Available: https://doi.org/10.1016/j.eswa.2025.128089

[116] H. Djallel, M. A. Ferrag, B. Nadjette, and S. Hamid, "Advancing cybersecurity with llms: A comprehensive review of intrusion detection systems and emerging applications," in *Proc. International Conference on Informatics and Applied Mathematics (IAM)*, 2024, accessed on 2025-10-19. [Online]. Available: https://ceur-ws.org/Vol-3922/paper7.pdf

[117] J. Zhang, "Leveraging large language models for autonomous threat detection in iot networks," in *Proc. the 8th International Conference on Electronic Information Technology and Computer Engineering*, 2024, pp. 545–550. [Online]. Available: https://doi.org/10.1145/3711129.3711223

[118] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *arXiv:2107.13586*, 2021. [Online]. Available: https://arxiv.org/abs/2107.13586

[119] Q. Ye, M. Axmed, R. Pryzant, and F. Khani, "Prompt engineering a prompt engineer," *arXiv:2311.05661*, 2023. [Online]. Available: https://arxiv.org/abs/2311.05661

[120] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly," *arXiv:1707.00600.*, 2020. [Online]. Available: https://arxiv.org/abs/1707.00600

[121] H. Xu, S. Wang, N. Li, K. Wang, Y. Zhao, K. Chen, T. Yu, Y. Liu, and H. Wang, "Large language models for cyber security: A systematic literature review," *arXiv:2405.04760*, 2024. [Online]. Available: https://arxiv.org/abs/2405.04760v4

[122] Y. Yao, P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, and N. Zhang, "Editing large language models: Problems, methods, and opportunities," in *Proc. the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023, pp. 10 222–10 240. [Online]. Available: https://doi.org/10.18653/V1/2023.EMNLP-MAIN.632

[123] N. Zhang, Y. Yao, B. Tian, P. Wang, S. Deng, M. Wang, Z. Xi, S. Mao, J. Zhang, Y. Ni, S. Cheng, Z. Xu, X. Xu, J.-C. Gu, Y. Jiang, P. Xie, F. Huang, L. Liang, Z. Zhang, X. Zhu, J. Zhou, and H. Chen, "A comprehensive study of knowledge editing for large language models," *arXiv:2401.01286*, 2024. [Online]. Available: https://arxiv.org/abs/2401.01286

[124] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[125] A. Natasha, M. Maria, G. Hadi, and J. L. Danger, "Can-bert do it? controller area network intrusion detection system based on bert language model," in *Proc. the 19th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, 2022, p. 1–8. [Online]. Available: https://doi.org/10.1109/AICCSA56895.2022.10017800

[126] L. G. Nguyen and K. Watabe, "A method for network intrusion detection using flow sequence and bert framework," in *Proc. IEEE International Conference on Communications (ICC)*, 2023, pp. 3006–3011. [Online]. Available: https://doi.org/10.1109/ICC45041.2023.10279335

[127] P. Cheng, L. Hua, H. Jiang, and G. Liu, "Lsf-idm: Automotive intrusion detection model with lightweight attribution and semantic fusion," *arXiv:2308.01237*, 2023. [Online]. Available: https://arxiv.org/abs/2308.01237

[128] M. A. Ferrag, M. Ndhlovu, N. Tihanyi, A. Dhabi, L. C. Cordeiro, M. Debbah, and T. Lestable, "Revolutionizing cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices," *IEEE Access*, vol. 12, pp. 23 733–23 750, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3363469

[129] L. D. Manocchio, S. layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann, "Flowtransformer: A transformer framework for flow-based network intrusion detection systems," *Expert Syst. Appl.*, vol. 241, no. C, 2024. [Online]. Available: https://doi.org/10.1016/j.eswa.2023.122564

[130] Y. Yang and X. Peng, "Bert-based network for intrusion detection system," *EURASIP J. on Info. Security*, vol. 11, 2025. [Online]. Available: https://doi.org/10.1186/s13635-025-00191-w

[131] S. Sattarpour, A. Barati, and H. E. Barati, "Efficient bert-based intrusion detection system in the network and application layers of iot," *Cluster Computing*, vol. 28, no. 138, 2025. [Online]. Available: https://doi.org/10.1007/s10586-024-04775-y

[132] A. Frederic, E. Moez, and M. B. Leila, "Efficient federated intrusion detection in 5g ecosystem using optimized bert-based model," *arXiv:2409.19390*, 2024. [Online]. Available: https://arxiv.org/abs/2409.19390

[133] U. Ünal and H. Dağ, "Anomalyadapters: Parameter-efficient multi-anomaly task detection," *IEEE Access*, vol. 10, pp. 5635–5646, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3141161

[134] V. Juttner, M. Grimmer, and E. Buchmann, "Chatids: explainable cybersecurity using generative ai," *arXiv:2306.14504*, 2023. [Online]. Available: https://arxiv.org/abs/2306.14504

[135] Y. Bai, M. Sun, L. Zhang, Y. Wang, S. Liu, Y. Liu, J. Tan, Y. Yang, and C. Lv, "Enhancing network attack detection accuracy through the integration of large language models and synchronized attention mechanism," *Appl. Sci.*, vol. 14, p. 3289, 2024. [Online]. Available: https://doi.org/10.3390/app14093829

[136] Q. Li, Y. Zhang, Z. Jia, Y. Hu, L. Zhang, J. Zhang, Y. Xu, Y. Cui, Z. Guo, and X. Zhang, "Dollm: How large language models understanding network flow data to detect carpet bombing ddos," *arXiv:2405.07638*, 2024. [Online]. Available: https://arxiv.org/abs/2405.07638

[137] A. Frederic, E. Moez, and M. Leila, "Llm-based continuous intrusion detection framework for next-gen networks," *arXiv:2411.03354*, 2024. [Online]. Available: https://arxiv.org/abs/2411.03354

[138] X. Zhang, Q. Li, Y. Tan, Z. Guo, L. Zhang, and Y. Cui, “Large language models powered network attack detection: Architecture, opportunities and case study,” *arXiv:2503.18487*, 2025. [Online]. Available: https://arxiv.org/abs/2503.18487

[139] V. Cobilean, H. S. Mavikumbure, C. S. Wickramasinghe, B. J. Varghese, T. Pennington, and M. Manic, “Anomaly detection for in-vehicle communication using transformers,” in *Proc. the 49th Annual Conference of the IEEE Industrial Electronics Society*, 2023, pp. 1–6. [Online]. Available: https://doi.org/10.1109/IECON51785.2023.10311788

[140] Q. Lai, C. Xiong, J. Chen, W. Wang, J. Chen, T. R. Gadekallu, M. Cai, and X. Hu, “Improved transformer-based privacy-preserving architecture for intrusion detection in secure v2x communications,” *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1810–1820, 2024. [Online]. Available: https://doi.org/10.1109/TCE.2023.3324081

[141] M. Fu, P. Wang, M. Liu, Z. Zhang, and X. Zhou, “Iov-bert-ids: Hybrid network intrusion detection system in iov using large language models,” *IEEE Transactions on Vehicular Technology*, vol. 74, no. 2, pp. 1909–1921, 2025. [Online]. Available: https://doi.org/10.1109/TVT.2024.3402366

[142] M. Wang, N. Yang, and N. Weng, “Securing a smart home with a transformer-based iot intrusion detection system,” *Electronics*, vol. 12, no. 9, p. 2100, 2023. [Online]. Available: https://doi.org/10.3390/electronics12092100

[143] S. Ullah, J. Ahmad, M. Khan, M. Alshehri, W. Boulila, A. Koubaa, S. Jan, and M. Ch, “Tnn-ids: Transformer neural network-based intrusion detection system for mqtt-enabled iot networks,” *The International Journal of Computer and Telecommunications Networking*, vol. 237, no. C, p. 110072, 2023. [Online]. Available: https://doi.org/10.1016/j.comnet.2023.110072

[144] Y. Yan, Y. Yang, Y. Gu, and F. Shen, “A multi-transformer fusion intrusion detection model for industrial internet,” in *Proc. the 5th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT)*, 2023, pp. 197–203. [Online]. Available: https://doi.org/10.1109/ECNCT59757.2023.10281050

[145] Z. Maasaoui, A. Battou, M. Merzouki, and A. LBATH, “Anomaly based intrusion detection using large language models,” in *Proc. the ACS/IEEE 21st International Conference on Computer Systems and Applications (AICCSA)*, 2024, pp. 1–8. [Online]. Available: https://doi.org/10.1109/AICCSA63423.2024.10912623

[146] A. Diaf, A. A. Korba, N. E. Karabadji, , and Y. Ghamri-Doudane, “Beyond detection: Leveraging large language models for cyber attack prediction in iot networks,” *arXiv:2408.14045v1*, 2024. [Online]. Available: https://arxiv.org/abs/2408.14045v1

[147] A. Diaf, A. A. Korba, N. E. Karabadji, and Y. Ghamri-Doudane, “Bartpredict: Empowering iot security with llm-driven cyber threat prediction,” *arXiv:2501.01664v1*, 2025. [Online]. Available: https://arxiv.org/abs/2501.01664v1

[148] H. Zhang, A. B. Sediq, A. Afana, and M. Erol-Kantarci, “Generative ai-in-the-loop: Integrating llms and gpts into the next generation networks,” *arXiv:2406.04276*, 2024. [Online]. Available: https://arxiv.org/abs/2406.04276

[149] B. Düzgün, A. Çayır, U. Ünal, and H. Dağ, “Network intrusion detection system by learning jointly from tabular and text-based features,” *Expert Systems*, vol. 41, no. 4, p. 13518, 2024. [Online]. Available: https://doi.org/10.1111/exsy.13518

[150] L. Gutiérrez-Galeano, J.-J. Domínguez-Jiménez, J. Schäfer, and I. Medina-Bulo, “Llm-based cyberattack detection using network flow statistics,” *Appl. Sci.*, 2025. [Online]. Available: https://doi.org/10.3390/app15126529

[151] P. R. B. Houssel, S. Layeghy, P. Singh, and M. Portmann, “ex-nids: A framework for explainable network intrusion detection leveraging large language models,” *arXiv:2507.16241*, 2025. [Online]. Available: https://arxiv.org/abs/2507.16241

[152] M. Heim, N. Starckjohann, and M. Torgersen, “(bachelor’s thesis) the convergence of ai and cybersecurity: An examination of chatgpt’s role in penetration testing and its ethical and legal implications,” Ph.D. dissertation, Norwegian Univ. Sci. Technol., 2025, accessed on 2025-10-19. [Online]. Available: https://hdl.handle.net/11250/3076387

[153] P. R. B. Houssel, P. Singh, S. Layeghy, and M. Portmann, “Towards explainable network intrusion detection using large language models,” *arXiv:2408.04342*, 2024. [Online]. Available: https://arxiv.org/abs/2408.04342

[154] N. O. Jaffal, M. Alkhanafseh, and D. Mohaisen, “Large language models in cybersecurity: A survey of applications, vulnerabilities, and defense techniques,” *AI*, vol. 6, no. 9, p. 216, 2025. [Online]. Available: https://doi.org/10.3390/ai6090216

[155] H. Ma, W. Zhang, D. Zhang, and B. Chen, “An iot intrusion detection framework based on feature selection and large language models fine-tuning,” *Sci Rep.*, vol. 15, no. 1, p. 21158, 2025. [Online]. Available: https://doi.org/10.1038/s41598-025-08905-3

[156] J. Chandra and P. Manhas, “Adversarial robustness in optimized llms: Defending against attacks,” *EasyChair Preprint*, no. 15857, 2025, accessed on 2025-10-20. [Online]. Available: https://easychair.org/publications/preprint/9r2v

[157] X. Xu, K. Kong, N. Liu, L. Cui, D. Wang, J. Zhang, and M. Kankanhalli, “An llm can fool itself: A prompt-based adversarial attack,” in *Proc.International Conference on Representation Learning (ICLR)*, 2024, pp. 1–23, accessed on 2025-10-20. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2024/file/0c72285e193ec90dca93258128698cfb-Paper-Conference.pdf

[158] X. Sophie, S. Alessandro, G. Stephan, G. Gauthier, and S. Leo, “Efficient adversarial training in llms with continuous attacks,” *arXiv:2405.15589*, 2024. [Online]. Available: https://arxiv.org/abs/2405.15589

[159] H. F. Atlam, “Llms in cyber security: Bridging practice and education,” *Big Data Cogn. Comput.*, vol. 9, no. 7, p. 184, 2025. [Online]. Available: https://doi.org/10.3390/bdcc9070184

[160] S. Yang, X. Zheng, X. Zhang, J. Xu, J. Li, D. Xie, W. Long, , and E. C. H. Ngai, “Large language models for network intrusion detection systems: Foundations, implementations, and future directions,” *arXiv:2507.04752v1*, 2025. [Online]. Available: https://arxiv.org/abs/2507.04752v1

[161] G. Claudia and I. Michele, “A formal framework for llm-assisted automated generation of zeek signatures from binary artifacts,” *Future Generation Computer Systems*, vol. 175, no. 108086, 2025. [Online]. Available: https://doi.org/10.1016/j.future.2025.108086

[162] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, “A survey on large language model (llm) security and privacy: The good, the bad, and the ugly,” *High-Confidence Computing*, vol. 4, no. 2, p. 100211, 2024. [Online]. Available: https://doi.org/10.1016/j.hcc.2024.100211

[163] A. Vassilev, A. Oprea, A. Fordyce, H. Anderson, X. Davies, and M. Hamin, “Adversarial machine learning: A taxonomy and terminology of attacks and mitigations,” *NIST Trustworthy*

*and Responsible AI*, 2025. [Online]. Available: https://doi.org/10.6028/NIST.AI.100-2e202

[164] S. Temara, "Maximizing penetration testing success with effective reconnaissance techniques using chatgpt," *arXiv:2307.06391*, 2023. [Online]. Available: https://arxiv.org/abs/2307.06391

[165] I. Hasanov, S. Virtanen, A. Hakkala, and J. Isoaho, "Application of large language models in cybersecurity: A systematic literature review," *IEEE Access*, vol. 12, pp. 176 751–176 778, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3505983

[166] K. Hamza, "Transformers and large language models for efficient intrusion detection systems: A comprehensive survey," *Information Fusion*, vol. 124, p. 103347, 2025. [Online]. Available: https://doi.org/10.1016/j.inffus.2025.103347

[167] P. Charan, H. Chunduri, P. Anand, and S. Shukla, "From text to mitre techniques: Exploring the malicious use of large language models for generating cyber attack payloads," *arXiv:2305.15336*, 2023. [Online]. Available: https://arxiv.org/abs/2305.15336

[168] A. Happe, A. Kaplan, and J. Cito, "Evaluating llms for privilege-escalation scenarios," *arXiv:2310.11409*, 2023. [Online]. Available: https://arxiv.org/abs/2310.11409

[169] G. Deng, Y. Liu, V. Mayoral-Vilches, P. Liu, Y. Li, Y. Xu, T. Zhang, Y. Liu, M. Pinzger, and S. Rass, "Pentestgpt: An llm-empowered automatic penetration testing tool," *arXiv:2308.06782*, 2023. [Online]. Available: https://arxiv.org/abs/2308.06782

[170] A. Happe, A. Kaplan, and J. Cito, "Llms as hackers: Autonomous linux privilege escalation attacks," *arXiv:2310.11409*, 2023. [Online]. Available: https://arxiv.org/abs/2310.11409

[171] A. Happe and J. Cito, "Getting pwn'd by ai: Penetration testing with large language models," in *Proc. 31st ACM Joint Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2023, p. 2082–2086.

[172] T. Naito, R. Watanabe, and T. Mitsunaga, "Llm-based attack scenarios generator with it asset management and vulnerability information," in *Proc. the 6th International Conference on Signal Processing and Information Security (ICSPIS)*, 2023, pp. 99–103. [Online]. Available: https://doi.org/10.1109/ICSPIS60075.2023.10344019

[173] D. I. Lajos Muzsai and A. Lukács, "Hacksynth: Llm agent and evaluation framework for autonomous penetration testing," *arXiv:2412.01778*, 2024. [Online]. Available: https://arxiv.org/abs/2412.01778

[174] H. Kong, D. Hu, J. Ge, L. Li, T. Li, and B. Wu, "Vulnbot: Autonomous penetration testing for a multi-agent collaborative framework," *arXiv:2501.13411*, 2025. [Online]. Available: https://arxiv.org/abs/2501.13411

[175] K. Nakano, R. Feyyazi, S. J. Yang, and M. Zuzak, "Guided reasoning in llm-driven penetration testing using structured attack trees," *arXiv:2509.07939*, 2025. [Online]. Available: https://arxiv.org/abs/2509.07939

[176] P. D. Luong, L. T. G. Bao, N. V. K. Tam, D. H. N. Khoa, N. H. Quyen, V. H. Pham, and P. T. Duy, "xoffense: An ai-driven autonomous penetration testing framework with offensive knowledge-enhanced llms and multi agent systems," *arXiv:2509.13021*, 2025. [Online]. Available: https://arxiv.org/abs/2509.13021

[177] X. Wu, Y. Tian, Y. Chen, P. Ye, X. Cui, J. Jia, S. Li, J. Liu, and W. Niu, "Curriculumpt: Llm-based multi-agent autonomous penetration testing with curriculum-guided task scheduling," *Appl. Sci.*, vol. 15, no. 16, 2025. [Online]. Available: https://doi.org/10.3390/app15169096

[178] S. Nakatani, "Rapidpen: Fully automated ip-to-shell penetration testing with llm-based agents," *arXiv:2502.16730*, 2025. [Online]. Available: https://arxiv.org/abs/2502.16730

[179] F. Teichmann, "Ransomware attacks in the context of generative artificial intelligence—an experimental study," *Int. Cybersecur. Law Rev.*, vol. 4, p. 399–414, 2023. [Online]. Available: https://doi.org/10.1365/s43439-023-00094-x

[180] M. Schmitt and I. Flechais, "Digital deception: Generative artificial intelligence in social engineering and phishing," *arXiv:2310.13715*, 2023. [Online]. Available: https://arxiv.org/abs/2310.13715

[181] P. V. Falade, "Decoding the threat landscape: Chatgpt, fraudgpt, and wormgpt in social engineering attacks," *Int. J. Sci. Res. Comput. Sci., Eng. Inf. Technol.*, vol. 9, pp. 185–198, 2023. [Online]. Available: https://doi.org/10.32628/CSEIT2390533

[182] J. Xu, J. W. Stokes, G. McDonald, X. Bai, D. Marshall, S. Wang, A. Swaminathan, and Z. Li, "Autoattacker: A large language model guided system to implement automatic cyber-attacks," *arXiv:2403.01038*, 2024. [Online]. Available: https://arxiv.org/abs/2403.01038

[183] S. Ćirković, V. Mladenović, S. Tomić, D. Drljača, and O. Ristić, "Utilizing fine-tuning of large language models for generating synthetic payloads: Enhancing web application cybersecurity through innovative penetration testing techniques," *Computers, Materials and Continua*, vol. 82, no. 3, pp. 4409–4430, 2025, accessed: 2025-10-19. [Online]. Available: https://doi.org/10.32604/cmc.2025.059696

[184] S. K. Shandilya, G. Prharsha, A. Datta, G. Choudhary, H. Park, and I. You, "Gpt based malware: Unveiling vulnerabilities and creating a way forward in digital space," in *Proc. International Conference on Data Security and Privacy Protection (DSPP)*, 2023, pp. 164–173. [Online]. Available: https://doi.org/10.1109/DSPP58763.2023.10404552

[185] M. Beckerich, L. Plein, and S. Coronado, "Ratgpt: Turning online llms into proxies for malware attacks," *arXiv:2308.09183*, 2023. [Online]. Available: https://arxiv.org/abs/2308.09183

[186] M. Botacin, "Gpthreats-3: Is automatic malware generation a threat?" in *Proc. IEEE Security and Privacy Workshops (SPW)*, 2023, pp. 238–254. [Online]. Available: https://doi.org/10.1109/SPW59333.2023.00027

[187] F. McKee and D. Noever, "The evolving landscape of cybersecurity:red teams, large language models, and the emergence of new ai attack surfaces," *Int. J. Cryptography Inf. Secur. (IJCIS)*, vol. 13, no. 1, p. 1–34, 2023. [Online]. Available: https://doi.org/10.5121/ijcis.2023.13101

[188] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, "From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy," *IEEE Access*, vol. 11, pp. 80 218–80 245, 2023. [Online]. Available: https://doi.org/10.1109/ACCESS.2023.3300381

[189] Y. M. P. Pa, S. Tanizaki, T. Kou, M. van Eeten, K. Yoshioka, and T. Matsumoto, "An attacker's dream? exploring the capabilities of chatgpt for developing malware," in *Proc. 16th Cybersecur. Express Test Workshop (CSET)*, 2023, p. 10–18. [Online]. Available: https://doi.org/10.1145/3607505.3607513

[190] J. Seymour and P. Tully, "Generative models for spear phishing posts on social media," *arXiv:1802.05196*, 2018. [Online]. Available: https://arxiv.org/abs/1802.05196

[191] J. Hazell, “Spear phishing with large language models,” *arXiv:2305.06972*, 2023. [Online]. Available: https://arxiv.org/abs/2305.06972

[192] K. Renaud, M. Warkentin, and G. Westerman, “From chatgpt to hackgpt: Meeting the cybersecurity threat of generative ai,” *MIT Sloan Manage. Rev.*, vol. 64, no. 3, pp. 1–4, 2023, accessed on 2025-10-19. [Online]. Available: https://sloanreview.mit.edu/article/from-chatgpt-to-hackgpt-meeting-the-cybersecurity-threat-of-generative-ai/

[193] M. Bethany, A. Galiopoulos, E. Bethany, M. B. Karkevandi, N. Vishwamitra, and P. Najafirad, “Large language model lateral spear phishing: A comparative study in large-scale organizational settings,” *arXiv:2401.09727*, 2024. [Online]. Available: https://arxiv.org/abs/2401.09727

[194] M. Sharma, K. Singh, P. Aggarwal, and V. Dutt, “How well does gpt phish people? an investigation involving cognitive biases and feedback,” in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroSandPW)*, 2023, p. 451–457. [Online]. Available: https://doi.org/10.1109/EuroSPW59978.2023.00055

[195] B. Singer, K. Lucas, L. Adiga, M. Jain, L. Bauer, and V. Sekar, “On the feasibility of using llms to autonomously execute multi-host network attacks,” *arXiv:2501.16466*, 2025. [Online]. Available: https://arxiv.org/abs/2501.16466

[196] M. Xu, J. Fan, X. Huang, C. Zhou, J. Kang, D. Niyato, S. Mao, Z. Han, X. Shen, and K. Y. Lam, “Forewarned is forearmed: A survey on large language model-based agents in autonomous cyberattacks,” *arXiv:2505.12786*, 2025. [Online]. Available: https://arxiv.org/abs/2505.12786

[197] P. Sun, X. Yun, S. Li, T. Yin, and C. S. andJiang Xie, “Advtg: An adversarial traffic generation framework to deceive dl-based malicious traffic detection models,” in *Proc. the ACM on Web Conference (WWW’25)*, 2025, pp. 3147–3159. [Online]. Available: https://doi.org/10.1145/3696410.3714876

[198] R. Sheatsley, N. Papernot, M. J. Weisman, G. Verma, and P. McDaniel, “Adversarial examples for network intrusion detection systems,” *Journal of Computer Security*, vol. 30, no. 6, pp. 1–26, 2022.

[199] M. elShehaby and A. Matrawy, “Adversarial evasion attacks practicality in networks: Testing the impact of dynamic learning,” *arXiv:2306.05494v3*, 2025. [Online]. Available: https://arxiv.org/abs/2306.05494v3

[200] S. Liu, D. Cao, J. Kim, T. Abraham, P. Montague, S. Camtepe, J. Zhang, and Y. Xiang, “Eatvul: Chatgpt-based evasion attack against software vulnerability detection,” in *Proc. the 33rd USENIX Security Symposium*, 2024, pp. 7357–7374, accessed: 2025-10-19. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/liu-shigang

[201] H. Liu, J. Xue, S. Zhao, Y. Liu, and Z. Lu, “The dual role of large language models in network security: Survey and research trends,” in *Proc. the ACM Workshop on Wireless Security and Machine Learning*, 2025, pp. 20–25. [Online]. Available: https://doi.org/10.1145/3733965.3733972

[202] J. Jiao, L. Jiang, Q. Zhou, and R. Wen, “Evaluating large language model application impacts on evasive spectre attack detection,” *Electronics*, no. 7, p. 1384, 2025. [Online]. Available: https://doi.org/10.3390/electronics14071384

[203] Blog, “Stopping ai-powered threats: Palo alto networks detects llm-generated attacks in real-time,” 2024, accessed on 2025-10-25. [Online]. Available: https://live.paloaltonetworks.com/t5/community-blogs/stopping-ai-powered-threats-palo-alto-networks-detects-llm/ba-p/1224047

[204] M. Lupinacci, F. A. Pironti, F. Blefari, F. Romeo, L. Arena, and A. Furfaro, “The dark side of llms: Agent-based attacks for complete computer takeover,” *arXiv:2507.06850*, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.06850