Beyond Signatures and IP Blocking: A Critical Survey of WAF Evaluation under Source-Path Diversity, Payload Mutation, and Parser Discrepancy

HaoLan Su

School of Computer and Intelligent Education, Lingnan Normal University, Zhanjiang, China
Corresponding author: 1879207743@qq.com

Abstract Web application firewall research is often organized by the signal available to a decision or enforcement layer: request content, client identity, browser behavior, or traffic history. That organization is useful for isolated experiments, but it leaves open how a WAF should be evaluated when these pressures occur together. A malicious request may arrive through a changing or shared source path, use a different surface form, and pass through parsers that do not agree on its meaning. This survey reviews signature and rule filtering, transport and client fingerprinting, challenge-based enforcement, surface-form machine learning, intent-oriented language models, behavioral linkage, and protocol normalization. It separates direct WAF measurements from adjacent evidence on proxy identification, encrypted-traffic classification, and general intrusion detection, and treats parser discrepancy as a distinct failure mode rather than another payload mutation. The reviewed corpus shows clear one-axis evidence: payload mutation and parser-level differences can change WAF outcomes, while source-path studies show that identity and transport signals can rotate, overlap, or be manipulated. We did not identify a direct focal-WAF experiment that varies source-path conditions and payload realization together while verifying the intended application-level effect. The survey therefore proposes composition-oriented evaluation: vary source-path conditions, payload realization, protocol representation, and request history in a controlled factorial design, and report WAF action separately from application-level effect. This framing makes claims about "intelligent WAFs" testable rather than purely architectural.

Keywords Index Terms—WAF, source-path diversity, payload mutation, parser discrepancy, behavioral detection

I. INTRODUCTION

A. The question behind "intelligent" WAFs

WAF products are commonly deployed as bundles. Signature rules may be combined with threat intelligence, bot scoring, JavaScript challenges, sandboxing, log analytics, and security-operations integration. Such integration can be valuable in practice, but it does not by itself establish a different detection architecture. The relevant technical question is narrower: what evidence shows that WAF decisions remain useful when source-path diversity weakens the association between a request and a stable client, and when payload mutation changes request form without changing its security-relevant intent?

That question is increasingly important because two delivery pressures challenge the assumptions behind conventional filtering, while a related representation problem can defeat the decision after the request reaches the WAF. First, attack traffic can be distributed across residential, datacenter, VPN, mobile, or other shared-IP paths. Second, an attack request can be transformed through encoding, casing, token substitution, structural alteration, or semantics-preserving rewriting. These changes affect different parts of the WAF pipeline. Source-path diversity challenges identity and reputation features; payload mutation challenges lexical and syntactic features; parser discrepancies challenge the consistency of the representation on which a rule is evaluated.

The literature contains evidence for each pressure, although the source-path evidence is often adjacent rather than WAF-specific. It is much less clear how those findings should be combined at the same WAF decision

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

boundary. A detector shown to retain performance under source-path rotation is not automatically robust to paraphrased payloads. A semantic classifier that survives token substitution is not automatically informed by cross-request behavior. Treating these results as if they were interchangeable creates stronger conclusions than the experiments support.

A useful way to see the problem is to follow one request through a deployed stack. Fig. 1 places the observation boundary in the request path. The edge may terminate TLS, attach a forwarded address, decode a body, reject a malformed chunk, and then pass a reconstructed request to a reverse proxy. The WAF can inspect one representation while the origin framework parses another. Even when all components agree on the bytes, the detector may see one request at a time, whereas the operator would prefer to know whether ten apparently unrelated requests belong to the same automation campaign. The phrase "detection capability" therefore depends on where the observation is made and what state is retained.

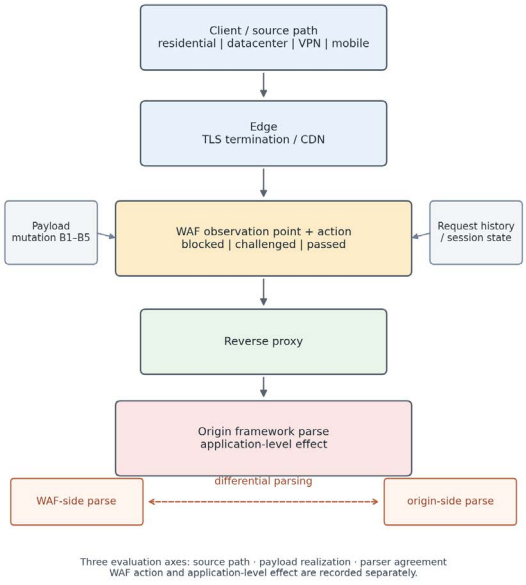


Fig. 1. Request path, WAF observation point, and the two representations involved in differential parsing.

The same issue appears in the attack literature from the opposite direction. WAF-A-MoLE does not need to change the underlying SQL-injection objective in order to alter what a classifier sees; its mutation operators work on the representation available to the detector. BWAFSQLi pushes this idea across a larger set of mutation strategies and configurations. The studies are not interchangeable, but together they show why "obfuscation" is too coarse a word for the payload side of the problem: case changes, comments, encodings, operator substitutions, and structural rewrites do not test the same invariance.

The source-path side is equally concrete. A residential proxy is not simply an IP address that changes. Provider pools can reuse addresses, carrier NAT can place unrelated users behind one prefix, and a CDN can replace the client's transport signature with an edge-side connection. An experiment that blocks a known hosting range may therefore be measuring infrastructure classification, while an experiment that uses a stable authenticated session is measuring a very different form of continuity. The survey keeps these cases separate instead of folding them into a single proxy/non-proxy label.

B. Scope and research questions

The survey focuses on detection at or around the WAF decision boundary for HTTP applications. It includes methods that inspect request content, client and transport identity, request sequences, or active client behavior. It also includes protocol-normalization work where the vulnerability lies in the disagreement between the WAF and the downstream application. General intrusion-detection and encrypted-traffic studies are used as adjacent evidence when they clarify a mechanism, an evaluation practice, or a transferable design principle; they are not presented as direct WAF results.

Four questions organize the review:

- RQ1. Which WAF detection approaches and supporting controls are represented in the literature, and what information does each use?
- RQ2. How do those approaches relate to source-path diversity, payload mutation, and protocol-level representation changes?
- RQ3. What limits the comparability of current evaluation results, especially with respect to task, source path, request context, and observability?
- RQ4. Which research directions would produce direct

evidence for combined source-path, payload, and protocol conditions rather than isolated-axis robustness?

C. Contributions

To answer RQ1–RQ4, this survey analyzes the literature along three linked dimensions. The first is the evidence source and operational role: request representation, client or transport identity, active interaction, cross-request behavior, and parser or normalization control. The second is the disturbance axis: source-path diversity, payload mutation, and protocol-level representation change. The third is the evaluation endpoint: WAF action, challenge state, parser agreement, and application-level effect, together with the status of each result as direct WAF, attack-surface, adjacent, or mechanism evidence. Following these dimensions through the request path makes it possible to ask not only what a method uses, but which signal remains available after the request path or payload surface changes.

This organization supports three contributions. First, it gives source-path diversity, payload mutation, and parser discrepancy a shared vocabulary while preserving their different failure mechanisms. Second, it provides an evidence audit that keeps task, observation point, and endpoint aligned when results are compared. Third, it turns the comparison into a composition-oriented research agenda: future WAF studies can vary the three axes together and report the decision path and application-level result separately. Most studies in the reviewed corpus examine one of these conditions at a time or group methods by algorithm family. The missing piece is therefore not another feature label, but a way to evaluate how the signals interact when source path, payload realization, and parsing are changed in the same experiment.

II. REVIEW SCOPE AND ANALYTICAL VOCABULARY

A. Review design

The reference set was assembled purposively to cover the detection and evasion mechanisms relevant to the three focal conditions. The database search was conducted in OpenAlex and Crossref on September 15, 2026, using a publication-date filter of 2019–2026. The reference set was closed on September 30, 2026; manually selected foundational studies, newer 2026 publications, and official product documentation were added separately.

Thirty-one English query strings covered seven topic groups: general WAF detection ("web application firewall detection taxonomy," "WAF mutation bypass survey," and "proxy pool web attack evasion"); source-path and proxy pools ("residential proxy web attack," "datacenter proxy bot detection," "VPN proxy WAF bypass," "mobile proxy web attack," and "residential proxy network measurement"); fingerprinting ("TLS fingerprint JA3 JA4," "HTTP/2 fingerprint bot detection," "client fingerprint web bot detection," and "transport layer fingerprint web security"); semantic and mutation detection ("WAF mutation testing," "web application firewall deep learning," "web attack detection neural network," "large language model web attack detection," "adversarial SQL injection benchmark," "WAF-aware mutation tool," and "SQL injection tamper encoding"); behavioral and session linkage ("behavior graph web intrusion detection," "request sequence anomaly detection," "web session intrusion detection," "privacy-preserving web intrusion detection," "cross-IP session TLS fingerprint," and "inter-arrival time anomaly web traffic"); evaluation ("AIDA web intrusion detection benchmark," "HTTP dataset WAF evaluation," and "OWASP Top 10 web application security"); and vendor-related context ("Cloudflare rate limit bypass proxy rotation," "GreyNoise residential proxy network," and "Akamai bot manager adversarial traffic"). These terms identify candidate literature for purposive full-text review; they are not an exhaustive census.

Each source was read against four questions: what task was actually measured; what variables were changed; what WAF, decision mechanism, or enforcement configuration was observed; and how far the result can be transferred to the present survey question. Sources were then assigned one of four evidence roles:

1. Direct WAF evidence: the study evaluates a WAF or WAF-like detector on an attack or evasion condition relevant to the claim.
2. Direct attack-surface evidence: the study measures the operational structure of source-path or parser behavior that defines the threat model, without measuring WAF robustness itself.
3. Adjacent evidence: the study evaluates a related detection task, such as residential-proxy identification or general intrusion detection.
4. Mechanism or context evidence: the source documents a signal, deployment mechanism, dataset, or tool, but does not provide a robustness experiment for the focal claim.

This classification prevents a common survey error: using a mechanism description as though it were a performance measurement. For example, JA3 or JA4 shows that TLS ClientHello information can be summarized into a fingerprint [3, 19]; it does not, by itself, show that a WAF can identify a particular proxy family under adversarial adaptation.

A result is considered direct only when the measured endpoint is a WAF or WAF-like decision, a WAF bypass, or a WAF-specific hardening outcome. Proxy measurements, fingerprinting studies, and generic anomaly-detection papers are retained for mechanism comparison, but their metrics are not silently promoted to WAF metrics. This distinction is especially important for accuracy figures: a classifier that identifies shared IPs is not thereby a classifier that identifies malicious requests.

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

The unit of comparison is the evidence pathway and enforcement role, not the marketing name attached to a product. This wording is intentional: a challenge can change enforcement without classifying payload intent, and normalization can change the inspected representation without making the security decision. For each paper, the review records whether the method observes raw request text, a normalized parse, transport metadata, browser execution, a sequence of requests, or an aggregate over an IP prefix. It also records whether the adversary is allowed to adapt. A fixed mutation set and an adaptive search against a surrogate classifier are both useful, but they answer different questions: the first estimates transfer across a declared transformation set, while the second asks whether the detector can be searched against. WAF-A-MoLE's guided mutation tree and BWAFSQLi's gray-box and black-box configurations illustrate that distinction [9, 24].

B. Source-path conditions and proxy-related variation

The survey uses four broad source-path conditions as an analytical vocabulary rather than as mutually exclusive ground-truth labels:

- A1: residential proxy. Traffic exits through addresses associated with residential access networks or residential proxy providers. Gateway rotation and overlap between providers are central properties.
- A2: datacenter proxy. Traffic exits through hosting or cloud infrastructure, often with more stable infrastructure-level signals but greater concentration in known address ranges.
- A3: relayed or edge-terminated path. Traffic reaches the WAF through an intermediary that relays or terminates the client connection. VPN relays and CDN edges are distinct subtypes, not interchangeable mechanisms; they are grouped here only because each can hide, replace, or transform client-side identity information before the WAF observation point.
- A4: mobile or carrier-shared path. Traffic is observed through mobile networks or large-scale carrier NAT, where many users can share address space and source-IP continuity is weak.

The categories are deliberately approximate. Giotsas et al. [12] distinguish carrier-grade NAT, VPN/proxy, and non-shared addresses using aggregate features over IP prefixes, but their classes are parallel to, rather than identical with, A1–A4. Their results are useful for grounding the source-path diversity problem: RTT variability, user-agent diversity, TLS-signature diversity, source-port diversity, and destination-host diversity become more informative when aggregated at the /24 level. The study does not establish a WAF-specific detection rate for any one proxy family.

The A-axis therefore refers to variation in the path and identity context visible to the WAF or adjacent detector. A1–A4 are analytical conditions, not mutually exclusive classes or a universal taxonomy: a request may, for example, traverse a residential proxy and then a CDN edge. A system can experience source changes, shared addresses, connection termination, client-fingerprint homogenization, or multiple users behind one prefix. Throughout, source-path diversity is the umbrella term; proxy diversity refers only to variation involving proxy providers or relays, not every source-path condition.

C. Payload mutation as a surface-form dimension

The B-axis describes transformations that preserve, approximate, or re-express the attack intent while changing the request surface. In this paper, attack intent is the security-relevant operation being attempted; application-level effect is the observable downstream result used to verify that a generated request still realizes that operation.

- B1, encoding mutation: percent encoding, alternate character encodings, and related representation changes.
- B2, case mutation: changes in keyword or token casing.
- B3, structural mutation: insertion, deletion, reordering, or substitution of syntactic elements while preserving the attack condition.
- B4, encoding-stack mutation: combinations of encodings, nested transformations, or transformations applied across multiple request layers.
- B5, semantic-preserving paraphrase: a higher-level rewrite that retains the intended operation while changing lexical or syntactic realization.

B1 concerns a single encoding layer, whereas B4 concerns composed or cross-layer transformations; the distinction matters because a detector may canonicalize one representation but not the sequence in which several representations are decoded. B4 is therefore still a payload-side transformation, whereas parser discrepancy concerns disagreement about the resulting fields or boundaries.

WAF-A-MoLE [9] supplies the main direct evidence for the first three families through seven semantics-preserving mutation operators. SQLMap tamper tooling [21] provides an operational example of how SQL-injection requests can be transformed, although a tool implementation is not itself evidence that a transformation preserves application semantics. BWAFSQLi [24] broadens the evaluation to fifteen mutation strategies and eleven WAF configurations, reporting increases in false-negative rate of up to 93.39% for the evaluated gray-box deep-learning systems and up to 58.49% for the evaluated black-box products. These studies justify treating payload mutation as an independent pressure. They do not test proxy variation at the same time.

B5 is kept separate because semantic-preserving rewriting is a stronger claim than token-level obfuscation. A BERT-based WAF can report high accuracy on an “advanced” category [13] and still lack a demonstrated test against

paraphrases generated to preserve attack intent. The absence of such a test is an evaluation gap, not evidence that semantic detection is ineffective.

The distinction between B3 and B5 matters in practice. A structural mutation can preserve the same token-level operation while changing delimiters, nesting, or the placement of comments. A paraphrase changes the linguistic or symbolic realization at a higher level; the request may still encode the same predicate, comparison, or command, but a detector trained on recurring strings has less to match. SQLMap's tamper scripts illustrate the operational side of the former category [21]. They are useful test generators, not a guarantee of semantic equivalence. In a defensible benchmark, each generated request needs to be sent to a reference application or interpreter, and the benchmark should record whether the intended effect, error condition, or query result was preserved.

The practical implication is that mutation should be treated as a feedback process rather than a one-off list of string substitutions. A generator can search for low-confidence requests, the application can check whether the intended effect survives, and the WAF can be retested against the resulting family of inputs. WAF-A-MoLE and BWAFSQLi motivate this loop from the adversarial side [9, 24]; neither study, by itself, establishes that a repaired rule will generalize to unseen applications or benign traffic.

Three studies make the generation side less abstract. DaNuoYi is an automatic injection-testing tool built around multi-task translation and evolutionary search with a shared mating pool and attack-specific mutation operators. In its evaluation on three real-world open-source WAFs and six injection types, the best setting produced up to 3.8× more valid bypassing inputs than the single-task baseline and up to 5.78× more than the context-free-grammar baseline [26]. WAFBOOSTER closes the loop from the defender side: it trains a shadow model, generates mutated malicious payloads, checks their validity with a payload corrector, and uses substring scoring, clustering, and regular-expression matching to construct candidate signatures. Across eight real-world WAFs, its reported true-rejection rate for mutated malicious payloads increased from 21% to 96%; the paper also reports no false rejections in the evaluated set and a false-acceptance rate roughly one-third of that produced by the compared payload-generation methods [25]. BWAFSQLi provides the complementary bypass-side condition: it evaluates SQL-injection mutations across fifteen strategies and eleven WAF configurations, separating gray-box deep-learning systems from black-box products, and reports false-negative-rate increases of up to 93.39% and 58.49%, respectively [24]. These results make the conditions behind the headline numbers visible. But they still address payload-side evaluation; none varies proxy paths in the same experiment.

D. Protocol parsing as a neighboring attack surface

Protocol-level evasion changes how the request is parsed rather than rewriting the attack payload itself. In this survey, normalization means aligning the WAF-side and application-side interpretations of security-relevant request fields before the decision is enforced. The target is equivalent security-relevant parsing, not necessarily byte-for-byte or full parse-tree identity. Canonicalization means deterministic preprocessing of an already parsed field or representation, such as decoding percent escapes or standardizing case. Canonicalization can reduce representational variation, but it does not by itself guarantee parser agreement. Break the Wall from Bottom [23] reports 311 protocol-level evasion cases across 14 WAFs and 20 web frameworks. WAFFLED [1] reports 1,207 unique bypasses across five WAFs and seven frameworks, focusing on multipart, XML, and JSON parser discrepancies. In WAFFLED, the attack payload remains unchanged while content-type-specific structure is mutated.

These studies are central to the survey because they expose a different failure mode from B1–B5. A payload classifier cannot recover from a parser disagreement if it never receives the same representation that the application later interprets. Protocol normalization therefore belongs in the synthesis, but it should not be counted as ordinary payload mutation. Collapsing the two would make the apparent robustness of a detector depend on which component happened to parse the request first.

There is a practical reason to keep this dimension visible even when the paper is about detection approaches. Normalization is not an optional preprocessing detail if the security decision is made on a representation different from the one later executed. A WAF can have a highly capable classifier and still lose at the boundary where multipart delimiters, duplicate parameters, XML entities, JSON coercions, or encoded separators are interpreted differently. The relevant test is differential: capture the WAF-side parse, capture the framework-side parse, and compare the fields that reach authorization, query construction, or command dispatch. A bypass that survives this comparison is a parsing result first and a machine-learning result second.

III. DETECTION APPROACHES AND SUPPORTING CONTROLS IN THE LITERATURE

To answer RQ1, the families below are compared by operational role and evidence boundary rather than treated as mutually exclusive classifiers or as components located at one layer. Sections 3.1–3.7 describe detection evidence, decision mechanisms, and representation control; challenge-based enforcement is included because it changes the action path rather than classifying payload intent.

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

A. Signature and rule filtering

Signature and rule filtering remains the clearest baseline for WAF operation. It matches known patterns, protocol conditions, or rule combinations against a request representation. Its main advantages are interpretability, predictable execution cost, and the ability to encode well-understood attack classes. Its limitations are visible in both payload and protocol studies.

On the payload side, WAF-A-MoLE [9] and BWAFSQLi [24] show that evaluated WAF configurations can be bypassed by adversarial mutations. These studies establish mutation sensitivity for the configurations and products they test; they do not isolate a universal weakness of signature rules, and the black-box results should not be assigned to a hidden detector family. They are relevant here because rule-based and learned detectors both depend on the request representation they inspect, but the evidence should not be read as a controlled comparison of those families.

On the protocol side, the evidence is different rather than simply stronger. Break the Wall from Bottom [23] and WAFFLED [1] find cases in which the WAF and the downstream framework assign different meanings to structurally altered requests. WAFFLED's in-the-wild validation further suggests that multiple content types are accepted within the evaluated web workflows, giving parser discrepancies practical relevance. The attack payload often remains untouched in these cases. A rule engine that is well matched to the payload can still miss the request if its input representation differs from the application's representation.

These findings suggest two distinct engineering requirements: canonicalize parsed inputs where this reduces irrelevant variation, and align WAF-side and application-side parsing when they might interpret the same request differently. Canonicalization operates after a parser has produced fields; normalization addresses disagreement in that parsing process. The former acts on a representation; the latter addresses disagreement about how that representation is produced. Rules should then be tested against transformations that preserve the application-level effect. Rule coverage cannot substitute for representation consistency.

The newer generation-and-repair studies point to a more active maintenance model. DaNuoYi is primarily a test generator: it increases the supply of valid bypassing inputs by transferring information across injection types [26]. WAFBOOSTER is closer to a repair pipeline, because it mines stable substrings from bypassing payloads and turns them into regular-expression signatures [25]. That distinction matters. Finding a bypass, generating a signature for it, and showing that the signature does not reject benign traffic are three separate measurements; a paper should not collapse them into one “robustness” score.

B. Client identity and transport fingerprinting

Client-identity methods derive signals from the transport and protocol stack. JA3 and JA4 summarize TLS ClientHello characteristics [3, 19]; HTTP/2 fingerprinting uses frame order, SETTINGS parameters, flow-control behavior, and pseudo-header patterns [20]. Cloud delivery infrastructure can forward a client fingerprint to an origin [4], but a WAF located behind a TLS-terminating edge may otherwise observe the edge's connection rather than the client's.

Fingerprinting is useful when the client implementation, automation stack, or proxy path leaves a stable and discriminative trace. It is less informative when a proxy pool homogenizes the visible handshake, when a real browser is used, or when the edge terminates and reconstructs the connection. The source-path literature also warns against treating one timing or transport feature as a durable identity signal. In Beyond RTT [2], RTT-based recall falls from 99% to 8% under traffic scheduling, while CorrTransform retains an F1 score of 92.01% in the evaluated residential-proxy identification task. This is clear adjacent evidence for adversarially unstable fingerprints, not a direct WAF attack-detection result.

Client fingerprints also have a different relationship to payload mutation than content detectors. A changed SQL token does not necessarily change the TLS or HTTP/2 fingerprint. That independence can make identity signals a useful complementary feature, but it does not make them a detector of the mutated payload. A fingerprint may associate requests with a client implementation while remaining agnostic to whether the request body contains an attack.

Fingerprinting also illustrates why deployment position matters. JA3 or JA4 is meaningful at a TLS endpoint that sees the client handshake; it is not the same signal at an origin behind TLS termination. The AWS CloudFront documentation shows that a downstream service can receive a JA4-derived value, but that is an explicitly forwarded feature, not an automatic property of every WAF deployment [4]. HTTP/2 fingerprints add another layer of detail. The Akamai study describes SETTINGS order and values, WINDOW_UPDATE increments, PRIORITY behavior, and pseudo-header order over more than ten million connections [20]. These are implementation traces, not payload semantics.

This distinction is useful when considering rotating proxies. A proxy can change the IP while preserving an identifiable browser stack, or it can normalize the handshake while leaving the source address unstable. A source detector that uses only one of these traces will be brittle in a different way from a rule engine that uses only request text. The evidence supports feature diversity, but not an assumption that fingerprints are permanent identities; clients update, headless tools imitate browsers, and intermediaries rewrite the very fields used for classification.

C. Challenge-based enforcement and client execution

JavaScript challenges and related mechanisms move part of the decision from passive observation to active interaction. Cloudflare Turnstile [7] and Azure WAF JavaScript Challenge [16] illustrate this family. The detector can require browser-side computation, state continuity, or an interaction pattern that a simple HTTP client is less likely to reproduce.

A challenge addresses automation cost rather than payload meaning. It can be combined with rate, reputation, and content signals, and it may remain useful after an address change if the system retains enough state to decide when to re-challenge. Once a genuine browser has completed the challenge, however, the mechanism does not establish that the subsequent request is benign. No experiment was identified in the reviewed corpus that combines residential-proxy rotation, intent-preserving payload rewriting, and challenge outcomes. Challenge systems are therefore best treated as a complementary control, not as evidence that the WAF recognizes B5 paraphrases.

D. Surface-form machine learning

Here, surface-form ML/DL names detectors evaluated through lexical, token, sequence, or request-feature representations; it is an evidence distinction, not a claim that a particular model architecture cannot encode semantics. Such detectors learn from characters, tokens, sequences, or request features. Their attraction is practical: a model can capture combinations of comments, whitespace, case, encodings, and delimiters that would be cumbersome to enumerate as rules. The same flexibility creates a less visible dependency. A model may learn the collection process as much as the attack pattern, picking up header order, URL naming conventions, user-agent strings, or artifacts of a particular request generator.

Demetrio et al. [9] evaluate character-based and token-based classifiers, including recurrent, probabilistic, tree, and SVM configurations, together with SQLiGoT variants. Their evaluated models are bypassed by semantics-preserving transformations. BWAFSQLi extends the adversarial test to eleven WAF configurations and fifteen mutation strategies, reporting false-negative-rate increases as high as 93.39% for the evaluated gray-box systems and 58.49% for the evaluated black-box products [24]. Those figures should not be compressed into the claim that machine learning is generally weak. They show that measured robustness depends on the transformation family, the access model, and the particular configuration under test.

Generic IDS literature gives a reason to be cautious with otherwise impressive benchmark numbers. Vinayakumar et al. [22] report substantial variation across datasets for a deep-learning IDS, while Khraisat et al. [14] and Ruff et al. [18] discuss dataset shift, class imbalance, and the absence of one anomaly-detection method that dominates across settings. Choi et al. [6] and Deng and Hooi [10] provide time-series and graph-based examples, but they are adjacent mechanism evidence rather than WAF robustness studies. For WAF evaluation, the key question is whether the split separates campaigns and applications, or merely separates rows drawn from the same collection process.

A second distinction concerns the representation given to the model. Canonicalization can make a detector less sensitive to superficial encoding changes, but only if the canonical form preserves security-relevant distinctions. A model that receives normalized fields may be more stable against token variation while remaining exposed to a parser discrepancy upstream. Here, “normalized fields” means fields produced after WAF/application parsing has been aligned; applying a deterministic canonicalization alone is not sufficient. Surface robustness and parser consistency are related controls, not substitutes.

E. Intent-oriented language models

Intent-oriented detectors are a subset defined by their evaluation claim: they attempt to represent the security meaning of a request rather than its exact wording. A transformer or language-model architecture alone does not establish this property; evidence requires tests in which attack intent is preserved while surface realization changes. LLM-WAF [13] is a useful mechanism anchor: it reports 96.8% accuracy, an F1 score of 0.95, a 1.4% false-positive rate, latency below 19 ms, and 89.1% accuracy for an “advanced” category. These are meaningful results under the paper’s stated data and configuration. They do not establish robustness to B5 semantic-preserving paraphrase, a new proxy path, cross-request attribution, or a disagreement between WAF and origin parsers.

The distinction matters because a transformer embedding is not, by itself, evidence of semantic understanding. The model may capture intent, but it may also rely on recurring lexical context, endpoint names, or generator-specific templates. A convincing test would hold the application-level effect fixed while changing words, operators, nesting, and request serialization. The reference application or interpreter would then verify that the intended effect survives. Without that step, “intent-oriented” describes a design ambition more securely than an observed invariant.

There is also an enforcement-path question. Body size, batching, timeout policy, and the need to inspect more than one representation all affect whether an intent model can sit inline. A detector may have a strong static classification result and still require a different deployment mode: asynchronous scoring, selective escalation, or a challenge decision followed by deeper inspection. These are not objections to language models; they are part of the deployment property that a WAF paper needs to measure.

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

F. Behavioral and session-linkage detection

Behavioral methods aggregate evidence across time, requests, identities, or sessions. Useful features include rate, burstiness, destination diversity, inter-arrival patterns, request entropy, and relationships among entities. Graph neural networks and multivariate anomaly detectors provide mechanism-level examples [6, 10, 18], but their value for WAFs depends on how traffic can be linked without treating a shared address as one person.

Among the reviewed sources, Beyond RTT [2] offers the clearest adjacent measurement under source-path pressure. Its CorrTransform model retains an F1 score of 92.01% in the evaluated residential-proxy identification task under a scheduling attack that reduces RTT-based recall from 99% to 8%. The task is not WAF attack detection. Its relevance is narrower: relational features can remain informative when one timing feature is deliberately disturbed, which makes them a plausible input to a stateful WAF experiment.

The A Global Inference and Assessment of Large Shared IP Addresses study [12] makes a related observation at a different granularity. Its classifier benefits from /24-level aggregate features that include diversity in user agents, TLS signatures, source ports, destinations, and RTT. Prefix aggregation is not an identity proof, and it is not the same as a cross-request behavior graph. It does, however, challenge the assumption that one IP corresponds to one stable actor. For WAF design, that cuts both ways: aggregation may reveal coordinated activity, while it may also join unrelated users and create collateral blocking.

The privacy question is therefore part of the detector design. A short-lived keyed session identifier, an authenticated account, or a browser attestation may support linkage without retaining raw identifiers indefinitely. Each choice changes who can be joined, what can be evaded, and which benign populations may be associated by mistake. A behavioral WAF should report these assumptions beside recall and false positives; "session context" is not a free feature but a stateful product decision.

G. Protocol normalization and contextual enrichment

Protocol normalization is an upstream representation control for downstream decision mechanisms. HTTP-Normalizer in WAFFLED [1] is designed to validate and normalize HTTP requests against current RFC requirements before forwarding them, and the study reports that it normalized or blocked the bypasses identified in its evaluation. Break the Wall from Bottom [23] reaches a related conclusion through automated discovery of parser-level discrepancies across WAFs and frameworks. The relevant test is differential: compare the WAF-side parse with the framework-side parse across content types, duplicate fields, boundary conditions, and encoding stacks. A classifier cannot compensate for inspecting a representation that the application will not use.

Threat-intelligence enrichment, sandbox detonation, log analytics, and security-operations integration can increase operational value, but they are not detection approaches by themselves. They add context, response, or investigation around a decision. The ordering still matters: enrichment applied after representation consistency is useful context; enrichment applied to the wrong parse can make an incorrect decision look well supported.

IV. CROSS-CUTTING SYNTHESIS

This synthesis answers RQ2 by tracing how each evidence pathway changes when source identity rotates, payload realization changes, or the WAF and application parse different representations.

A. What source-path diversity changes

The source-path literature does not support a single "proxy detector" conclusion. It supports a more specific observation: the stability and meaning of source identity vary with the path. Residential providers rotate gateways and can exhibit pool overlap [5]. Shared-IP inference benefits from prefix-level diversity signals [12]. RTT can be manipulated by scheduling [2]. CDN termination can replace the client-side transport view with an edge-side view [4].

For WAFs, the implication is that IP reputation alone is an incomplete identity signal under source-path diversity. A detector needs either a richer set of features or a state model that can combine evidence across requests. The evidence does not show that every proxy family is equally hard to distinguish, nor that a particular fingerprint fails universally. It shows why claims based on one stable source attribute do not transfer cleanly across paths.

The operational consequence is not simply that more features are needed. Feature value depends on where the feature is generated. A TLS fingerprint generated at the edge may be useful for challenge selection but unavailable to the origin WAF. A session key may survive address changes but be absent for unauthenticated traffic. An IP prefix may reveal aggregation while also joining unrelated households. This is why a list of signals is less informative than a description of the observation boundary and the state transitions around it.

B. What payload mutation changes

The payload literature provides more direct WAF evidence than the source-path literature. WAF-A-MoLE [9] and BWAFSQLi [24] together demonstrate mutation-related degradation in evaluated research configurations and, in BWAFSQLi, commercial black-box products. Their results cover token, structural, and encoding-related transformations. They therefore establish a practical evaluation requirement: mutation must be part of the test distribution, and the mutation policy must be described well enough to distinguish semantics-preserving changes from malformed

or non-equivalent requests.

The evidence becomes less direct as the transformation moves from token-level obfuscation toward semantic paraphrase. A detector can recognize a known attack family while relying on lexical patterns that disappear under paraphrase. Conversely, a detector trained on semantic features might generalize better, but that hypothesis has not been directly established in the reviewed WAF corpus.

There is also a timing issue. A parser discrepancy can be exploited without waiting for a behavioral model to accumulate enough requests to notice anything unusual. In that sense, normalization is a front-line control, whereas behavior is a longer-horizon control. Conversely, a normalized request can still be a malicious request whose surface form evades rules. Treating all failures as "the detector missed it" hides this difference in position and remediation.

C. Why parser discrepancies deserve a separate column in the argument

Parser-level studies [1, 23] reveal a failure mode that sits before semantic classification. If the WAF and application disagree about boundaries, fields, encodings, or content types, the classifier receives an input that is not equivalent to what the application executes. The issue is therefore not simply that the detector lacks a better model; it may be observing the wrong object.

This distinction also explains why protocol results should not be used to populate ordinary payload-mutation claims. WAFFLED explicitly keeps the attack payload unchanged. Its bypasses are direct evidence of parser discrepancies, and they motivate normalization and differential parsing tests. Treating them as B1–B5 evidence would blur an important engineering boundary.

The audit is diagnostic: it shows where a conclusion depends on combining results from different tasks. One paper may establish that a proxy-identification detector survives a scheduling attack, while another shows that a WAF is vulnerable to SQL mutation. Together they motivate a deployment hypothesis, but they do not constitute a measured result for a WAF exposed to both conditions. The same distinction applies to B5: a semantic model is a plausible candidate for paraphrase resistance, while paraphrase robustness remains an empirical question.

D. The missing joint experiment

The central gap concerns composition. The reviewed corpus contains candidate signals and useful one-axis experiments, but no identified study varies source-path conditions and payload realization in the same WAF evaluation while also validating the downstream application-level effect. Source-path work shows that source identity can rotate, overlap, or be manipulated [2, 5, 12]. WAF studies show that payload representation can alter the decision [9, 24], while parser-level work shows that WAF and application interpretations can diverge even when the attack payload is not rewritten [1, 23]. A deployment claim that combines these findings therefore needs a controlled experiment in which the dimensions are changed together and the application-level effect is checked.

That gap also changes how evidence pathways should be compared. A behavioral detector is better positioned than an IP blocklist to retain evidence across changing source addresses; an intent-oriented model is better positioned than a literal signature to tolerate some surface changes. Those are design expectations, not measured superiority claims. The missing experiment is therefore not a contest between labels. It is a test of whether the layers continue to work when their preferred signals are disturbed simultaneously.

E. Comparing approaches by evidence and operational role

TABLE I COMPARISON OF EVIDENCE PATHWAYS AND OPERATIONAL ROLES

Evidence pathway or control	Primary evidence	Evidence under source-path variation	Evidence under payload mutation	Main unresolved issue
Signatures and rules	Rules, known patterns, parsed/canonicalized request fields	No source-path factor reported in the reviewed WAF tests	Mutation-related bypasses are observed across evaluated WAFs, but not as a controlled rule-only comparison	Joint protocol and mutation robustness
TLS/HTTP fingerprints	ClientHello and HTTP/2 behavior [3, 19, 20]	Fingerprint stability is path- and observation-point-dependent; adjacent evidence only	Not a payload detector; no direct mutation evidence	Robustness to adaptive clients and proxy homogenization
Active challenges	Browser execution and continuity [7, 16]	Joint rotation evidence is not reported in the reviewed tests	Not established as payload-intent detection	Joint evaluation with real browsers and rotating paths

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

Evidence pathway or control	Primary evidence	Evidence under source-path variation	Evidence under payload mutation	Main unresolved issue
Surface-form ML/DL	Characters, tokens, request features [9, 22, 24]	Not established without source/session features	Direct mutation evidence for evaluated configurations	Dataset shift and adversarially chosen transformations
Intent-oriented language models	Contextual language representations [13]	Not established without session or source features	Semantic robustness remains untested under validated B5 paraphrase	Shortcut learning, latency, and controlled paraphrase tests
Behavioral/session linkage	Rates, correlations, graphs, prefix aggregates [2, 6, 10, 12, 18]	Mechanistically suited to continuity; positive evidence is adjacent, not WAF-specific	Not established unless payload features are fused and tested	Privacy-preserving linkage and joint WAF benchmarks
Protocol normalization	Aligned security-relevant parsing before inspection [1, 23]	Not primarily a source-identity control	Addresses parser disagreement, not ordinary payload mutation	HTTP/2, custom parsers, and deployment cost

Table I separates observed evidence from mechanism-based expectations; it is not a league table. No reviewed pathway or control dominates every axis. The pattern is complementarity: normalization aligns the representation, content models inspect surface form or tested semantic cues, and behavioral signals provide cross-request context. A WAF combining these layers still needs evaluation under their combined threat conditions.

V. EVALUATION PRACTICE AND EVIDENCE QUALITY

To answer RQ3, the evaluation audit separates task definition, temporal context, data validity, product observability, and reporting practice rather than treating them as one generic evaluation problem.

A. Task mismatch

The reviewed sources answer different questions. Inside Residential IP Proxies [5] measures proxy infrastructure and gateway behavior. Beyond RTT [2] measures residential-proxy identification. Giotsas et al. [12] measure shared-address classification rather than malicious-request detection. WAF-A-MoLE [9] and BWAFSQLi [24] measure payload-driven WAF evasion. WAFFLED [1] and Break the Wall [23] measure parser discrepancies. These results are complementary, but they cannot be added as if they were repetitions of the same endpoint.

A survey should therefore preserve the task boundary in every transfer. “A behavioral method retains performance under scheduling in proxy identification [2]” is a defensible statement. “Behavioral WAF detection is robust to residential proxy attacks” is a stronger statement that requires a WAF-specific experiment.

B. Single-request versus session-level evaluation

Many commonly used web-attack datasets are organized as labeled requests [8, 11]. This is convenient for supervised learning and reproducible benchmarking, but it can hide the value of cross-request evidence. The detector does not get to observe source rotation, request timing, destination diversity, or the relationship among several partially suspicious requests. Conversely, session-level systems require state, identity linkage, and a clear privacy model. The literature has not converged on a shared protocol for evaluating these trade-offs.

ET-BERT [15] demonstrates the broader value of contextual representations for encrypted-traffic classification, but it is not a WAF robustness study. Its relevance is methodological: sequence context can carry information that is absent from isolated packets or requests. The transfer should be reported as an analogy, not as direct evidence.

C. Dataset provenance and mutation validity

The CSIC HTTP dataset [8], ECML/PKDD data [11], and related IDS datasets provide useful starting points, but their age, collection process, and attack distribution affect what a model learns. CICIoT2023 [17] illustrates a more recent benchmark design for large-scale attacks, yet its IoT setting does not make it a WAF benchmark. Generic IDS surveys [14, 18] repeatedly identify dataset shift and class imbalance as obstacles to reliable comparison.

Adversarial evaluation adds a second provenance problem. A mutation should preserve the application-level behavior being studied. If a transformation changes the SQL semantics, breaks parsing, or produces a request that no real application accepts, the observed bypass may not measure the intended robustness property. WAF-A-MoLE [9] and BWAFSQLi [24] are valuable precisely because they make mutation an explicit part of the experiment. Future studies should also publish transformation validity checks, application responses, and the boundary between valid evasion and malformed input.

D. Black-box products and hidden mechanisms

A black-box bypass result is valuable at the product level. It becomes misleading when the paper assigns that result

to a specific hidden detection mechanism without architectural evidence. BWAFSQLi [24] separates gray-box deep-learning configurations from black-box products; that distinction should be retained. Similarly, vendor documentation can demonstrate that a challenge or fingerprint feature exists [4, 7, 16], but it does not quantify its robustness against an adaptive adversary.

Any automated mutation workflow needs a visible denominator: how many generated requests remained valid at the application, how many bypassed the WAF, and how many were simply malformed. DaNuoYi makes "valid test inputs" its central output [26], while WAFBOOSTER reports true rejection and false rejection after signature generation [25]. Those are useful but different denominators. Blocking a generated test set is not equivalent to preserving benign traffic under production load, and a rule that closes one mutation family may leave another representation untouched.

E. Metrics and reporting

Accuracy, F1, false-negative rate, latency, and challenge completion are not interchangeable. A WAF study should report the attack-preservation criterion, false-negative rate under each transformation family, false-positive cost on benign traffic, and the WAF action (blocked, challenged, or passed). For challenges, record whether the challenge was presented, passed, or failed and whether a later request reached the application. Record transport or infrastructure errors separately when no WAF decision is available. For source-path diversity, report how identity is defined after rotation, whether the same browser or session is retained, and whether prefix or session context is available. For parser discrepancies, report both WAF-side and application-side interpretations.

Table II consolidates these requirements into a minimum reporting checklist. A joint benchmark should report denominators in separate dimensions rather than as one flat set of outcomes. Using a reference application or interpreter, count requests that are accepted and achieve the intended effect, requests that are accepted but do not preserve the intended effect, and malformed or unsupported requests. Then record the WAF action, challenge presentation and outcome, and application-level effect independently. This prevents a bypass rate from conflating parser failure, challenge handling, and successful attack delivery.

TABLE II MINIMUM REPORTING CHECKLIST FOR COMPOSITION-ORIENTED WAF EVALUATION

Report item	Required value or description
Observation point	Client, edge, WAF, reverse proxy, or origin; state which fields are visible.
Parser / normalization	Parser versions, canonicalization rules, normalization policy, and disagreement flag.
Source-path condition	Residential, datacenter, VPN relay, CDN edge, carrier-shared, rotation schedule, and session continuity.
Mutation family	B1–B5 operator, composition order, adaptive or fixed generation, and payload validity check.
Attack preservation	Reference application result: intended effect achieved, not achieved, or not applicable.
WAF outcome	Blocked, challenged, or passed; keep transport and infrastructure failures separate.
Challenge state	Not presented, presented and passed, or presented and failed.
Context and version	Session state, model/product version, rule set, browser/client, and deployment configuration.
Denominators	Benign set, valid attack set, malformed set, and indeterminate trials reported separately.

VI. RESEARCH AGENDA

To answer RQ4, the agenda follows the paper's evidence distinction. It begins with a joint benchmark, then addresses linkage, mutation-invariant models, parser-aware pipelines, and reporting standards. The order is deliberate: evaluation must make the claimed robustness observable before model improvements can be interpreted.

A. Joint benchmarks that preserve attack intent and application-level effect

The highest-value next step is a benchmark in which source-path conditions, payload realization, and protocol representation are varied in a controlled factorial design. Each trial should record three separate outcomes: (1) WAF action—blocked, challenged, or passed; (2) challenge state—not presented, passed, or failed; and (3) application-level effect—intended effect achieved, not achieved, or not applicable because the request did not reach execution. Parser disagreement should be an orthogonal flag when the WAF and origin interpret security-relevant fields differently. If transport or infrastructure failure prevents a WAF decision, record the trial as indeterminate rather than as blocked. Keeping these fields separate avoids treating challenge as a terminal action or malformed input as a successful semantic bypass, and supports comparison across systems with different action spaces.

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

The benchmark can use controlled emulations of source paths to make interactions visible and repeatable. Protocol normalization or parser discrepancy should be tested as a separate factor from payload mutation. Application-side validation should establish whether the intended effect occurred, while WAF action and challenge state are logged independently.

Such a benchmark would allow direct estimates of interaction effects. It could answer whether a detector that handles payload mutation on a fixed source continues to work after source rotation, and whether behavioral context compensates for the loss of a stable IP. The proposed name "SourcePathMutationBench" is used here for a benchmark concept, not for an existing dataset.

B. Behavioral–semantic fusion with privacy-aware linkage

A second direction is to combine request content with cross-request context. The core design problem is identity linkage: source IP is unstable and shared, while raw session identifiers or browser fingerprints can be sensitive. Candidate solutions include short-lived authenticated session keys, keyed hashes retained for limited windows, and prefix-level or browser-level aggregation with explicit privacy controls. The model can then combine semantic scores, request timing, destination diversity, and mutation consistency.

Graph-based anomaly detection [10] and multivariate anomaly reviews [6, 18] provide methodological starting points. Beyond RTT [2] supports the intuition that correlation features can remain useful when a single timing feature is manipulated. These sources do not validate a WAF implementation; they motivate an experiment in which the fused detector is evaluated against both proxy scheduling and payload mutation.

The same benchmark-and-repair loop can support defense, but the repair target has to be explicit. Is the goal to make the score stable under mutation, increase the decision margin, or route uncertain requests to a challenge? Each choice changes the false-positive and latency trade-off. A useful experiment would report the trade-off rather than presenting rule repair as a universal fix.

C. Mutation-invariant semantic models

A third direction is to train and evaluate models around intent-preserving transformations. The evaluation should include the B1–B4 mutation families defined in this survey, using operators documented in [9, 24] where applicable, as well as a carefully validated B5 paraphrase set. Training can include adversarial augmentation, contrastive objectives that pull equivalent attacks together, and application-level checks that ensure semantic preservation.

The key outcome is not simply a higher accuracy number. It is a controlled comparison showing whether the model's decision remains stable when lexical and structural cues change. Reports should include benign paraphrases, cross-application transfer, and adaptive attacks that search for a low-scoring realization.

D. Parser-aware WAF pipelines

Protocol normalization should be treated as part of the detection pipeline rather than as a generic payload-mutation defense. Differential testing can compare the WAF's parse tree with the application's parse tree across HTTP versions, content types, duplicate fields, boundary conditions, and encoding stacks. WAFFLED [1] and Break the Wall [23] show why this is necessary. The remaining research questions include HTTP/2 and HTTP/3 behavior, custom application parsers, reverse-proxy chains, and the cost of strict normalization in legacy deployments.

The minimum reporting fields are summarized in Table II. Using the same fields across studies makes it possible to compare whether two WAF evaluations test the same property, without treating a larger feature bundle as evidence of robustness.

E. Evaluation standards for claims of intelligence

The term "intelligent WAF" should be tied to an explicit evidence contract rather than to the presence of several bundled features. A defensible claim should specify the observed input layers, the attack transformations, the session context, the product or model configuration, the denominator, and the failure definition. Claims about semantic understanding should include intent-preserving tests. Claims about bot, proxy, or other source-path classification should include path variation and adaptive timing. Claims about integrated protection should identify which component made the decision.

This standard retains commercial integration as a valid deployment pattern while making the detection claim auditable. The presence of several features is then treated as an implementation description, not as evidence of robustness by itself.

VII. LIMITATIONS

The review uses the collected literature and a bounded set of search terms. The corpus spans WAF, networking, and machine-learning sources, including recent papers and preprints; A1–A4 and B1–B5 are analytical categories, and a single request may combine several path properties or mutation operators.

The survey reports each result within the task, configuration, and metric defined by its source. Proxy detection and generic intrusion-detection results remain labeled as adjacent evidence, while the numerical synthesis is reserved for studies with comparable WAF endpoints.

VIII. CONCLUSION

RQ1—The reviewed approaches divide by the evidence available to the decision path: signatures and surface-form models inspect request representations; fingerprints and

source features inspect client or transport context; challenges impose active interaction; behavioral methods retain cross-request context; and normalization aligns the representation before inspection. These are complementary operational roles, not interchangeable labels.

RQ2—Source-path diversity, payload mutation, and parser discrepancy disturb different parts of that path. Source rotation weakens single-attribute identity signals, payload mutation changes the surface seen by content detectors, and parser discrepancy can make the WAF inspect an object different from the one the application executes. The literature supplies direct evidence for these axes separately, with the strongest WAF-specific results on payload and parser conditions.

RQ3—Comparability is constrained by task mismatch, observation point, request granularity, mutation validity, hidden product mechanisms, and inconsistent denominators. A proxy-identification result cannot be reported as WAF attack-detection robustness, and a passed challenge is not the same outcome as a request whose intended application-level effect was executed.

RQ4—The next useful step is a composition-oriented benchmark that varies source-path conditions, payload realization, and protocol representation in a controlled design, while recording WAF action, challenge state, parser agreement, and application-level effect independently.

Taken together, the review argues that the meaningful unit of WAF evaluation is not an isolated detector label but the evidence path through which a request is observed, transformed, classified, and executed. Source-path diversity changes the continuity of identity signals; payload mutation changes the surface available to content models; parser disagreement can change the object that reaches the application. A WAF can therefore be called intelligent in a research sense only when it is evaluated across these interacting conditions and its decision is connected to the resulting application-level effect. The central contribution of this survey is the evaluation logic that makes that claim testable.

REFERENCES

[1] S. A. Akhavani, B. Jabiyev, B. Kallus, C. Topcuoglu, S. Bratus, and E. Kirda, "WAFFLED: Exploiting parsing discrepancies to bypass web application firewalls," in Proc. IEEE Annu. Comput. Security Appl. Conf. (ACSAC), 2025, pp. 689–702, doi: 10.1109/ACSAC67867.2025.00062.

[2] T. Ali, S. Yoosuf, M. Rabhi, M. Al-Sabah, and H. Yun, "Beyond RTT: An adversarially robust two-tiered approach for residential proxy detection," in Proc. 2026 Network and Distributed System Security Symp. (NDSS), 2026, doi: 10.14722/ndss.2026.242086.

[3] J. Althouse, "JA4+ network fingerprinting," FoxIO, 2023. [Online]. Available: https://blog.foxio.io/ja4+-network-fingerprinting

[4] Amazon Web Services, "Amazon CloudFront launches support for JA4 fingerprinting," AWS News, 2024. [Online]. Available: https://aws.amazon.com/about-aws/whats-new/2024/10/amazon-cloudfront-ja4-fingerprinting/

[5] E. Chiapponi, M. Daciér, and O. Thonnard, "Inside Residential IP Proxies: Lessons Learned from Large Measurement Campaigns," in Proc. IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2023, pp. 501–512, doi: 10.1109/EuroSPW59978.2023.00062.

[6] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," IEEE Access, vol. 9, pp. 120043–120065, 2021, doi: 10.1109/ACCESS.2021.3107975.

[7] Cloudflare, "Turnstile documentation," 2026. [Online]. Available: https://developers.cloudflare.com/turnstile/

[8] Information Security Institute, Spanish National Research Council, "HTTP dataset," 2010. [Online]. Available: https://www.kaggle.com/datasets/ispangler/csic-2010-web-application-attacks

[9] L. Demetrio, A. Valenza, G. Costa, and G. Lagorio, "WAF-A-MoLE: Evading web application firewalls through adversarial machine learning," in Proc. 35th ACM/SIGAPP Symp. Appl. Comput. (SAC), 2020, pp. 1745–1752, doi: 10.1145/3341105.3373962.

[10] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 5, 2021, pp. 4027–4035, doi: 10.1609/aaai.v35i5.16523.

[11] ECML/PKDD, "Discovery challenge dataset," 2007. [Online]. Available: https://gitlab.fing.edu.uy/gsi/web-application-attacks-datasets/-/tree/master/ecml_pkdd

[12] V. Giotsas, L. Salamatian, A. Cordelle, N. Wood, and M. Fayed, "A global inference and assessment of large shared IP addresses," in Proc. ACM SIGCOMM, 2026, doi: 10.1145/3789240.3829192.

[13] Y. Khalaf, "LLM-WAF: An intelligent web application firewall powered by large language models for advanced threat detection," Int. J. Wireless Microw. Technol., vol. 16, no. 4, pp. 317–327, 2026, doi: 10.5815/ijwmt.2026.04.18.

[14] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, Art. no. 20, 2019, doi: 10.1186/s42400-019-0038-7.

[15] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification," in Proc. ACM Web Conf. (WWW), 2022, pp. 633–642, doi: 10.1145/3485447.3512217.

[16] Microsoft, "Web application firewall JavaScript challenge," Microsoft Learn, 2025. [Online]. Available: https://learn.microsoft.com/en-us/azure/web-application-firewall/waf-javascript-challenge

[17] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," Sensors, vol. 23, no. 13, Art. no. 5941, 2023, doi: 10.3390/s23135941.

[18] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, K. Kloft, T. G. Dietterich, and K.-R. Müller, "A Unifying Review of Deep and Shallow

ISSN 3106-7018 (Print) ISSN 3106-5791 (Online)

Anomaly Detection," Proc. IEEE, vol. 109, no. 5, pp. 756–795, 2021, doi: 10.1109/JPROC.2021.3052449.

[19] Salesforce, "JA3: TLS fingerprinting," GitHub repository, 2017. [Online]. Available: https://github.com/salesforce/ja3

[20] O. Segal, A. Fridman, and E. Shuster, "Passive fingerprinting of HTTP/2 clients," Akamai, Black Hat Europe, 2017.

[21] SQLMap Project, "SQLMap: Automatic SQL injection and database takeover tool," GitHub repository. [Online]. Available: https://github.com/sqlmapproject/sqlmap

[22] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[23] Q. Wang, J. Chen, Z. Jiang, R. Guo, X. Liu, C. Zhang, and H. Duan, "Break the Wall from Bottom: Automated Discovery of Protocol-Level Evasion Vulnerabilities in Web Application Firewalls," in Proc. IEEE Symp. Security Privacy (SP), 2024, pp. 185–202, doi: 10.1109/SP54263.2024.00129.

[24] B. Zhang, C. Liu, R. Ren, Q. Wang, and J. Ren, "BWAFSQLi: Bypassing web application firewall with adversarial SQL injections," ACM Trans. Softw. Eng. Methodol., vol. 35, no. 9, Art. no. 290, 2026, doi: 10.1145/3788286.

[25] C. Wu, J. Chen, S. Zhu, W. Feng, K. He, R. Du, and Y. Xiang, "WAFBooster: Automatic Boosting of WAF Security Against Mutated Malicious Payloads," IEEE Trans. Dependable Secure Comput., vol. 22, no. 2, pp. 1118–1131, 2025, doi: 10.1109/TDSC.2024.3429271.

[26] K. Li, H. Yang, and W. Visser, "DaNuoYi: Evolutionary Multitask Injection Testing on Web Application Firewalls," IEEE Trans. Softw. Eng., vol. 51, no. 9, pp. 2412–2431, 2025, doi: 10.1109/TSE.2023.3343716.