Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 17, No. 3, 2026

Artificial Intelligence-Based Web Attack Detection: A Survey of Machine Learning, Deep Learning, Transformer and Large Language Model Approaches

Haiyang Wang, Yuejin Zhang *

Beijing Union University, Beijing, China
*** Corresponding author:** Yuejin Zhang

Abstract: Web services are now pervasive in almost all industries, but this has made them a major target for hackers. Traditional signature-based solutions are no longer effective against zero-day and polymorphic threats that evolve rapidly. The much heralded game-changer is artificial intelligence (AI) particularly deep learning and more recently large language models. But the literature is fragmented: CNNs learn local features, Transformers capture long-range dependencies, and LLMs offer semantic understanding - yet each suffers from real-world limitations on interpretability, efficiency and security. In this survey we chart this progress and suggest that it's not model accuracy but trustworthiness that matters most. We carve up benchmark datasets (CSIC2010, CICIDS2017) and see a disconnect between lab and reality. Our message: the next big jump will not be driven by larger parameters, but smaller, more visible and dynamic systems that can be trusted by security analysts.

Keywords: Web Attack Detection; Artificial Intelligence; Deep Learning; Transformer; Large Language Model; Cybersecurity.

1. Introduction

Finance, healthcare, and government now run on Web applications—and attackers know it. SQL injection, cross-site scripting, and their variants are not new, but the defensive arms race has reached a tipping point. Traditional Web Application Firewalls that rely on fixed signatures are perpetually one step behind: they catch what they have seen before, but fail against novel exploits that slip through unanticipated gaps. That fundamental mismatch has pushed the community toward smarter, data-driven solutions [1,2,11].

Early AI attempts looked promising. Classic machine learning—SVMs, random forests—could learn attack indicators from traffic data. In practice, however, their success hinged on hand-crafted features, a tedious and brittle process. Domain experts had to guess which URL lengths, character frequencies, or parameter structures mattered; and when traffic patterns shifted, those features lost relevance. Deep learning entered with great fanfare precisely because it promised to automate that feature engineering. And to some extent it delivered: CNNs proved adept at spotting malicious keyword clusters; RNNs and LSTMs modeled sequential HTTP patterns; hybrid architectures pushed accuracy further[1,2,11].

Still, these gains came with strings attached. Deep models are computationally hungry, they demand massive labeled datasets, and they operate as black boxes—three obstacles that cripple real-time deployment in security operations centers. Meanwhile, the Transformer architecture, borrowed from NLP, offered a new twist: self-attention could relate distant parts of an HTTP request, overcoming the locality bias of CNNs and the sequential bottleneck of RNNs. Several studies [8,9,13] confirmed their edge in intrusion and traffic classification. Yet the parameter explosion and training data requirements persisted[3,5,11].

More recently, large language models have entered the arena. LLMs bring pre-trained world knowledge and few-shot reasoning, which could be a game-changer for detecting attacks that lack labeled samples. Early work [6,10,16] shows LLMs can interpret suspicious behavior in context, not just match signatures. But they are also colossal, slow, and prone to hallucination—hardly ideal for high-volume Web traffic.

In this disjointed terrain, we identified a need for a cross-methodological systematic review. This paper does not merely describe techniques; we evaluate them, point out their limitations and identify what is important to practitioners. Our work offers: (1) a taxonomy of ML, DL, Transformer, and LLM methods; (2) a comparative analysis of typical models (CNN, RNN, LSTM, GNN, Transformer, LLM) and their trade-offs; (3) a summary of datasets, metrics, and challenges; and (4) a look into the future at lightweight design, explainability, and adaptive defense. The remainder of the paper is laid out as follows. Section 2 discusses ML and DL. Section 3 is about Transformers. Section 4 discusses LLMs. Section 5 covers data sets, metrics and issues. Section 6 ends with future directions.

2. Machine Learning and Deep Learning-Based Web Attack Detection

2.1. Machine Learning-Based Web Attack Detection

Before deep learning took over, ML models were the workhorse of Web attack detection [1,2,11]. Unlike static signatures, they could infer attack patterns from statistical properties of HTTP requests—length, character distribution, URL structure, parameter values. With feature vectors in hand, classifiers like SVM, decision trees, and random forests separated normal from malicious traffic.

For SQLi and XSS, this worked reasonably well: attack payloads often contain telltale strings, odd symbols, or JavaScript snippets that ML algorithms can latch onto. But the catch was feature engineering. Every new attack type required experts to redefine what to look for, making the system slow to adapt. Moreover, these shallow models cannot grasp

semantics—the same token might be benign in one context and dangerous in another. And labeled data, as always, is scarce and expensive. So while ML laid the groundwork, its ceiling is low against today's polymorphic and context-aware threats[1,2,11].

2.2. Deep Learning-Based Web Attack Detection

Deep learning promised to lift that ceiling by learning features end-to-end. Rather than hand-tuning, these models ingest raw HTTP payloads and build hierarchical representations [3-5]. We discuss three major families[3,4,5].

2.2.1. Convolutional Neural Network-Based Methods

CNNs excel at detecting local motifs. A malicious SQL payload, for example, might harbor "UNION", "SELECT", or "WAITFOR" interleaved with anomalous punctuation—CNNs discover such combinations via sliding filters. Their strength is speed and pattern matching. But the same localized receptive field becomes a weakness when an attacker scatters the exploit across the URL, headers, and body. The CNN cannot easily reassemble those distant clues into a unified threat. Hence researchers looked elsewhere[3,4].

2.2.2. Recurrent Neural Network and LSTM-Based Methods

HTTP requests are inherently sequential. RNNs process tokens one by one, maintaining a hidden state that carries context forward. For short requests this works fine, but long payloads cause vanishing gradients. LSTMs introduced gated memory cells to retain information over longer spans. Consider a request with multiple parameters: each alone looks innocent, but their combination—say, a peculiar header plus an odd query string—may signal an attack. LSTMs can capture such cross-parameter dependencies. The downside is sequential training, which is slow and does not parallelise well. Even with gating, extremely long sequences remain challenging, and the computational cost at scale is prohibitive for many security teams[4,5].

2.2.3. Hybrid Deep Learning Models

To get the best of both worlds, hybrids cascade CNN for local features and LSTM/BiLSTM for sequential patterns (see Figure 1). This two-stage pipeline picks up both local signatures and distributed anomalies. However, stacking modules multiplies parameters, raising training time and inference overhead. Interpretability suffers even more, as it is harder to tell which part of the network flagged a request[3,4,5].

The hybrid architecture, as illustrated in Figure 1, consists of cascading a CNN module to extract local features with an LSTM/BiLSTM module to model sequential dependencies and then inputting the resulting representations into a classifier. This two-step design identifies both localized attack patterns (such as malicious keyword clusters) and distributed patterns across various components of an HTTP request. Nonetheless, computational costs and training overheads are increased by the parameter growth in such ensembles, and the lack of interpretability of these stacked architectures is a practical issue to security analysts.

3. Transformer-Based Web Attack Detection

3.1. Introduction of Transformer Architecture

A typical HTTP request is not a flat sequence; it is a structured composite where meaning can be spread across headers, cookies, and body. A malicious instruction might appear piecewise—part in the URL, part in a cookie—with no local pattern betraying it. Transformers sidestep this by using self-attention to compare every pair of positions simultaneously. Originally proposed for machine translation [13], this design gives the model a global view without the recurrent bottleneck. Training is also parallelizable, which speeds up GPU usage [8,9,13].

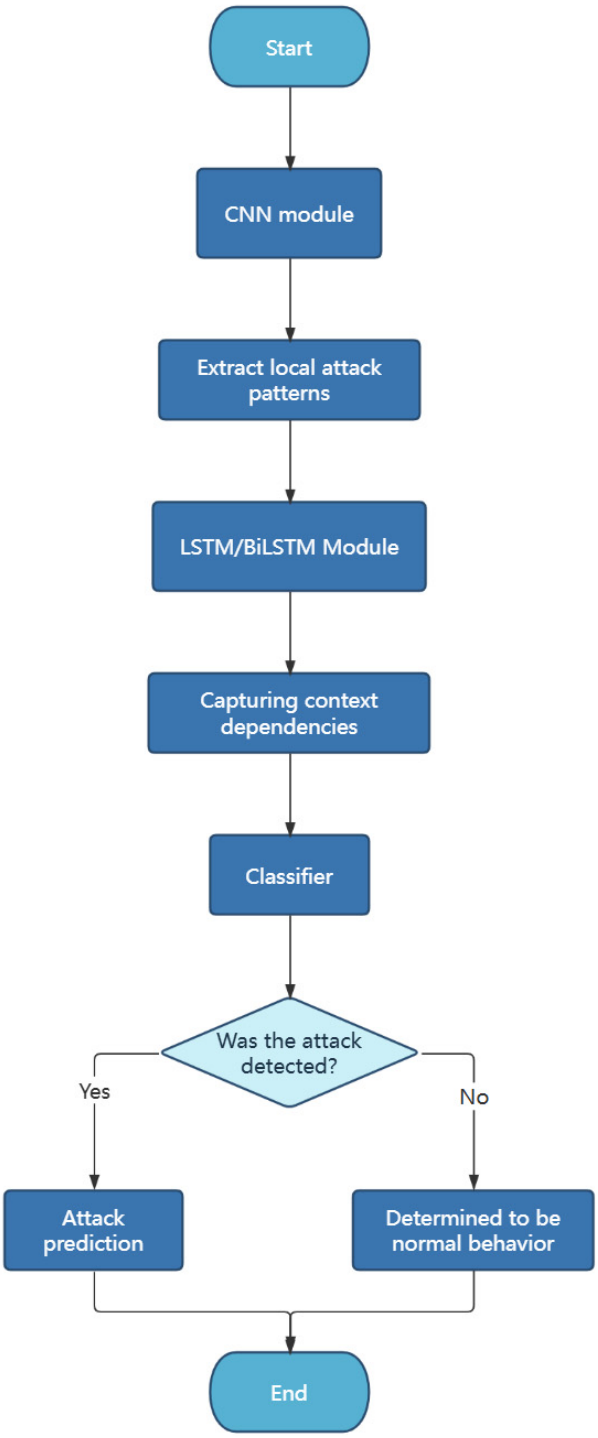


Figure 1. Framework of hybrid deep learning model for Web attack detection

3.2. Transformer-Based Intrusion and Web Attack Detection

Several works have adapted Transformers for network security. The RTIDS system [8], for instance, employs

attention to weigh interactions among traffic features, achieving strong benchmark results. Compared to CNNs and LSTMs, Transformers offer: (i) richer representation learning with minimal feature engineering; (ii) direct modelling of long-range dependencies; (iii) efficient parallel training. But the parameter count is high, and they require even more data—exactly what security datasets lack. Class imbalance and scarce attack samples further hamper training [8,9].

3.3. Graph-Based Attention

Beyond request-level analysis, some studies model user behaviour as graphs. PWAGAT [7] builds a graph of user activities (page visits, timings, interaction patterns) and applies graph attention to weight neighbour importance. This behavioural perspective can uncover stealthy attack paths that individual requests miss. Yet graph construction is non-trivial: it demands domain expertise and sufficient activity logs, and there is no standard recipe.

3.4. Strengths and Weaknesses Summarised

According to the available literature, Transformer-based Web attack detection algorithms offer substantial benefits over conventional machine learning and traditional deep learning models[8,9,13].

The benefits are as follows:

(1) Good semantic knowledge ability.

The contextual relationship in the HTTP requests can be analyzed using transformer models, which is advantageous to detect complex malicious payloads.

(2) Good feature fusion.

Attention mechanisms enable models to integrate various forms of security features.

(3) Flexibility to various security jobs.

Transformer architectures can be used to:

Detection of web attacks;
intrusion detection;
malicious traffic classification;
DDoS detection.

Nevertheless, despite their benefits, there are a number of problems:

(1) Complexity of the model.

Transformer models of large sizes demand significant computing power.

(2) Ease of deployment.

Web security systems that operate in real-time tend to demand low latency, and Transformer inference prices are expensive.

(3) Reliance on labeled datasets.

Security datasets are often lacking in samples of new attacks, restricting the effectiveness of model training[1,2,11].

(4) Problems with interpretability.

Despite the fact that attention weights give a certain amount of information, it is difficult to understand the final detection decision.

Although these developments are promising, there are still major barriers to using Transformer architectures for Web attack detection. Their performance, due in part to the self-attention mechanism, is superior but at the expense of significant computational costs - a trade-off that makes real-time deployment in production Web services challenging. Furthermore, the success of these models is highly dependent on access to large-scale annotated training data, which is often not available in real-world security scenarios where attack samples are limited and ever-changing. Even the graph-based extensions, despite providing new insights into the behavior, bring their own challenges in graph construction and aggregation, which have no clear solutions. These challenges - ranging from efficiency, data dependency to deployability - suggest the need for more sophisticated paradigms that can go beyond the constraints of existing architectures.

Large language models have started to become an object of significant research interest in this context as a possible next step. In contrast to traditional Transformer models, which are trained on small security datasets, LLMs introduce pre-trained semantic knowledge and few-shot reasoning skills that can potentially solve some of the most stubborn issues mentioned above, especially when it comes to detecting unknown attacks. Recent attempts in this new direction are reviewed in the next section.

4. Large Language Models for Web Attack Detection

4.1. Large Language Models Introduction in Cybersecurity.

Despite the fact that traditional deep learning and transformer-based methods have greatly enhanced the performance of Web attack detection, the majority of current methods are still based on predefined attack types and a large amount of labeled training data. In real-world Web security scenarios, attackers constantly evolve their strategies, create polymorphic payloads, and exploit new vulnerabilities, making it difficult to detect them using existing detection techniques.

Large language models (LLMs) are a new wave of artificial intelligence technologies that possess a high level of natural language understanding, reasoning ability, and knowledge transfer capability. In contrast to traditional deep learning systems that typically require training on particular tasks, LLMs can execute multiple security-related tasks by using instruction-based interaction and contextual learning[14,15,16,17].

As Web requests, attack payloads, system logs and security reports typically include complex textual data, LLMs offer new opportunities to intelligent Web attack detection. Using large-scale pre-training knowledge, LLMs can examine semantic relationships in security data, comprehend the intentions of attacks, and help security systems detect threats that have never been seen before [6,10].

LLM-based solutions have a number of possible benefits compared to traditional machine learning and deep learning solutions.

4.2. Web Attack Classification using LLM

Recently, the use of LLMs in detecting Web attacks has become a topic of interest. A representative study, LLM-AE-MP: Web Attack Detection Using a Large Language Model with Autoencoder and Multilayer Perceptron, suggested a hybrid architecture that integrates large language models with conventional neural network architectures[6].

The framework incorporates three key elements, namely;

a semantic feature representation learning large language model (LLM), an autoencoder to compress and reconstruct features, and a multilayer perceptron (MLP) to classify attacks.

Figure 2 shows the general workflow of the LLLM-AE-MP framework.

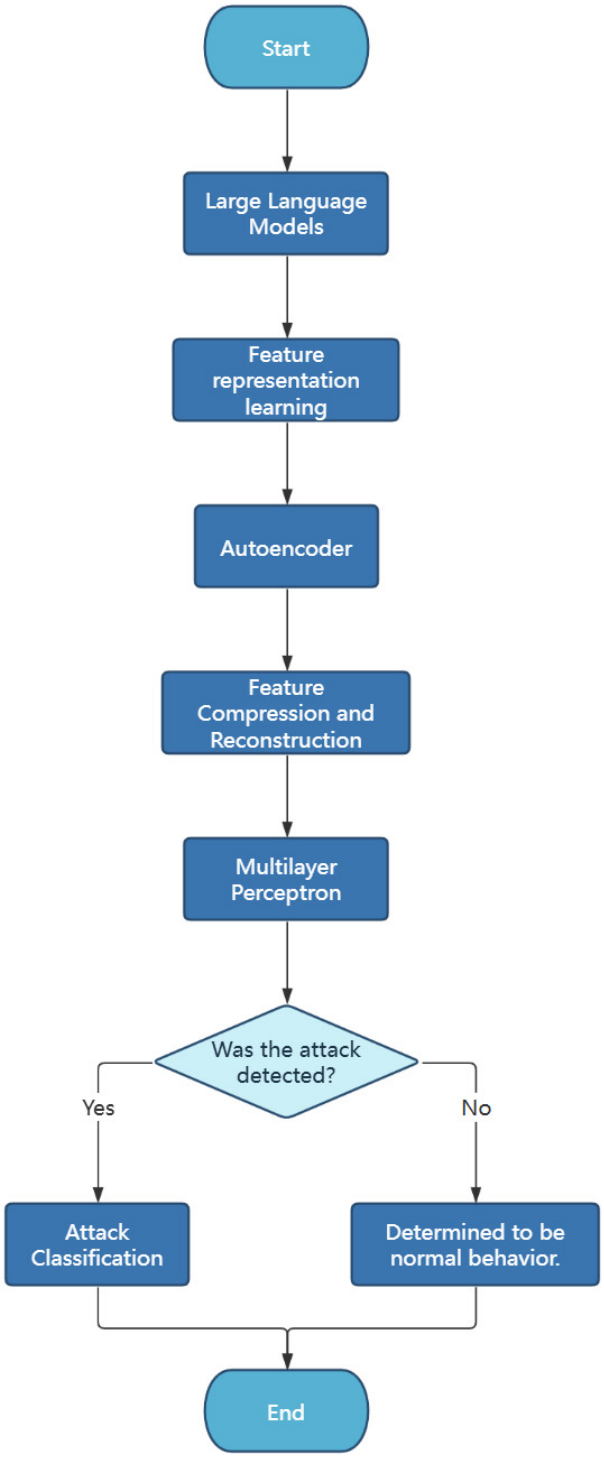


Figure 2. Framework of LLM-AE-MP for Web attack detection.

The LLM-AE-MP model has three steps: semantic feature extraction through an LLD, dimensionality reduction and reconstruction. via an autoencoder, and classification by an MLP. The intuition is simple- LLMs model the semantic richness of Web requests that keyword-based signatures fail, whereas the lightweight autoencoder-MLP stack maintains inference efficient. Experiments in 66 show that this hybrid method enhances detection of complex attacks whose malicious. features are spread over various request elements. The trade-off, but the cost of running an LLM at scale is computational,which is a major impediment to production deployment[6].

4.3. Unknown Web Attack Detection with the help of LLM.

Unknown attack detection is one of the most promising uses of LLMs in cybersecurity[10,16].

Conventional detection approaches typically train their models on attack signatures from historical data. But attackers constantly develop new methods, leading to zero-day attacks and novel attack patterns. As these attacks may not follow the same distribution or have the same features as existing signatures, traditional detection systems may suffer from performance degradation[1,10,11].

The paper Unknown Web Attack Threat Detection Based on Large Language Model suggested a smart system of identifying unknown threats of Web attacks through graph-based behavior modeling, contrastive learning, and analysis with the use of large language models[10].

The model suggested in 1010 takes a three-pronged approach. To begin with, it builds a Web behavior graph based on user actions: recording page accesses, temporal sequences, and possible attack paths instead of viewing each HTTP request as a single event. This graph structure maintains relational context which is lost in single-request analysis. Second, graph contrastive learning is used to learn stable behavioral representations of unlabeled data to minimize the reliance on manually labeled attack samples. Lastly, the features extracted about the graph are analyzed by an LLM to produce threat descriptions that can be read by humans and convert raw behavioral anomalies into actionable security intelligence. The design reflects a wider trend: LLMs are no longer regarded as classifiers but reasoning engines capable of explaining and interpreting complex patterns of attacks.

4.4. LLM-Based Web Attack Detection Challenges.

LLM-based detection systems have great potential, but they are challenged by significant challenges. The most obvious is the cost of computation- inference on billion-parameter models is orders of magnitude more costly than conventional ML methods, and real-time processing of high-volume Web traffic is not feasible without aggressive compression or distillation. The second issue is reliability: it is known that LLMs hallucinate, and in a security setting, a false explanation or a missed detection may be very serious. Another risk is adversarial prompts that are created to undermine LLM reasoning. Third, general-purpose LLMs do not have specialized security knowledge unless they are fine-tuned on domain data, but there are few curated security corpora to use when fine-tuning. To overcome these challenges, model compression, retrieval-augmented generation, and domain-specific pre-training will probably need to be combined[6,10,16].

5. Datasets, Evaluation Metrics and Challenges

5.1. Popular Datasets to Detect Web Attacks.

The quality and features of datasets are a major factor in the performance assessment of artificial intelligence-based Web attack detection techniques. In contrast to machine learning tasks in general, cybersecurity datasets must include various attack patterns, realistic traffic patterns, and enough samples of malicious activities. Nonetheless, gathering real-life Web attack information is still difficult because of privacy

issues, security constraints, and the ever-evolving nature of cyber threats[3,5,11].

Current research has used a number of public datasets to test Web attack detection models, such as Web application datasets and network intrusion datasets. These datasets offer various viewpoints to analyze malicious behaviors, including HTTP request-level analysis and network traffic-level detection.

There are three popular datasets in the literature. CSIC2010 is an older dataset, but still a popular choice for HTTP-level attacks since it includes labelled normal and malicious requests (SQL injection, buffer overflow, information leakage attacks, etc). Its primary limitation is that it is synthetically generated: the network traffic does not contain the noise and variability of real networks, and the attack payloads are from the past generation of Web attacks. CICIDS2017 has more realistic network flows and a wider range of attacks - DoS, DDoS, brute-force, botnet, and infiltration - making it a popular dataset for Transformer-based intrusion detection systems. But its class imbalance and lack of realism compared to real-world traffic means that it's still not fully representative of modern network traffic. UNSW-NB15 was specifically created to overcome the limitations of previous datasets by incorporating recent types of attacks such as exploits, fuzzing and reconnaissance, which has made it a popular choice for testing ML and DL approaches.

5.2. Evaluation Metrics

We measure the performance of detection by standard classification measures: Accuracy, Precision, Recall, F1-score and AUC-ROC. But since Web security data is highly imbalanced in terms of classes, Accuracy is not reliable and can be very misleading. The trade-off between Precision and Recall is more important to consider, as the former indicates the burden of false alarms on security analysts, whereas the latter indicates how well the system will detect real attacks. Practically, we give more weight to F1-score and AUC-ROC than raw Accuracy, which gives more meaningful evaluations in the case of imbalance[3,5,11].

5.3. Problems with AI-Based Web Attack Detection.

Despite the considerable enhancement of Web attack detection capability by artificial intelligence, there are still a number of challenges that have not been addressed. We classify them into three types: data-related constraints, deployment constraints and inherent security threats of AI models themselves[1,3,5,11].

6. Conclusion and Future Directions.

6.1. Future Directions

Despite the considerable advances in artificial intelligence-based Web attack detection strategies, current solutions have a number of limitations regarding flexibility, effectiveness, interpretability, and real-world implementation. According to the existing trends of machine learning development, deep learning, transformer models, and large language models, a few future research directions are discussed below.

6.1.1. Small and Efficient AI-based Detection Models.

As AI models become more complex, the cost of computation has emerged as a significant constraint to practical Web security applications[5,8,9].

The deep learning models in traditional form tend to have a high number of parameters, and transformer-based models and large language models demand even more computational power. Though these models are capable of attaining a high detection accuracy, it is still difficult to implement them in real-time Web protection systems.

Thus, the creation of light and effective detection models is a significant future trend.

Possible solutions are:

- ·model compression;
- ·knowledge distillation;
- ·optimization of parameters;
- ·lightweight transformer architectures.

The future Web attack detection systems must be more balanced between the performance of detection and computational efficiency, so that they can be implemented in resource-constrained settings.

6.1.2. Zero-Shot and Unknown Web Attack Detection.

Detection of unknown attacks is one of the most important issues in Web security[1,10].

The majority of the current machine learning and deep learning models are supervised learning algorithms which need adequate labeled attack samples. Nonetheless, attackers in practice keep on changing attack strategies, which leads to new patterns of attacks that might not be present in available data.

Recent research with large language models has shown great promise in detecting unknown attacks. Through semantic knowledge and transfer learning, LLM approaches can detect abnormal behaviors without prior knowledge of attack types.

Further studies should also investigate:

- ·zero-shot learning;
- ·few-shot learning;
- ·adaptive threat detection;
- ·lifelong learning processes.

These methods can enhance the responsiveness of Web security systems to threats.

6.1.3. Explainable Artificial Intelligence to Web Security.

Despite the fact that AI models have already made a significant contribution to detection performance, most deep learning and transformer-based models are still hard to interpret.

In real-world cybersecurity applications, security analysts require not just the correct detection outcome but also explanations about:

- ·why a request is considered malicious;
- ·what features are involved in the decision;
- ·what attack behavior is detected.

A possible solution is explainable artificial intelligence (XAI) that enhances the transparency of AI-based security systems.

Web attack detection models can be further combined with XAI techniques in future research to create more trustworthy and interpretable security solutions[12].

6.1.4. Combination of Large Language Models and Security Knowledge

The strong semantic reasoning capabilities of large language models open up new possibilities to intelligent cybersecurity applications[6,10,16].

Existing research has shown that LLMs can be used to help with:

Classification of web attacks;
unknown threat detection;

analysis of security behavior.

Nevertheless, general-purpose LLMs might not have specific cybersecurity expertise and can generate inaccurate results in multifaceted security situations.

The incorporation of LLMs with should be studied in the future:

security knowledge bases;
threat intelligence;
retrieval mechanisms;
security training, specific to domain.

This integration can enhance the accuracy, reliability and practical usefulness of Web attack detection systems based on LLM.

6.1.5. Real-Time and Adaptive Web Attack Defense Systems are real-time and adaptive systems that defend against web attacks.

The contemporary Web environments are very dynamic, and detection systems must be able to react promptly to the ever-evolving threats.

Intelligent Web security systems in the future must not merely identify attacks but also aid adaptive defense systems.

Potential research avenues are:

real-time traffic analysis;
automated response to threats;
ongoing model updating;
smart security agents.

The integration of AI detection with adaptive defense measures will enable future systems to offer a more proactive cybersecurity protection[10].

6.2. Conclusion

Web applications are now critical parts of contemporary digital infrastructures, yet their openness and complexity have also brought with it growing security threats. Rule-based and signature-based detection techniques are useful in detecting known attacks but are often ineffective when dealing with emerging threats and unknown attack patterns.

This article was a thorough review of artificial intelligence-based Web attack detection methods. The development of the conventional machine learning algorithms to deep learning, transformer architectures, and large language models was thoroughly examined.

Machine learning algorithms offer a starting point to the problem by training on attack-related patterns in security data, but they rely heavily on manually engineered features. Deep learning models also enhance detection capability by automatically learning sophisticated representations of Web requests and network traffic. Transformer-based approaches improve feature learning using self-attention mechanisms and show great promise in modeling complex dependencies. More recently, large language models have opened new avenues for detecting Web attacks by offering semantic knowledge, reasoning skills, and better adaptability to novel threats.

Although there has been a lot of progress, there are still some challenges to be addressed, such as the limitations of datasets, zero-day attacks detection, model complexity, interpretability, and deployment in the real world. The future studies ought to aim at creating lightweight, explainable, adaptive, and trustworthy AI-based Web attack detection systems.

As artificial intelligence technologies continue to develop, the combination of deep learning, transformer architectures, and large language models is likely to be a significant trend in developing intelligent Web security defense systems of the next generation.

References

[1] Guo, Y. (2023). A review of machine learning-based zero-day attack detection: Challenges and future directions. Computer Communications, 198, 175-185. https://doi.org/ 10.1016/j. comcom. 2022.11.001NIST.

[2] Yang, Y., Gu, Y. H., & Yan, Y. (2023). Machine learning-based intrusion detection for rare-class network attacks. Electronics, 12(18), 3911. https://doi.org/ 10.3390/ electronics 12183911.

[3] Alaoui, R. L., & Nfaoui, E. H. (2022). Deep learning for vulnerability and attack detection on web applications: A systematic literature review. Future Internet, 14(4), 118. https://doi. org/10.3390/fi14040118.

[4] Arroju, S., Adunoori, B., Jadhav, R., Jakkula, J., & Sreelakshmi, R. (2025). Detecting web attacks with end-to-end deep learning. Journal of Computer Allied Intelligence, 3(4), 36-46.

[5] Wu, Y. Q., Zou, B. L., & Cao, Y. F. (2024). Current status and challenges and future trends of deep learning-based intrusion detection models. Journal of Imaging, 10(10), 254. https:// doi. org/10.3390/jimaging10100254.

[6] Yang, J., Wu, Y. G., Yuan, Y. P., Xue, H. Z., Bourouis, S., Abdel-Salam, M., Prajapat, S., & Por, L. Y. (2025). LLM-AE-MP: Web attack detection using a large language model with autoencoder and multilayer perceptron. Expert Systems with Applications, 274, 126982. https://doi.org/10. 1016/j. eswa.2025.126982.

[7] Xu, Y. J., Fang, Y., Liu, Z. L., & Zhang, Q. (2023). PWAGAT: Potential web attacker detection based on graph attention network. Neurocomputing, 557, 126725. https://doi.org/10. 1016/j. neucom.2023.126725.

[8] Wu, Z. H., Zhang, H., Wang, P. H., & Sun, Z. B. (2022). RTIDS: A robust transformer-based approach for intrusion detection system. IEEE Access, 10, 64375-64387. https:// doi. org/ 10.1109/ACCESS.2022.3183012.

[9] Farias Junior, E. P., De Neira, A. B., Borges, L. F., & Nogueira, M. (2025). Transformers model for DDoS attack detection: A survey. Computer Networks, 270, 111433. https://doi.org/ 10. 1016/ j. comnet.2025.111433.

[10] Xu, Y. J., Zhang, Q., Deng, H. X., Liu, Z. L., Yang, C., & Fang, Y. (2025). Unknown web attack threat detection based on large language model. Applied Soft Computing, 173, 112905. https://doi.org/10.1016/j.asoc.2025.112905.

[11] Hozouri, A., Mirzaei, A., & Effatparvar, M. (2025). A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. Discover Artificial Intelligence, 5, 314. https://doi.org/10.1007/s44163-025-00314-8.

[12] Fasla, H., & Mehmood, A. (2026). Explainable artificial intelligence for web application threat detection: A systematic review. Artificial Intelligence Review. https://doi.org/10. 1007/ s10462-025-10987-9.

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998-6008). https://doi.org/10.48550/ arXiv. 1706. 03762.

[14] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT (pp. 4171-4186). https://doi.org/10.18653/v1/N19-1423.

[15] Liu, Y. H., Ott, M., Goyal, N., Du, J. F., Joshi, M., Chen, D. Q., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019).

RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692. https://doi.org/10. 48550/ arXiv. 1907.11692.

[16] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems (Vol. 33, pp. 1877-1901). https://doi.org/10.48550/arXiv.2005.14165.

[17] Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In Proceedings of EMNLP (pp. 3045-3059). https://doi.org/10. 18653/v1/ 2021. emnlp-main.240.

[18] Zhou, K. Y., Li, J. K., Lian, M., Wang, Z. X., Zhang, H. Y., Qiao, Y., & Li, W. (2022). Learning to prompt for vision-language models. International Journal of Computer Vision, 130, 2337-2348. https://doi.org/ 10.1007/ s11 26 3-022-01632-4.