International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 10 | Oct 2026 www.irjet.net p-ISSN: 2395-0072

Structured Comprehensive Literature Review

Detecting Autonomous AI Browsing Agents Using Explainable Machine Learning and Behavioral Metrics: A Comprehensive Review

Aditi Singh Rathore[1], Dr. Uday Pratap Singh[2]

[1]*Student, Department of Artificial Intelligence and Data Science, Poornima Institute of Engineering and Technology, Jaipur, Rajasthan, India.*

[2]*Assistant Professor, Department of Artificial Intelligence and Data Science, Poornima Institute of Engineering and Technology, Jaipur, Rajasthan, India*

Abstract - *"Autonomous AI browsing agents" which integrate the linguistic or multimodal models with the browser control in order to perform multiple-step tasks pose significant difficulties in security defense of scripted bot traffic on websites. In this review, we try to investigate whether such autonomous AI browsing agents may be identified from human users or conventional scripted bots through behavioral metrics and explainable machine learning and what problems still exist. This work should be treated as structured, comprehensive literature review, not as a systematic one nor as experimental investigation. From the initial set of papers dedicated to mouse dynamics bot detection, automated browsing characterization with honeypot sites, biometric fusion of web logs and mouse movements, and bot detection with synthetic trajectory data and explainable biometrics, we further explore 2018–2026 literature in fields of behavioral biometrics, fingerprinting, machine learning, and explainability. Based on existing literature, it can be said that the mouse dynamics and timing signals have been studied most thoroughly; that static fingerprinting is cheap but evadable and lacks informational utility for real browser-driving agents; that combining fingerprinting signals increases the evasion cost in the studied testbeds; and that explainability is used for the other biometric modalities, with the latter being studied through model-centric methods only. In recent preprints, it has been shown that there is evidence that artificial intelligence agents may also leave traces in their behavior and execution and that the distinction between human, bots, and agents should be considered. The major weaknesses are small and controlled datasets, heterogeneity in evaluation, adversarial attacks, drift, and security and privacy vulnerabilities. Existing weaknesses of the studies include lack of three-class research, lack of benchmark standardization, little explanation for analysts, and lack of privacy-preserving detection.*

Keywords - Autonomous AI Agents; AI Browsing Agents; Web Bot Detection; Behavioral Biometrics; Mouse Dynamics; Explainable AI; Machine Learning; Human-Computer Interaction

1. INTRODUCTION

Automation is one of the characteristics of today's Web. Automated request to the resources on the web and their automated response analysis is the cornerstone of search engine crawlers, monitoring systems, pricing aggregators, and testing tools; according to industry reports quoted in scientific literature, 37% to 40% and higher of visits to websites are made by bots [1]-[3]. The numbers are estimated by the industry and should be treated as such, rather than hard numbers. Automation does not necessarily mean any bad intent and encompasses actions like indexing, hyperlink verification, feed download, and data extraction [2]. Malicious automation is helpful for credential stuffing, file discovery, and vulnerability probing [1], as well as for scalping, hoarding, price scraping, and spamming account creation [2].

Development of the technical abilities of bots happened gradually. In the beginning, bots were just scripts which could be differentiated from browsers by certain features such as their headers, lack of ability to use JavaScript, and others [2]. After that, browser automation software allowed bots to have all the abilities of the browsers and interact with websites using mouse movements and clicks [2]. A further class of evasive bots combines a browser-like fingerprint with humanlike timing and navigation, and a parallel literature shows that challenge-response tests can themselves be defeated by machine learning [4]–[6]. Figure 1 summarises this progression as a conceptual synthesis.

The latest development is the advent of autonomous AI browser agents: an agent that combines language or multimodal models with a browser control system that pursues user objectives through perception and decision-making through many iterations of action [7]–[11]. Recent measurement work describes AI browsing agents as an emerging class of AI-powered bots that typically operate through real browsers and perform everyday tasks, which makes them difficult to detect [12], and characterises web agents as capable of solving anti-bot mechanisms and mimicking human behaviour [13]. The difference from scripted automation is conceptual as well as technical. A script executes a predetermined sequence; an agent selects

each action from the observed state of the page, so its action sequence is not fixed in advance. Detectors that rely on stable request patterns, static fingerprints or fixed timing are therefore less well matched to this adversary, although the literature also reports behavioural artefacts of agents that can be exploited [12], [14].

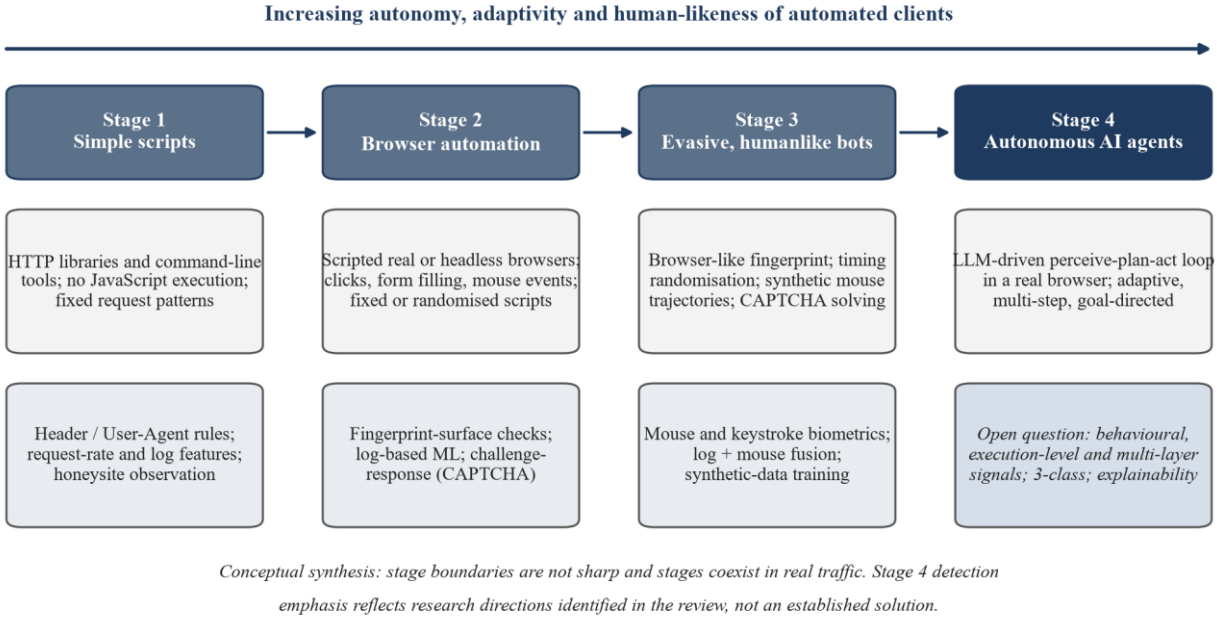


Figure 1. Evolution from traditional web bots to autonomous AI browsing agents (conceptual synthesis from the reviewed literature).

This shift raises a question of problem formulation. Most behavioural bot-detection studies reviewed here pose a binary human-versus-bot problem [2], [3], [15]. A recent preprint argues that a binary label space cannot represent AI agents as a distinct traffic class and reports, on its own controlled benchmark, that binary classifiers labelled a substantial fraction of agent sessions as human (39.1% and 34.5% for two model families) [14]. We present the human / traditional bot / autonomous AI agent distinction as an emerging perspective and an open research question, supported so far by a small number of preprints, rather than as an established requirement of every detector.

Explainability is the second theme of this review. A detector that returns only a "bot probability" gives a security analyst little basis for triage, for tuning thresholds, for recognising a systematic failure such as the misclassification of users of assistive input devices, or for answering a user who is challenged or blocked. Automated decisions that significantly affect individuals are also subject to regulatory constraints in some jurisdictions [16], [17]. The explainable-biometrics literature indicates that explanation methods have been applied mainly to face-based systems, that explanations are evaluated almost exclusively with model-centric measures, and that end-user perception is rarely examined [18].

1.1 Motivation, scope and objectives

The motivation of this review is the absence of a synthesis that connects four bodies of work that have developed largely separately: web-bot detection, behavioural biometrics and mouse dynamics, fingerprinting of browsers and networks, and explainable machine learning. The scope of the review will encompass research on automated and AI-powered browsing, including behavioral, execution-level, fingerprinting, network and web logs data, concentrating mainly on the time period from 2018 till 2026 and earlier literature when appropriate. Detection of social media bots based on content and creation of CAPTCHA for this purpose falls outside of the scope unless the technique contributes to detection. The goals are to review the existing evidence, identify explainability approaches that can be applied and the way they are evaluated and highlight areas where the literature does not provide answers.

1.2 Review questions

The central review question is: How can autonomous AI browsing agents be distinguished from humans and conventional automated bots using behavioural metrics and explainable machine learning, and what limitations remain in current detection approaches? It is decomposed into ten supporting questions:

RQ1. What behavioural characteristics distinguish humans, traditional bots and autonomous AI browsing agents?
RQ2. What behavioural metrics have been used for web-bot and behavioural-biometric detection?
RQ3. Which machine-learning approaches have been used for behavioural bot detection?
RQ4. How are mouse dynamics, keystroke dynamics, navigation patterns, timing and web logs used as signals?
RQ5. How can browser, device, TLS and network fingerprints complement behavioural metrics?
RQ6. Which XAI techniques are applicable to ML-based AI-agent detection?
RQ7. How are explanations evaluated in security and biometric applications?
RQ8. What limitations make existing approaches vulnerable to humanized or adaptive agents?
RQ9. What datasets and benchmarks are available?
RQ10. What research gaps remain for explainable detection of AI browsing agents?

1.3 Contributions and organisation

The review makes six contributions: (1) a structured taxonomy of signals for detecting automated and AI-driven browsing; (2) a comparative synthesis of behavioural, fingerprinting, network, web-log and execution-level methods; (3) a review of machine-learning methods used for behavioural bot detection; (4) an analysis of explainability techniques relevant to interpretable detection and of how explanations are evaluated; (5) an evidence-based identification of gaps around human / bot / AI-agent differentiation; and (6) a research agenda, clearly separated from established findings. Section 2 gives conceptual background. Section 3 describes the methodology. Section 4 reviews autonomous AI browsing agents, Section 5 behavioural metrics, Section 6 machine-learning methods and Section 7 explainable machine learning. Section 8 synthesises and compares detection approaches, Section 9 reviews datasets and evaluation, and Section 10 discusses challenges. Sections 11 and 12 present research gaps and future directions, Section 13 discusses the findings, Section 14 addresses threats to validity and Section 15 concludes.

2. BACKGROUND AND CONCEPTUAL FOUNDATIONS

2.1 Web bots

A web bot is a program that issues web requests and processes responses without direct human interaction. Benign uses include web indexing, website monitoring, data extraction and feed fetching [2]; benign bots such as search-engine crawlers are expected to honour the robots exclusion conventions, whereas many malicious bots do not [1]. The literature classifies bots by sophistication, but without a universal definition. One scheme distinguishes simple bots (scripts without a browser-like fingerprint or humanlike behaviour), moderate bots (browser-like fingerprint and main browser functions, but no humanlike behaviour) and advanced bots (browser-like fingerprint and humanlike behaviour) [2]. Other studies label bots by function, purpose or complexity [2]. These differing definitions complicate comparison across studies (Section 14).

2.2 Traditional scripted automation

Traditional automation is typically implemented with HTTP libraries and command-line tools. In a seven-month honeysite study, most clients that claimed to be Chrome or Firefox were found to misstate their identity, largely matching the TLS and header fingerprints of common Go and Python HTTP libraries and tools such as wget and curl [1], and only a small minority (0.63%) of bot sessions supported JavaScript functionality [1]. Such clients have a bot-like fingerprint and perform no mouse movement, which led one study to treat them as detectable in a deterministic way [2].

2.3 Browser automation

Browser-automation frameworks such as Selenium allow scripts to drive a full browser, supporting JavaScript, form filling and clicks [1], [2]. Vanilla configurations expose fingerprints that reveal automation, but further configuration can conceal them, and bots can alternatively drive a regular browser [2]. The fingerprint surface that distinguishes automation tools from ordinary browsers has been used both to detect bots and to detect the scripts that perform bot detection [19], and the resilience of fingerprinting as a means of blocking crawlers has been studied explicitly [20]. Third-party anti-bot services have been evaluated empirically [21].

2.4 Autonomous AI browsing agents

We use the following working definition: an autonomous AI browsing agent is a system that couples a (typically language- or multimodal-model-based) planner with a browser-control layer and pursues a user-specified goal over multiple steps by deciding each action from the observed page state. Foundational and benchmark work on language-model agents interleaves reasoning with acting [7], provides realistic web environments [8], datasets of tasks across real websites [9], and studies grounding of model plans in page elements [10] and screenshot-based end-to-end operation [11]. Section 4 develops this definition further.

2.5 Behavioural biometrics

Behavioural biometrics identify or characterise people from how they act, for example through mouse dynamics, keystroke dynamics, swipe dynamics or gait, in contrast with physiological traits such as fingerprints or faces [22].

Applications divide into identification (one-to-many) and authentication (one-to-one verification) [22]. Behavioural biometrics can be acquired with near-total transparency and are less invasive than many alternatives [3]. Bot detection differs from authentication in its target: it seeks to separate classes of actor (human versus automated) rather than to verify an individual, although the same signals and learning tools are used.

2.6 Mouse dynamics

Mouse dynamics comprise the cursor movements, clicks and drags a user generates on a graphical interface, usually recorded as timestamped coordinates and grouped into actions such as movement, point-and-click and drag-and-drop [22], [23]. The psychological grounding includes Fitts' law on the speed–accuracy trade-off in pointing [24], Hick's law on decision time [25], and the kinematic theory of rapid human movements [26], which models hand-movement velocity profiles as sums of lognormal strokes and underpins the neuromotor features used for bot detection [3]. Early authentication work includes neural-network classification of mouse behaviour [27] and re-authentication from mouse movements [28]; a 2024 survey of the field reviews tasks, raw attributes, features, datasets, algorithms and limitations [22].

2.7 Keystroke dynamics

Keystroke dynamics, the timing of key presses and releases, have been studied for longer than mouse dynamics [22]. Representative work covers authentication [29], benchmark comparison of anomaly-detection algorithms on a shared dataset [30], and large-scale deep-learning approaches [31]. Keyboard behaviour has also been combined with mouse behaviour for detecting blog bots [32].

2.8 Navigation and interaction behaviour

Navigation behaviour is observed at the server as sequences of requests (sessions) and at the client as scrolling and interaction with page elements. Session-based studies segment web logs by session identifier or by client attributes, commonly closing a session after 30 minutes of inactivity [2], and compute features such as request counts, response-code proportions, resource-type proportions and URL-depth statistics [2]. Early work exploited implicit patterns of human browsing, including mouse movement, stylesheet loading and following of invisible links [1], [33].

2.9 Browser fingerprinting

Browser fingerprinting collects attributes such as JavaScript APIs, plugins, fonts and canvas renderings to characterise a client [1], [34]. Fingerprints evolve over time, which complicates tracking [35], and in bot detection they can be evaded by changing the fingerprints of automation tools or by using regular browsers [2]. Fingerprinting also depends on the client executing JavaScript, which most bots in one large honeysite study did not do [1].

2.10 TLS and network fingerprinting

At the network layer, the TLS handshake (version, cipher suites, signature algorithms, elliptic curves and compression methods) reflects the underlying TLS library and therefore the client implementation [1]. Discrepancies between a declared User-Agent and the handshake have been used to identify HTTPS interception [36] and, by Li et al., to expose bots misrepresenting their identity [1]. TLS-level analysis has also been used to study application behaviour beyond browsers [37], and recent work applies JA4 fingerprints with gradient-boosted classifiers to separate bots from users [38]. IP-based attributes are weaker identifiers because bots may use proxies to evade rate limiting [1].

2.11 Machine-learning-based bot detection

Machine-learning detectors learn models of human and bot behaviour from past sessions and classify new visitors [2]. The log based detector has made use of algorithms like SVM, Random Forest, AdaBoost and MLP, and the mouse based detectors have made use of CNNs based on movement or classifiers based on actions [2]. The clustering algorithm has also been used to detect bots in addition to detection being offline (post session) or online (during session) [2]. One of the key challenges being faced is that of ground truth, where supervised detectors need labels for bots and users and the detection process becomes a chicken-and-egg situation [1].

2.12 Explainable AI and explainable machine learning

Terminology in this area is not uniform, and we adopt the following working distinctions, drawing on the cited literature. Interpretability concerns the degree to which a human can understand the cause of a decision [39]. Transparency refers to properties of a model that make it understandable in itself, at the level of the whole model, its components or its training algorithm [40]. Explainability is the provision of an explanation for a model's output, often through a separate method [40], [41]. Intrinsically interpretable models (for example sparse rules or small trees) are understandable by construction, whereas post-hoc explanations are produced after training for a model that is not itself transparent [41], [42]. A further axis separates model-specific methods, which depend on a model's internals, from model-agnostic methods, which treat the model as a black box [41], [43], and global explanations, which describe overall behaviour, from local explanations, which concern a single prediction [43], [44]. It has been argued that for high-stakes decisions inherently interpretable models should be preferred over post-hoc explanation of black boxes [42]; this position is

contested in the broader literature and is cited here as one view.

3. REVIEW METHODOLOGY

This article is a structured comprehensive literature review. It is not a formal systematic literature review (SLR): it does not report a protocol registered in advance, a PRISMA-style flow of retrieved and screened records, or independent dual screening. It does, however, borrow the organising discipline of the explainable-biometrics SLR [18], which examines tasks, datasets, machine-learning models, XAI methods and evaluation methods and separates methodology, findings, discussion, threats to validity and conclusions. Consistent with that discipline, we state below what was and was not done, and we report no counts of records retrieved or screened because no such counts were systematically recorded.

3.1 Research questions

The central question and RQ1–RQ10 are given in Section 1. RQ1, RQ2 and RQ4 are addressed mainly in Sections 4, 5 and 8; RQ3 in Section 6; RQ5 in Sections 5 and 8; RQ6 and RQ7 in Section 7; RQ8 in Sections 8 and 10; RQ9 in Section 9; and RQ10 in Sections 11–13.

3.2 Literature sources

The review started from a core set of studies selected by the authors: mouse-dynamics bot detection with sequence learning [15], characterisation of automated browsing with honeysites [1], the explainable-biometrics SLR [18], detection of advanced web bots by combining web logs with mouse biometrics [2] and BeCAPTCHA-Mouse [3]. A survey of mouse-dynamics biometrics [22] was also analysed in full and is used as a taxonomic reference. The set was expanded through (i) searches on publisher platforms and indexes (IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, USENIX), arXiv and general academic web search; (ii) backward citation chasing from the core studies; and (iii) targeted discovery of recent AI-agent detection work, which is published mostly as 2026 preprints. Bibliographic details of every cited item were checked against the Crossref metadata service, the arXiv programming interface or the publisher page; searches and checks were carried out in October 2026.

3.3 Search strategy

Search was iterative and purposive rather than exhaustive. Terms were combined and refined as relevant vocabulary was found, and the period 2018–2026 was prioritised, with older foundational works (for example on pointing, keystroke dynamics and explanation methods) retained where they ground later work.

3.4 Search keywords

The principal phrases were: "AI browsing agent detection", "autonomous browser agent detection", "web bot detection", "behavioral bot detection", "mouse dynamics bot detection", "keystroke bot detection", "behavioral biometrics bot detection", "automated browsing behavior", "explainable bot detection", "explainable AI cybersecurity", "XAI behavioral biometrics", "human bot agent classification", "browser automation detection" and "evasive web bots".

3.5 Inclusion criteria

A study was included if it (i) addressed at least one of the review themes (behavioural, execution-level, web-log, fingerprint or network evidence for distinguishing automated from human activity; ML methods for such detection; XAI methods or their evaluation in security or biometrics); (ii) described an identifiable method, dataset, evaluation or conceptual contribution; (iii) was published in a peer-reviewed venue, or was an arXiv preprint that directly addresses AI browsing agents; (iv) was available in English; and (v) had bibliographic details that could be verified.

3.6 Exclusion criteria

Items were excluded if their bibliographic details could not be verified; if they were commercial marketing or white papers offered as sole evidence for technical claims; if they concerned content-based social-media bot detection without browsing-behaviour signals; if they were duplicates or superseded versions; or if they addressed CAPTCHA design without bearing on detection. Industry traffic statistics are mentioned only as reported in peer-reviewed sources.

3.7 Literature classification

Each study was classified along the dimensions that structure the later sections and the taxonomy in Figure 6: detection target (human, traditional bot, evasive bot, AI agent), signal layer (challenge-response, fingerprint, network, web-log, behavioural interaction, execution-level), feature type, ML method, explainability method and evaluation method.

3.8 Data extraction framework

For each study we extracted: reference and venue; peer-review status; detection target; data source and scale; participants; bot or agent types; signals and features; ML method; XAI method (if any); evaluation metrics and protocol; main finding; stated limitations; and data or code availability. Where a study did not report an item we record "Not discussed" or "Not evaluated" rather than inferring it.

3.9 Quality assessment

No numerical scoring was applied. Instead, the following criteria were used to temper the weight given to a study: peer-review status; clarity of data collection and ground truth; transparency of the evaluation protocol (for example user-independent splits); consideration of evasion; availability of data or code; and the authors' acknowledgement of limitations. The explainable-biometrics SLR similarly assessed quality when selecting papers [18]. Preprints are flagged as such in the text and tables.

3.10 Evidence labelling and threats to validity

Throughout the paper, statements that report a cited study's findings carry citations. Statements that are the authors' own reasoning are signposted ("in our interpretation", "synthesis"), research gaps are labelled as such, and future directions are explicitly marked as proposals. Threats to the validity of this review are discussed in Section 14.

4. AUTONOMOUS AI BROWSING AGENTS

4.1 Definition and architecture

The agent literature describes a loop in which the system perceives the state of a web page, decides on an action, executes it through a browser, and observes the result. Reasoning and acting can be interleaved so that the model plans, acts and revises its plan from the outcome of each step [7]. Various approaches to perception exist. Some rely on page structure or HTML [9], while others rely on screenshots [11] for operations, and others rely on textual plans that are based on the elements identified by both of these means [10]. Realistic benchmark environments and tasks have been developed for testing the performance of agents performing multi-step tasks [8], [9]. The following discussion of experiments in empirical detection will employ multi-step tasks such as flight booking, online shopping, and forums [12].

These basic tasks are comprised of the actions carried out by the human who uses a web browser to navigate, click, scroll, enter information in forms, make choices, wait for the information and recover from mistakes through re-planning if the action fails. The above notion of agent represents the end product of the discussion on the above agent literature, and the capability of any given agent in carrying out these functions depends on the systems.

4.2 Browser control

For an agent to be able to employ the control layer where actions would be dispatched to the browser, the presence of a real browser [12] is a must. This point is crucial from the perspective of detection because only real browsers can exhibit genuine browser-level and TLS-level characteristics [1]. An action dispatch is not necessarily equivalent to physical input either. One study reports that a widely used automation framework does not emit the raw pointer-move and wheel-delta streams that a physical input device produces, and that this absence signature persisted under trajectory manipulation [14]; the authors stress that it is an artefact of browser automation rather than evidence of agent reasoning [14].

4.3 Empirical characteristics reported so far

Empirical evidence on agent behaviour is recent and comes almost entirely from preprints on instrumented websites. In a controlled measurement of seven AI browsing agents and human participants, the agents differed from humans and from one another in typing (paste-based versus keystroke-based entry), form filling (change-event-based), repeated delete-and-retype patterns, direct jumps to click targets without continuous mouse movement, and scrolling style [12]. Another study finds that an agent's actions and interaction timings, captured by a passive JavaScript tracker, suffice to identify the underlying language model with up to 96% F1 across fourteen models; randomised delays degraded classification but a classifier retrained on delayed traces largely recovered [45]. A multi-layer study of six agent frameworks finds structural differences in HTTP request assembly, connection setup and browser actions, with a decision-tree classifier reaching 97% accuracy in its setting [46]. A further study reports that some agents bypassed all evaluated anti-bot mechanisms, that agents could be distinguished from humans and from one another through network, HTTP and browser-level fingerprints, and that stealth mechanisms often increased rather than decreased detectability [13]. Vanilla agents who rely on large multimodal models for mobile interfaces had unnatural touch kinematics that could be detected easily, and several strategies like heuristic noise and behavior matching have been suggested in order to lower their detectability [47].

Taken together, these studies suggest that current agents leave detectable behavioural and execution-level traces, but the evidence base is narrow: a handful of agents, specific tasks, instrumented sites, small human samples and non-peer-reviewed reports. The studies also show that the adversary is adaptive, since the humanisation of agent behaviour is itself an active research topic [47].

4.4 Conceptual comparison

Table 1 provides examples of how humans, traditional bots, and AI bots compare. It is important to note that all rows contain evidence when there was any available; otherwise, they have been interpreted as "Conceptual" based on existing literature.

Table 1. Definitions and characteristics of humans, traditional bots and autonomous AI agents (cells marked "Conceptual" are the authors' interpretation)

Characteristic	Human	Traditional bot	Autonomous AI agent	Basis
Fixed behaviour	No; varies within and between individuals [22]	Typically yes: predetermined scripts or rules; moderate bots follow links randomly without waiting [2]	No predetermined sequence; actions chosen from page state (conceptual), yet agent- and model-specific regularities reported [45]	Evidence + conceptual
Page understanding	Semantic and visual	None or limited to selectors and patterns (conceptual)	Perceives page structure and/or screenshots through language or multimodal models [9]–[11]	Evidence for agents
Dynamic planning	Yes	No: fixed flow (conceptual)	Stepwise reasoning and acting with revision [7], [8]	Evidence for agents
Browser control	Physical input devices through the browser	HTTP libraries or automation frameworks [1], [2]	Programmatic control of a real browser [12]	Evidence
Mouse interaction	Continuous trajectories with neuromotor structure [3]	None, or straight-line movement between links [2]	Direct jumps to targets reported for some agents [12]; synthetic motion possible [47]	Evidence
Keyboard interaction	Variable dwell and flight times [29]	Injected strings with fixed or randomised delays (conceptual)	Paste-based or key-by-key entry; delete-and-retype patterns [12]	Evidence for agents
Timing behaviour	Variable reaction times [25]	Fixed or scripted sleeps, some scaled to text length [2]	Inter-action latency includes model inference (conceptual); timing identifies models [45]	Evidence + conceptual
Adaptation	Yes	Limited, rule-based (conceptual)	Adapts to layout changes and errors (conceptual); stealth features added by frameworks [13]	Conceptual + evidence
Task persistence	Variable	High for a single repetitive task (conceptual)	Multi-step goal pursuit with error recovery (conceptual)	Conceptual
Humanlike behaviour	Not applicable	Low; advanced bots engineered to mimic humans [2]	Variable; may mimic humans [13] but leaves artefacts [12], [14]	Evidence

4.5 How agent detection differs from traditional bot detection

There are three important distinctions that can be made concerning our approach to this problem. In the first place, because the agent's action chain depends on the page content, pattern-based detectors attempting to detect particular navigation or timing patterns become inefficient, but the features derived from the process of input generation (event stream, typing pattern, target switch) may be effective [12], [14]. Secondly, because of the use of real browsers by bots, the set of features revealing the discrepancy between the claimed and real client becomes much less informative compared to HTTP library bots [1], even though some layering of distinctions between agent frameworks has already been reported [13], [46]. Finally, the traffic is not necessarily malicious, since users may intentionally delegate actions to their bots, and therefore it is up to policy whether any reaction is appropriate or not. This motivates both the three-class perspective [14] and the emphasis on explanation in Section 7.

5. BEHAVIOURAL METRICS FOR AUTOMATED-AGENT DETECTION

5.1 Signal classification

We classify signals along three cross-cutting dimensions. Static signals describe the client and change little within a session (for example User-Agent, fonts, TLS parameters); dynamic signals arise from activity during the session. Behavioural signals describe how the actor interacts; contextual signals describe the environment or channel. Temporal features concern timing and order, and spatial features concern geometry, such as the path of a pointer. Figure 2 organises the signal layers discussed in this review, and Figure 3 details the behavioural metric families.

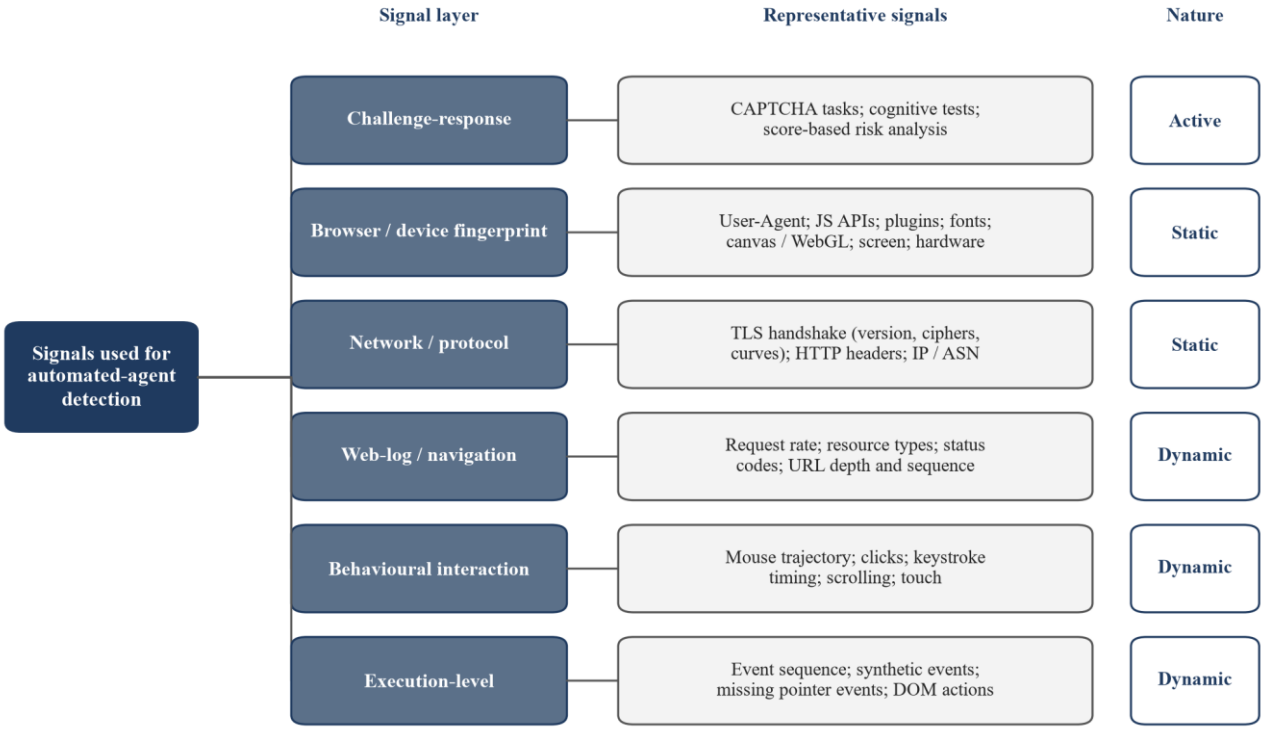


Figure 2. Taxonomy of signals used for automated-agent detection, by signal layer and nature (conceptual synthesis).

5.2 Mouse metrics

Mouse features in the literature are defined from timestamped cursor coordinates [22]. A survey of the field tabulates travelled distance, elapsed time, movement offset, straightness or efficiency, jitter, tremor (the ratio of interpolated path length to trajectory path), velocity and acceleration in horizontal, vertical and tangential directions, tangential jerk, angle of movement, angular velocity, rate of curvature, total angles and bending energy, and distributional descriptors such as central moments, skewness and kurtosis [22]. Pauses and direction changes follow from these quantities and from the timestamps. Beyond hand-crafted descriptors, neuromotor modelling represents each movement as a sum of lognormal strokes and derives a fixed-length feature vector from the stroke parameters and their number; this representation was designed to capture initial acceleration, final deceleration and fine endpoint correction, which are features of human movement that are hard for bots to emulate [3], [26]. This method does not use handcrafted features: the gestures are encoded into 2D matrices for CNN [2] or converted to time-series inputs, which are based on position change and speed of direction for sequential models [15]. The psychological laws provide us with extra features like target width and amplitude by Fitts' law, as well as endpoint dispersion, which is used very seldom in the literature for mouse authentication [22], [24]. To the best of our knowledge, there is no research where they tested the entropy of trajectory as a feature for recognition.

5.3 Click metrics

Click actions entail not only the action of clicking but also the action of approaching. The click parameters include click intervals and times, double-click time, distance between the click point and the target center, endpoint scatter (as compared to the target size), and movement prior to clicking [22],[24]. Pre-click movement is of particular interest for agent detection: agents have been observed to jump directly to click targets without continuous mouse movement [12], and a missing pre-click

movement track is a component of the absence signature noted above [14]. Earlier work on web robots similarly used mouse movement as evidence of a human visitor [1], [33].

5.4 Keyboard metrics

Keystroke dynamics are conventionally summarised by dwell (hold) time of a key, flight time between keys, inter-key intervals, typing speed and the variability and burstiness of these timings [29], [30]. Deep models operate directly on timing sequences at scale [31]. In agent detection, features include inter-key latency, hold latency and keystroke counts [12], and the manner of text entry itself (paste-based versus key-by-key, or delete-and-retype) differentiates agents [12]. Mouse and keyboard features were combined for blog-bot detection [32].

5.5 Navigation metrics

Session-level features computed from web logs include the total number of requests, total bytes, counts of GET, POST and HEAD requests, proportions of 3xx and 4xx responses, proportions of image, stylesheet and script requests, the HTML-to-image ratio, standard deviation of URL depth, maximum and average requests per page, and measures of consecutive sequential requests [2]. Timing between requests and the type of links requested were used in earlier crawler detection [1]. Further indicators include compliance with robots exclusion rules, which benign bots are expected to follow [1], loading of stylesheets and following of invisible links [1], [33]. At the client, scrolling distance, duration and pattern are used as features [12].

5.6 Temporal metrics

Timing signals include inter-event times, reaction or decision latency, clustering of actions and idle periods. Bots may randomise delays or scale them with the length of page text to appear human [2], and agent timing carries information about the underlying model, although randomised delays reduced such identification [45]. Hick's law links decision time to the number and uncertainty of alternatives [22], [25], which motivates reaction-time analysis, but we did not find a study that validates it as an agent-detection feature. In one benchmark, the rate of mouse events was among the most discriminative behavioural features [14].

5.7 Execution-level metrics

Execution-level metrics concern how an action is dispatched rather than what the action is: the order of events preceding a click, the presence or absence of raw pointer and wheel events, programmatic click signatures, change-event-based form filling and other synthetic event patterns [12], [14]. In one benchmark, a minimal set of two behavioural features (a mouse-event rate and a teleport-before-click ratio) gave complete observed agent recall at every evasion level, with the signal redundantly encoded across features [14]. In our interpretation, these signals are attractive because they are hard to fake by trajectory synthesis alone, but they depend on the automation stack. They may change if frameworks begin to emit device-level input, and the same authors caution that the signal reflects browser automation rather than agent reasoning [14].

5.8 Browser and network metrics

Browser-level signals include the User-Agent, HTTP headers, JavaScript APIs, plugin and font lists, canvas and WebGL renderings and device characteristics [1], [34]. Network-level signals include TLS handshake parameters, HTTP header ordering and IP attributes [1], [38]. Several limitations are documented. Fingerprints can be forged or evaded [2], [19], [20]. JavaScript fingerprints were obtained from only 0.59% of 1,760,124 honeysite sessions because most bots did not execute JavaScript [1]. In the agent measurement study, browser fingerprints showed limited discriminative power when shared by several agents, whereas behavioural fingerprints were distinctive [12]. Conversely, a multi-layer approach combining network and browser-interaction features distinguished six agent frameworks [46].

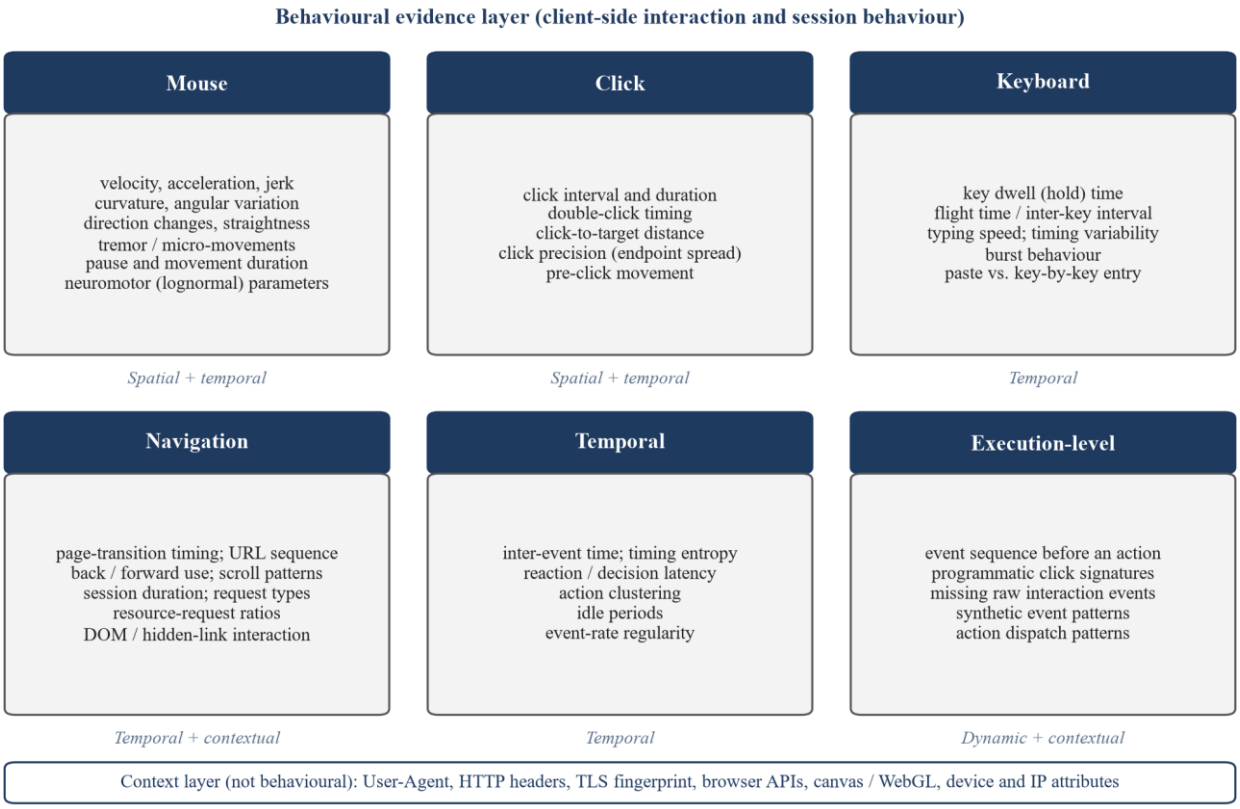


Figure 3. Behavioural metrics for AI browsing-agent detection, grouped by metric family and by spatial or temporal character (conceptual synthesis).

5.9 Feature taxonomy

Table 2 synthesises these metrics into a taxonomy that records, for each family, the signal attributes defined above. Two patterns emerge from the literature. Mouse kinematics, session-level log features and timing have received the most attention, with mouse-based work concentrated on authentication and bot detection [2], [3], [15], [22]. Keyboard, scroll and execution-level features are better represented in the recent agent-detection preprints than in the earlier bot-detection studies [12], [14].

Table 2. Behavioural feature taxonomy (S = static, D = dynamic, B = behavioural, C = contextual, T = temporal, Sp = spatial; attributes are the authors' classification)

Family	Metrics	Attributes	Typical use	Sources
Mouse kinematics	Velocity (x, y, tangential), acceleration, jerk, angular velocity, rate of curvature	D, B, T, Sp	Authentication; bot detection	[3], [22]
Mouse geometry	Travelled distance, straightness/efficiency, jitter, tremor, curvature, direction changes, bending energy, total angles, central moments	D, B, Sp	Trajectory shape; synthetic-vs-real discrimination	[3], [22]
Mouse representation	Neuromotor (lognormal) parameters; position-difference and directional-speed series; (x, y, dt) matrices	D, B, T, Sp	Learned or model-based representations	[2], [3], [15]
Click	Click interval and duration, double-click timing, click-to-target distance and precision, pre-click movement	D, B, T, Sp	Pointing performance; absence of approach movement	[12], [14], [22]

Family	Metrics	Attributes	Typical use	Sources
Keyboard timing	Dwell/hold time, flight time, inter-key interval, typing speed, variability, bursts	D, B, T	Authentication; agent typing patterns	[12], [29], [30]
Text-entry mode	Paste versus key-by-key entry, change-event form filling, delete-and-retype	D, B, C	Agent discrimination	[12]
Navigation (log)	Request counts, bytes, methods, response-code and resource-type proportions, URL depth, sequential requests	D, B, T	Session-level bot detection	[2]
Navigation (client)	Scroll distance, duration and pattern; back/forward use; hidden-link and stylesheet-loading behaviour	D, B, T, Sp	Human-visitor evidence	[1], [12], [33]
Temporal	Inter-event time, timing entropy, reaction/decision latency, action clustering, idle periods, mouse-event rate	D, B, T	Timing regularity; model-specific timing	[2], [14], [45]
Execution-level	Event sequence, raw pointer/wheel event presence, programmatic click signatures, synthetic events	D, C, T	Automation-stack artefacts	[12], [14]
Browser/device	User-Agent, JavaScript APIs, plugins, fonts, canvas/WebGL, screen, hardware	S, C	Client identification; spoofing checks	[1], [12], [34]
Network	TLS handshake parameters (JA4), HTTP headers and order, IP/ASN	S, C	Claimed-versus-actual client mismatch	[1], [38], [46]

6. MACHINE LEARNING FOR WEB BOT AND AGENT DETECTION

6.1 Classical supervised learners

Log-based detectors have relied mainly on support vector machines [48], random forests [49], AdaBoost and multilayer perceptrons, trained on session features [2]. Logistic regression and decision trees are standard baselines and are intrinsically interpretable candidates; transparency is the usual reason for favouring such models when decisions must be justified [42]. Random forests were the best-performing classifier in all scenarios evaluated in BeCAPTCHA-Mouse, followed by an LSTM network [3]. Gradient boosting [50] and its scalable implementation XGBoost [51] have been used for agent detection from behavioural and browser features [12], and for bot classification from TLS (JA4) fingerprints [38]. Tree ensembles handle heterogeneous tabular features and, with attribution methods, offer comparatively accessible explanations (Section 7), but they do not model temporal order unless the sequence is first summarised into features.

6.2 Neural and sequence models

Convolutional networks have been applied to mouse movements encoded as matrices, with per-page predictions aggregated by majority vote over a session [2], and further image-style encodings of mouse data are surveyed in [22]. Recurrent models, namely LSTM [52] and GRU [53] networks, consume raw trajectories or timing sequences; BeCAPTCHA-Mouse compared LSTM and GRU variants with classical learners and found that the LSTM could not match the random forest when the number of training samples was small [3]. A time-series representation combining position differences and directional speed for deep sequence learning was reported to outperform hand-crafted-feature machine learning and visualisation-based deep learning on the authors' data [15]; because only abstract-level information was available to us for that study, we do not draw further conclusions. Transformer architectures [54] have been suggested as promising for temporal biometric data [18], and a transformer for tabular data was used as a baseline in the three-class agent-detection study [14].

6.3 Anomaly detection, one-class learning and generative approaches

Anomaly detection [55] avoids the need for labelled bots by modelling normal behaviour. One-class support vector machines [56], isolation forests [57] and autoencoders [58] are standard instruments. In BeCAPTCHA-Mouse, a one-class SVM trained only on real trajectories was compared with a multiclass SVM trained on real and

synthetic trajectories, and training on real samples alone detected most attacks with accuracy below 70% for both global and neuromotor features [3]. Generative adversarial networks [59] serve two roles in this literature: producing realistic synthetic trajectories for training and evaluation, and, through the discriminator, detecting synthetic samples, including those produced by other generators [3]. An earlier mobile-device study combined touchscreen and sensor behaviour with generative training for bot detection [60]. Neural data augmentation has also been used to compensate for limited labelled bot traffic [61].

6.4 Ensembles and fusion

Ensembles combine several classifiers on one signal, as in the log-based module of [2], while fusion combines signals. Decision-level fusion of a web-log module and a mouse-movement module has been studied under several rules (logical OR, probability averaging and a rule prioritising the mouse score when it is confident) [2]. Fusion with other modalities is a general theme in mouse-dynamics research [22].

6.5 Comparison and cautions

Table 3 compares these families on input, paradigm, strengths, limitations, interpretability, cost and suitability for behavioural sequences. The entries are general characteristics supported by the cited sources where indicated and by the authors' interpretation otherwise. No method is universally superior. The ranking of algorithms depends on the unit of analysis (single trajectory, session or request), the realism of bot data, the amount of training data and the evaluation protocol: the random forest led in one trajectory-level benchmark [3], a convolutional network was the choice for matrix-encoded mouse sessions [2], and gradient-boosted trees were used for tabular fingerprint and behaviour features [12], [38]. Two practical considerations cut across families. Detection thresholds depend on well-calibrated scores, and modern neural networks are often poorly calibrated [62]. Detectors also face distribution shift as bot behaviour evolves [63] (Section 10). Online detection after a small number of requests is possible in some settings [2], which bears on the latency–accuracy trade-off.

Table 3. Comparison of machine-learning methods used for behavioural bot and agent detection (general characteristics; interpretive where uncited)

Method	Typical input	Paradigm	Strengths	Limitations	Interpretability	Cost	Temporal modelling / suitability
Logistic regression	Aggregated features	Supervised	Simple baseline; coefficients readable	Linear boundary; manual features	High	Low	None unless sequences summarised
Decision tree	Aggregated features	Supervised	Rule-like paths [42]	Unstable; limited accuracy alone	High (small trees)	Low	None; suits session features
Random forest	Aggregated or neuromotor features	Supervised ensemble	Robust on tabular data; best in one trajectory benchmark [3], [49]	Needs representative bots; no sequence order	Medium; importance and SHAP	Low–medium	Via summarised features
SVM	Aggregated features	Supervised	Used in log and mouse studies [2], [3], [48]	Kernel/parameter sensitivity; scale	Low–medium	Medium	Via summarised features
Gradient boosting / XGBoost	Behavioural and fingerprint features	Supervised ensemble	Strong on heterogeneous tabular data [12], [38], [51]	Overfits small data; explanation by attribution	Medium; importance and SHAP	Medium	Via summarised features
MLP	Feature vectors	Supervised	Flexible; used in log and mouse studies [2], [3]	Opaque; calibration [62]	Low	Medium	Limited

Method	Typical input	Paradigm	Strengths	Limitations	Interpretability	Cost	Temporal modelling / suitability
CNN	Matrix-encoded movement	Supervised	Learns spatial patterns [2]	Encoding choice affects results; opaque	Low	Medium–high	Indirect via encoding
RNN / LSTM / GRU	Raw trajectories, timing sequences	Supervised	Models order directly [3], [52], [53]	Needs more data; below RF at small sample size [3]	Low	High	Strong; suits event streams
Transformer	Sequences or tabular features	Supervised	Proposed for temporal biometrics [18], [54]	Data hungry; limited evidence in this domain	Low; attention contested [64]	High	Strong in principle
Autoencoder / anomaly detection	Normal behaviour only	Unsupervised	No labelled bots needed [55], [57], [58]	Threshold choice; humanlike bots may lie within normal range	Low–medium	Medium	Possible with sequence encoders
One-class SVM	Real (human) samples	One-class	Needs only human data [56]	Below 70% against synthetic attacks in one study [3]	Low	Medium	Via features
Ensembles / fusion	Multiple modules or signals	Mixed	Combines complementary signals [2]	Rule design; correlated errors	Medium (explicit rules)	Medium–high	Depends on modules

7. EXPLAINABLE MACHINE LEARNING FOR DETECTION SYSTEMS

7.1 Why explainability matters

Explainable AI has been surveyed for cybersecurity in general [65]. For bot and agent detection the case rests on several practical needs, some documented and some our interpretation. In biometrics, black-box models raise concerns about privacy, bias and error, and explanations are expected to increase acceptance and accountability [18]. In automated access control, a blocked or challenged user may be entitled to a meaningful account of the decision in some jurisdictions [16], [17]. For analysts, a feature-level reason (for example, that a session was flagged because no pointer movement preceded clicks) supports triage and rule updates and helps expose systematic errors such as high false-positive rates for particular input devices. In fraud and security settings, explanations also help detect spurious cues, such as artefacts of the data-collection environment. A "bot probability" without reasons supports none of these uses.

7.2 Global and local explanation

Global explanations describe the model's overall behaviour (which features matter most, how outputs vary with them), whereas local explanations account for a single prediction [43], [44]. Both are relevant: global explanations support model validation and policy, and local explanations support the review of individual sessions. Methods may be intrinsic (transparent models) or post-hoc, and model-specific or model-agnostic [41], [42].

7.3 Techniques

Table 4 summarises the main techniques. SHAP assigns each feature an additive contribution to a prediction using Shapley values from cooperative game theory [66]. LIME fits a simple interpretable model in the neighbourhood of an instance to approximate a black box locally [67]. Feature importance in tree ensembles is commonly computed from split gain, as in the agent-detection study that also used SHAP [12]. Permutation importance measures the loss in performance when a feature is permuted; it was introduced for random forests [49] and later generalised to the importance of a variable across a class of well-performing models [68]. Partial dependence plots show the marginal effect of a feature on predictions [50], and individual conditional expectation (ICE) plots disaggregate this effect per instance [69]. Counterfactual explanations rely on the smallest change to an input that would affect the output [17], supplemented by methods for creating multiple counterfactuals [70], and anchors create precise if-then statements locally [71]. Surrogate models and rule learning are used to learn an interpretable model

from a complex model [43]. Attention-based interpretability of sequence models has been found controversial, because attention does not always explain sequence models accurately [64].

Table 4. XAI techniques and their characteristics for ML-based bot and agent detection

Technique	Scope	Model dependence	Output	Strengths	Limitations / security concerns	Possible use here
SHAP [66]	Local; aggregable to global	Agnostic	Additive feature contributions	Consistent attribution framework; used for agent detection [12]	Correlated features; can be manipulated [73]	Per-session reasons for analysts
LIME [67]	Local	Agnostic	Local surrogate weights	Intuitive; any classifier	Instability; manipulable [73], [74]	Spot checks of flagged sessions
Gain-based feature importance	Global	Specific (trees)	Importance scores	Cheap [12]	Biased toward some feature types; model-dependent [68]	Model validation
Permutation importance [49], [68]	Global	Agnostic	Performance drop per feature	Simple; extends to model classes [68]	Correlated features; costly	Audit of feature reliance
Partial dependence [50]	Global	Agnostic	Marginal effect curves	Visual and simple	Hides heterogeneity; correlated inputs	Effect of timing or speed features
ICE [69]	Local to global	Agnostic	Per-instance curves	Reveals heterogeneity	Visual clutter	Checking subgroup effects
Counterfactuals [17], [70]	Local	Agnostic	Minimal input change that flips outcome	Actionable; supports recourse [17]	May reveal evasion paths	Analyst "what would change this?"
Anchors [71]	Local	Agnostic	High-precision if-then rules	Readable rules with coverage	Costly; sensitive to sampling	Rule suggestions for analysts
Surrogate models / rule extraction [43]	Global or local	Agnostic	Approximating tree or rules	Human-readable	Fidelity gap to original model	Policy-level summaries
Attention weights [64], [72]	Local	Specific (attention models)	Weights over inputs	Built into sequence models	Faithfulness contested	Cautious inspection of sequence models

7.4 Explainability in the bot and agent detection literature

Explanation has so far been largely a by-product rather than a design goal in this literature. The agent-fingerprinting study uses both gain-based feature importance and SHAP to interpret an XGBoost classifier trained on behavioural and browser features [12]. A TLS-based bot classifier reports feature-significance analyses identifying handshake-derived features as most influential [38]. The minimal-feature-set study achieves a form of transparency by exhaustive search over small feature subsets, showing that two behavioural features isolate agents in its benchmark [14]. The mouse-dynamics survey provides explicit mathematical definitions of features [22], and the neuromotor features of BeCAPTCHA-Mouse have a kinematic interpretation through the lognormal model [3], [26]; in our interpretation, such feature sets are interpretable by construction at the feature level even when the downstream classifier is not. Pixel-like or raw-

sequence representations fed to convolutional or recurrent networks [2], [15] are less directly interpretable. The decision-level fusion rule of Iliou et al. is transparent in itself, since the roles of each module are explicit [2]. Across the explainable-biometrics literature, XAI was applied mainly to face-based systems, with behavioural biometrics considered by only two of the 47 selected studies [18].

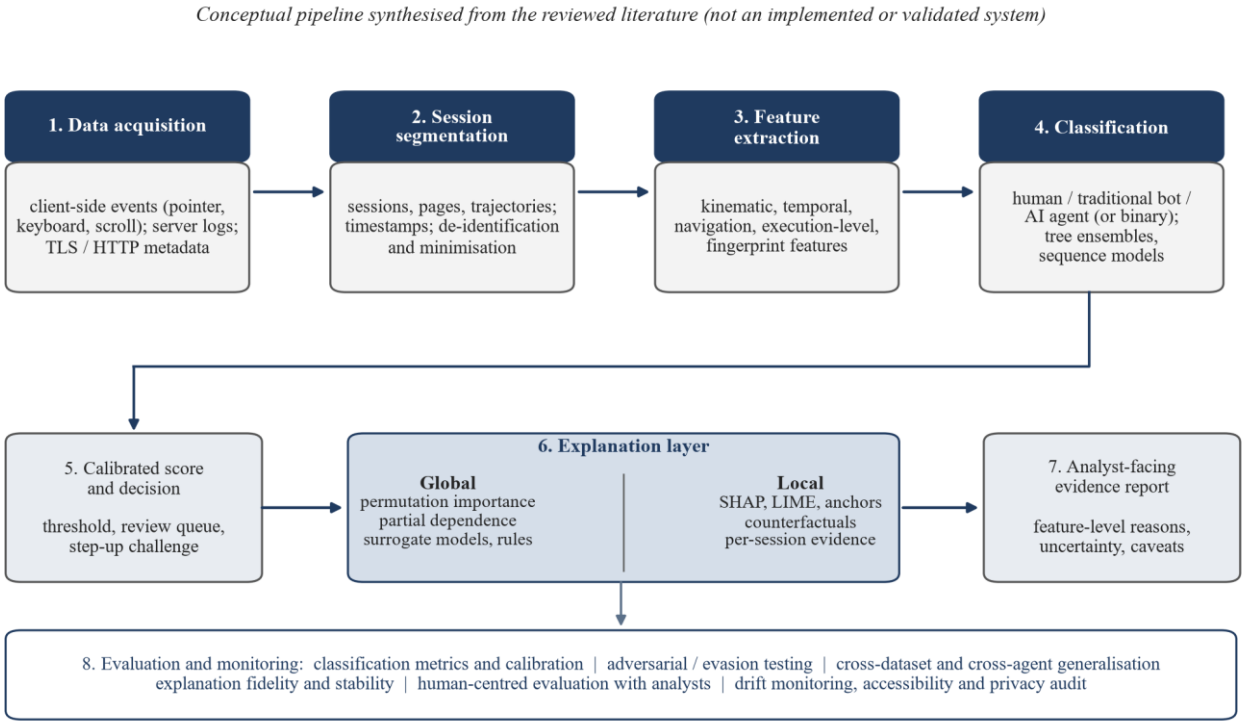


Figure 4. Conceptual explainable ML pipeline for behavioural agent detection, synthesised from the reviewed literature. It is not an implemented or validated system.

7.5 Evaluating explanations

Quality of explanation is multi-faceted in nature. Fidelity refers to the faithfulness of the explanation to the model. The aspects of stability and consistency relate to whether similar inputs and equally accurate models would be given similar explanations. Comprehensibility is about understanding the explanation by the target user. A widely cited taxonomy distinguishes application-grounded evaluation with domain experts on real tasks, human-grounded evaluation with lay users on simplified tasks, and functionally grounded evaluation using proxy measures without humans [75]. Frameworks for measuring explanation goodness, user satisfaction, mental models and trust have been proposed [76], and social-science research indicates that human explanations are contrastive and selective [39].

The explainable-biometrics SLR shows how far practice lags these frameworks. All selected studies used model-centric metrics, visualisation was the most common (44 studies, seven of which used it alone), three studies performed no evaluation, none specified the target user of the explanation and none adopted user-centric metrics of understandability [18]. We found no study in the bot- or agent-detection set that evaluated explanations with security analysts, which we treat as a gap (Section 11).

7.6 Bias, fairness and security implications

Explanation approaches themselves are also potential points of attack and error. Discriminatory classifiers can be created that behave like they do without producing any problematic LIME or SHAP explanations [73], and neural network explainability may be brittle to small changes [74]. Importances can also vary for equally good models [68]; thus, a list of importances cannot be trusted too literally. While explanations may help in exposing that a classifier is using disability or device type indicators, such explanations do not address the issue directly. In addition to the problems with explanations above in the context of detection, our explanation approach faces another challenge, namely that disclosure of details of an explanation to the user might assist an adaptive adversary in avoiding critical indicators. The tradeoff between leaking useful explanation to the analyst and unnecessary information to the adversary needs further exploration empirically, and there does not seem to be reviewed research addressing the problem.

8. COMPARATIVE ANALYSIS OF EXISTING DETECTION APPROACHES

The analysis of the evidences in this section is thematic rather than analytical. The sections 8.1 to 8.4 analyze the

behavioral biometric, automated browsing, multi-signal, and adversarial evidences, respectively. Section 8.5 has the comparative tables.

8.1 Behavioural biometrics and mouse dynamics as discriminators

Indeed, the biometric-behavioural research provides a possible rationale behind the separation of humans from machines, since human point movements involve the use of motor control mechanisms, which might not be present in synthetic movements [3], [26]. Within this line of research, there are several points of intersection. First, complex descriptors contain more information than simple ones: BeCAPTCHA Mouse is based on the notion that previously there was a usage of primitive cues like time or average speed and provides evidence of the 93% average accuracy of detecting high realism bots using neuromotor information about the whole movement and combining synthetic/real training sets [3]. Second, the representation of the raw signal is itself a design choice: hand-crafted features [3], [22], image-like matrices [2] and time-series encodings [15] have all been used, and the comparative claims of each study are specific to its data. Thirdly, there is no consistent unit of decision: a single path between two clicks [3]; a matrix of pages generated via voting in the course of the session [2]; or a session/trial of many events [12], [14]. Fourth, how realistic the training bots are determines how well a detector generalises: in BeCAPTCHA-Mouse, classifiers trained only on real trajectories detected most attacks at below 70% accuracy, whereas training with synthetic samples raised accuracy substantially [3]. The practical implication, in our interpretation, is that evaluations against a single family of simple bots overstate real-world robustness.

Web-log studies show a complementary pattern. Session features computed from logs separate simple bots from humans well, but earlier work summarised by Iliou et al. reports lower performance as bots become more sophisticated [2], and mouse-based detectors evaluated on moderate and advanced bots were reported with highly variable results across studies [2]. In Iliou et al.'s own testbed, the mouse module outperformed the log module for both moderate and advanced bots, which the authors attribute to humanlike mouse movement being harder to simulate than humanlike page-visiting behaviour [2]. This is a finding about their bots and their testbed (static pages, a small participant pool) and not a general ranking.

8.2 Automated browsing, fingerprints and identity spoofing

The honeysite approach of Li et al. addresses the ground-truth problem directly. By registering new domains and never advertising them, any traffic received is attributable to bots and their operators, which sidesteps manual labelling and the circularity of needing a detector to build a detector's dataset [1]. Over seven months, 100 honeysites running five open-source web applications recorded 26.4 million requests from more than 287,000 IP addresses [1]. Comparing declared identity with the TLS handshake and HTTP headers, the study found that 86.2% of the 30,233 clients claiming to be Chrome or Firefox were not what they claimed, mostly matching common Go and Python HTTP libraries and command-line tools; of the 18,085 addresses claiming to be Firefox, 98.5% did [1]. The general lesson is that a claimed browser identity need not correspond to the underlying automation technology, and that lower-layer signals that the client does not deliberately manipulate can expose it.

This lesson has limits for AI agents. Li et al.'s approach assumes that crawlers do not forge a TLS fingerprint of another tool and that the observed bot population mostly lacks JavaScript execution [1]. Agents that drive real browsers present genuine browser TLS stacks [12], so the discriminating information must come from elsewhere. Recent work reports that browser fingerprints shared across agents discriminate poorly while behavioural fingerprints discriminate well [12], and that multi-layer fingerprints, which combine network, HTTP and browser-interaction behaviour, separate agent frameworks from humans, legacy crawlers and one another [13], [46]. Fingerprinting and behavioural metrics are therefore complementary rather than competing: static fingerprints are cheap and effective against naive clients, whereas behavioural and execution-level signals address clients whose static presentation is browser-like.

8.3 Multi-signal detection

Iliou et al. extended an earlier log-based framework for advanced bots [77] by adding a mouse module and combining the two by decision-level fusion that relies on the mouse score when it is confident and averages the two scores otherwise. In their first-phase experiments, logical OR fusion was less effective than the mouse module alone because errors from the weaker module propagate, averaging masked individual errors, and the prioritised fusion classified all visitors in their test set correctly; none of the tested bots bypassed both modules, though some bypassed one [2]. They also note that the framework does not defend against replay of recorded human behaviour and relies on the visitor running the collection script [2]. The agents have been studied [14] to realize that their discriminatory signal is represented in several different attributes. Multi-layer fingerprinting is inherently an integration of several layers [13], [46]. In our view, the multi-signal detection will increase the cost of evasion because the attacker will have to deceive on several signals. This observation has been derived based on small-scale testbeds.

8.4 Adversarial and evasive behaviour

There are different ways in which automated clients try to pretend themselves as human users. Humanizing of mouse

trajectories using function-based and GAN-based synthetic trajectory generation, and reinforcement learning that produces mouse trajectories to trick a score-based CAPTCHA [3], [5] is one of the techniques among many others. Timing randomisation and behaviour scripting include sleeps scaled to text length, crawling at particular times and spending time on revisited content [2]. Evasion by learning has been explored through bots trained against detectors using generative adversarial networks and deep reinforcement learning [78], [79]. Fingerprint and identity spoofing include claiming arbitrary User-Agents [1] and configuring or replacing automation tools to alter fingerprints [2], [19]. CAPTCHA solving by machine learning has been demonstrated for image-based challenges [4], and an empirical study of deployed CAPTCHAs examined user solving performance and perception in the context of a long-running arms race [6]. Behavioural imitation by agents is considered as a problem to be solved using the minimax game between the detector and the agent [47], and some web agents reportedly avoid all known ways of detection of bots [13]. We were unable to locate any study proving that the TLS handshake process could be manipulated by an AI agent; all known TLS-based solutions assume the opposite [1].

In general settings, adversarial machine learning is an evidence for the vulnerability of all learned detectors to input manipulation [80], while concept drift means the degradation in performance of any learned model because of change in behavior [63]. All detectors using only one signal are vulnerable to attacks due to the fact that the attack surface of this detector is limited only by this single signal, which any adversary can learn or emulate, such as fingerprints that can be manipulated [2] or trajectory that can be simulated artificially [3].

8.5 Comparative tables

The table below (Table 5a) is the comparison between the articles according to the dimensions required for the review, while Table 5b is the comparison of detection layers. The information given below is the one stated in the sources; "Not discussed" and "Not evaluated" mean lack of information. Information regarding the performance is omitted intentionally due to the different evaluation conditions (Section 9).

Table 5a. Comparison of existing studies on bot and agent detection (P = peer-reviewed; PP = preprint)

Study	Year	Detection target	Data source	Behavioural features	ML method	XAI	Main finding	Limitation
Park et al. [33] (P)	2006	Web robots	Web server traffic	Mouse/keyboard events, stylesheet loading, invisible links (as summarised in [1])	Not discussed here	Not discussed	Implicit human browsing patterns used to infer robots	Full text not analysed
Chu et al. [32] (P)	2013	Blog bots	Blog site data	Mouse and keystroke patterns	Not discussed here	Not discussed	Behavioural biometrics applied to blog-bot detection	Detailed results not analysed
Li et al. [1] (P)	2021	Malicious and benign bots; identity spoofing	100 honeysites, 26.4 M requests	Request behaviour; TLS and header fingerprints; JS fingerprints	Fingerprint matching (not ML classification)	Not applicable	Most claimed-browser clients mismatched TLS/header fingerprints; honeysites give ground truth	Five web applications; most bots lack JS; targeted attacks missed
Iliou et al. [2] (P)	2021	Moderate and advanced web bots	Testbed web logs and mouse movements	Log features; (x, y, dt) mouse matrices	Ensemble of classifiers; CNN; decision-level fusion	Not discussed	Mouse module more effective than logs; fusion robust in testbed	Small testbed; no replay defence; needs JS

Study	Year	Detection target	Data source	Behavioural features	ML method	XAI	Main finding	Limitation
Acien et al. [3] (P)	2022	Mouse-based bot detection	BeCAPTCHA-Mouse benchmark (58 users)	Neuromotor (lognormal) features; global features	SVM, RF, KNN, MLP, LSTM, GRU; GAN discriminator	Not evaluated	Real+synthetic training improves detection of realistic bots	Controlled task; self-generated bots
Niu et al. [15] (P)	2021	Web bots (mouse-based)	Not analysed in full text	Position differences and directional speed series	Deep sequence learning	Not evaluated	Reported to outperform handcrafted and visual-encoding baselines	Abstract-level information only
Wang et al. [12] (PP)	2026	Seven AI browsing agents vs humans	Instrumented honey website; 7,000 agent trials, 546 human instances	Typing, scrolling, mouse (50 features); browser features (418)	XGBoost multi-class	Gain importance; SHAP	Behavioural fingerprints distinctive; browser fingerprints limited	Seven agents; student humans; held-out-task drop
Choudhary et al. [14] (PP)	2026	Human / bot / AI agent	Controlled benchmark with evasion ladder	Mouse-event rate, teleport-click ratio, others	MLP, SAINT, GBMs	Feature-subset search	Binary labels misroute agents; two features isolate agents	Signal is an automation artefact; controlled data
Kang et al. [46] (PP)	2026	Six web-agent frameworks	Live instrumented domain	Browser actions plus network layer	Decision tree	Not discussed	Multi-layer fingerprints separate agents	Six frameworks; single deployment
Fayolle et al. [13] (PP)	2026	Six LLM web agents	Honeysites with anti-bot mechanisms	Network, HTTP, browser fingerprints	Fingerprint analysis	Not discussed	Some agents bypass all anti-bot mechanisms; stealth can raise detectability	Limited agent set
Lugoloobi et al. [45] (PP)	2026	Identity of LLM behind agent	14 LLMs, 4 environments	Action and timing traces	Classifiers on traces	Not discussed	Model identifiable from actions; delays only partly protect	Synthetic web environments
Zhu et al. [47] (PP)	2026	Mobile GUI agents	New touch-dynamics dataset	Touch kinematics	Detector vs. humanisation methods	Not discussed	Vanilla agents detectable; humanisation feasible	Mobile setting
Jarad and Bicakci [38] (PP)	2026	Web bad bots	JA4 TLS fingerprint database	None (network layer)	XGBoost, CatBoost	Feature importance	JA4 components informative for bot classification	Database labels; not agent-specific
Khan et al. [22] (P)	2024	User authentication (survey)	123 papers surveyed	Mouse and widget features	Statistical, ML and DL (survey)	Not evaluated	Taxonomy, datasets, challenges	Not bot-specific

Study	Year	Detection target	Data source	Behavioural features	ML method	XAI	Main finding	Limitation
Tucci et al. [18] (P)	2024	Explainable biometrics (SLR)	47 of 496 papers	Not applicable	CNNs dominant	Grad-CAM most used	Model-centric explanation evaluation only	Few behavioural cues

Table 5b. Comparison of detection layers

Detection layer	Signals	Examples	Advantages	Limitations
Challenge-response	Solved tasks; risk scores	CAPTCHA; score-based analysis [2]	Direct human verification	Solvable by ML [4], [6]; accessibility barriers [81]
Browser / fingerprint	User-Agent, APIs, canvas, fonts	FingerprintJS-style collection [1], [12]	Cheap; effective against naive clients	Spoofable [2], [19]; limited when shared [12]
Network	TLS handshake, headers, IP	JA4/TLS matching [1], [38]	Exposes claimed-vs-actual client	Weaker for real-browser agents; proxies [1]
Web-log	Request counts, types, codes	Session classifiers [2]	Server-side; no client script	Degrades against advanced bots [2]
Timing	Inter-event time, latency	Timing traces [45]	Hard to fix in advance	Randomisation can mask [2], [45]
Mouse trajectory	Kinematics, shape	Neuromotor features [3]	Rooted in motor control	Device variability [22]; synthesis [3]
Keyboard	Dwell, flight, entry mode	Typing features [12]	Distinguishes entry methods	Sparse for non-text tasks; privacy
Navigation	URL sequence, scroll	Scroll, link behaviour [12], [33]	Reflects task strategy	Task-dependent
Execution behaviour	Event streams, synthetic events	Absence signature [14]	Hard to forge by trajectory synthesis	Stack-dependent; may change
ML classification	Feature vectors, sequences	RF, XGBoost, CNN, LSTM [2], [3], [12]	Learns combinations	Opacity; drift [63]
XAI interpretation	Attributions, rules, counterfactuals	SHAP, importance [12]	Accountability, analyst triage	Instability [73]; leakage; rarely user-tested [18]

9. DATASETS, FEATURES AND EVALUATION METRICS

9.1 Available datasets and benchmarks

Table 6 summarises datasets associated with the reviewed studies. Public datasets for mouse dynamics exist but are less common than for keystroke dynamics, and the lack of standardised data collection makes comparison difficult [22]. The mouse-dynamics survey lists, among others, the Chaoshen, ISOT, Balabit, DFL, BB-MAS, TWOS and Minecraft datasets, which differ in task regime (fixed static sequences versus free or app-restricted use) and population [22]. These were collected mostly for user authentication rather than bot detection. For bot detection, BeCAPTCHA-Mouse introduced a public benchmark of 15,000 trajectories: real trajectories from 58 users who each repeated a task, plus function-based and GAN-generated synthetic trajectories of varying realism [3]. Keystroke dynamics has a long-standing shared benchmark for comparing detectors [30].

Datasets for AI agents are new and small relative to the human-bot corpora. They consist of controlled collections on instrumented websites: seven agents with 1,000 trials each against 546 human instances from 56 undergraduate participants [12]; a labelled corpus of agent traces across fourteen language models and four web environments

[45]; and a benchmark with evasion sessions replaying human cursor data and GAN-generated trajectories [14]. Honeysite collections observe agent frameworks and human baselines under several anti-bot mechanisms [13], [46], and a mobile touch-dynamics dataset supports agent humanisation research [47]. For network signals, a TLS-fingerprint database was used for bot classification [38]. We did not verify the public availability of each resource unless the source states it.

Table 6. Datasets and evaluation methodologies associated with the reviewed studies (as reported by the sources; availability not verified unless stated)

Dataset / study	Modality	Scale and participants	Bots / agents / synthetic	Ground truth and protocol	Limitations
BeCAPTCHA-Mouse benchmark [3]	Mouse trajectories	15,000 trajectories; real data from 58 users	Function-based and GAN synthetic bots	Labels by construction; 70/30 splits per classifier	Controlled task; bots from authors' generators
Iliou testbed [2]	Web logs and mouse	Authors' sessions; then 28 users, 56 sessions	Moderate and advanced bots (scripted)	User-independent folds in second phase	Static pages; small sample
Aristaeus honeysites [1]	Logs; TLS; JS fingerprints	100 sites; 26.4 M requests; 7 months	Real bot traffic (benign and malicious)	Ground truth by design (no human traffic)	Five web apps; no AI-agent focus
Mouse datasets surveyed [22]	Mouse	Chaoshen, ISOT, Balabit, DFL, BB-MAS, TWOS, Minecraft	Mostly none (authentication)	Task regimes differ	Not bot-oriented; poor standardisation
Keystroke benchmark [30]	Keystroke timing	Shared benchmark dataset	Imposters (not bots)	Detector comparison protocol	Authentication, not agents
FP-Agent data [12]	Browser and behavioural	7,000 agent trials; 546 human instances (56 students)	Seven AI agents	80/20 split; held-out-task test	Imbalance; narrow humans
Agent trace corpus [45]	Actions and timings	14 LLMs; 4 environments	LLM agents	Labelled corpus released (stated)	Synthetic environments
Three-class benchmark [14]	Behavioural features	Includes 2,299 evasion sessions	Agents; GAN and replay evasion	Multiple seeds and model families	Controlled; automation artefact
Mobile touch data [47]	Touch dynamics	New dataset (size not analysed)	Mobile GUI agents	Detector-agent minimax design	Mobile only
JA4 database [38]	TLS fingerprints	Database of real fingerprints	Bots (not agent-specific)	Held-out test set	Label provenance

9.2 Dataset characteristics and limitations

Several recurring properties limit inference. Human participants are often small or non-representative: the first phase of Iliou et al.'s evaluation used the authors themselves, and the second used 28 additional participants producing 56 sessions [2]; the agent measurement study used undergraduate students only [12]. Bots and agents are usually created by the researchers themselves, such as trajectories generated by their own generator system [3] or scripts for moderate and complex bots [2] or limited agents doing some specific jobs [12]. Class balance varies: some datasets equalise human and bot sessions [2], whereas the agent study has several thousand agent trials against hundreds of human instances [12]. The presence of ground truth is guaranteed by definition for honeysites [1] and by design for laboratory experiments which do not consider the variability of traffic. Traffic that is tagged with ground truth irrespective of whether it belongs to AI agents or not, is not a part of the set we are interested in. Availability of benchmarks and harnesses [3], [45] facilitates reproducibility whereas lack of data sharing and task specific nature hinder reproducibility. Cross-device variation in screen, input device and sampling affects mouse data and has been little investigated [22].

9.3 Evaluation metrics

Accuracy is the metric that is mentioned most often, but it provides no meaningful information in the case of imbalance among classes. Other frequently used metrics are precision, recall, and F1 score. The choice of the positive class, and the per-class computation of metrics, are two important features. ROC-AUC evaluates ranking quality and was used in BeCAPTCHA-Mouse [3]. False-positive and false-negative rates have asymmetric costs in deployment: a false positive denies or burdens a legitimate user, which has accessibility implications, and a false negative admits automation. Detection latency matters for online use; Iliou et al. evaluate accuracy as a function of the number of requests seen [2]. Calibration determines whether scores can be used as probabilities for thresholding [62]. Robustness is assessed through evasion sessions, synthetic attackers or injected delays [3], [14], [45]. Cross-dataset generalisation is rarely assessed; in the agent study, a held-out-task experiment reduced the behavioural classifier's F1 by up to 0.369 [12], and the mouse-dynamics survey calls for benchmarking on public datasets [22]. Explanation-specific measures are discussed in Section 7.

9.4 Comparability

Numerical results should not be ranked across studies. They differ in the unit of analysis (trajectory, page, session, trial), in the realism and type of bot or agent, in the human populations, in the splitting protocol (for example user-independent folds in [2] versus random splits elsewhere), in label spaces (binary versus multi-class) and in what is predicted (human versus bot, agent identity, or model family). In this review we therefore report within-study figures only where they illustrate a finding, and never as a ranking.

10. CHALLENGES AND LIMITATIONS

Behavioural detection is promising but not risk-free. This section critically analyses the limitations that recur in the reviewed literature; the interpretations offered are the authors' own unless cited.

10.1 Data, ground truth and bias

High-quality datasets are scarce. Establishing ground truth about bots is very difficult due to circularity [1]; humans subjects are either small in number or selected from a biased population [2], [12]; artificial bots or agents are colored with assumptions of their creators [3]. The finding of one experiment may not be applicable in another case; for instance, Iliou et al.'s research reveals that bots can be detected more easily because of their homogeneity in contrast to that of humans in a heterogeneous human population. Dataset bias therefore affects not only accuracy but the apparent difficulty of the task.

10.2 Human variability and device diversity

Mouse behaviour varies within and between people, and the survey of mouse dynamics highlights intra-subject and inter-subject variability, the effect of different computers, screen resolutions and mice, the absence of investigation into interoperability, and the need to consider trackpads and touch devices [22]. Fingerprint drift over time also complicates static signals [35]. Models trained on one device population may therefore misjudge another.

10.3 Accessibility

Challenge-response tests have long been criticised as barriers for people with disabilities [2], [3], [81], and behavioural detection is often proposed precisely to avoid such interruption [2]. It is not, however, free of accessibility risk. In our interpretation, users of screen readers, switch devices, eye trackers, voice control, keyboard-only navigation or tremor-affected pointing may generate event patterns that differ from the populations used to train detectors, and some assistive technologies emit programmatic or non-continuous input that resembles the execution-level signals discussed in Section 5. The reviewed studies do not evaluate detectors with assistive-technology users, and their human samples are limited, for example to undergraduate students [12]. We regard accessibility testing as a prerequisite for any deployment, not an optional extension.

10.4 Privacy and regulation

Collecting pointer and keystroke streams raises privacy concerns because such data can in principle be used to characterise individuals; the CAPTCHA literature already raises concerns about how sensitive information is processed under data-protection law [3]. Under the European General Data Protection Regulation, biometric data is defined as personal data resulting from specific technical processing of physical, physiological or behavioural characteristics that allows or confirms the unique identification of a person, and automated decisions with significant effects are subject to specific safeguards [16]. Whether the detection of non-personalized behavior bot falls under these clauses would be determined by the design and jurisdiction; we do not provide legal advice, but it is our position that aspects like data minimization, feature extraction on the device, and data retention are a matter of design and have never been considered in any of the aforementioned studies on detection. The explanation of behavioral features could itself reveal sensitive information about the user.

10.5 Adversarial adaptation, drift and false outcomes

The behavioral characteristics of AI agents are also subject to evolution, and, as one of the limitations pointed out in the agent study paper, these could be modified [12]. Detection is a dynamic process, where adversaries may

evolve too [47], [80], and adaptation of bots is listed among future work items by the authors of the advanced bot paper [2]. Concept drift [63] is a term meaning that if a particular model was applied for a specific dataset at a certain point in time, it will become outdated with time. False positives increase stress on the legal users, particularly on those whose behavior is unusual, while false negatives allow automation to slip by; the optimal balance between these two depends on the field and is seldom mentioned in scientific papers in terms of the cost aspect.

10.6 Explainability limitations

Explanation is always an approximation. There are many ways to manipulate post hoc procedures [73], [74]; there are different importance orders for equally good models [68], and attention can be a bad explanation under some interpretations [64]. Evaluating explanations with the people who use them is rare [18]. A faithful explanation of a flawed model can also entrench its flaws if it is mistaken for a justification.

10.7 Computational overhead and real-time constraints

Collecting and processing every visitor's mouse movements is resource-demanding, and Iliou et al. suggest restricting it to the first requests of a session, where their framework already attained high accuracy in the reported experiments [2]. Real-time use also constrains model complexity and explanation generation, since model-agnostic explainers such as LIME and SHAP require repeated model evaluation [66], [67]; the latency of explanation in production detection is not examined in the reviewed studies.

11. RESEARCH GAPS

The following gaps are identified by comparing what the reviewed studies report with what the central review question requires. Each gap is tied to evidence in the literature; where evidence is partial this is stated. Because the review is not exhaustive (Section 14), a gap means "not found in the reviewed set" and not "demonstrably absent from all research".

Gaps concerning the object of study are the most strongly supported. Work that targets autonomous AI browsing agents specifically appears almost entirely as 2026 preprints [12]–[14], [45]–[47]; none of the core peer-reviewed bot-detection studies addressed agents [1]–[3], [15] (G1). Behavioural detectors in the core set pose a binary human-versus-bot problem [2], [3], [15], and only one reviewed study formulates an explicit three-class human / bot / agent problem [14], so G2 and G3 are supported, although the latter rests on a single preprint. Another agent study distinguishes agents from humans and from one another but does not include traditional bots as a class [12].

Gaps concerning evidence and benchmarks follow. The study on mouse dynamics highlights the absence of benchmarking on public datasets [22], while the systematic literature review on explainable biometrics highlights the absence of standardized metrics and benchmarks for explanation [18] (G4). Agent datasets are artificial and much smaller than the actual traffic [12], [45] (G5). The generalization between agents and tasks appears contradictory because although one study shows that classifiers generalize the identities of agents irrespective of size and model [45], the other denies that behavioral classifiers generalize between tasks [12] (G8; partial support). Adversarial evaluation exists in several forms (synthetic trajectories, evasion ladders, injected delays) [3], [14], [45], but adaptive adversaries and replay attacks remain future work [2] (G9; partially supported). Longitudinal studies of changing agent behaviour were not found, and one study explicitly cautions that updates may alter fingerprints [12] (G12).

Gaps concerning explainability are central to this review. XAI in biometrics concentrates on faces and rarely examines behavioural cues [18], and in the bot- and agent-detection studies explanation is a by-product of feature importance or SHAP rather than a design objective [12], [38] (G6). Explanations were not evaluated with security analysts in the reviewed studies, and the biometrics SLR found no user-centric evaluation at all [18] (G7).

Gaps concerning responsible deployment are supported by absence in the reviewed set and by literature on the harms of challenge-response systems. No reviewed detector was evaluated with assistive-technology users, although CAPTCHAs are criticised for accessibility barriers [2], [81] and human samples were narrow [12] (G10). Privacy-preserving behavioural detection is not studied in the reviewed detection papers, though privacy concerns about bot-detection data are noted [3] (G11). Finally, fusion of behavioural, execution-level, browser and network signals is emerging [2], [13], [46] but has not been combined with systematic explanation and analyst evaluation (G13; partially supported). Table 7 lists each gap with its evidence and implication, and Figure 5 relates coverage to gaps.

Table 7. Challenges and research gaps identified from the reviewed literature (labels G1–G13 correspond to Figure 5)

Challenge / gap	Evidence in the reviewed literature	Implication
G1 Few AI-agent studies	Agent work is mostly 2026 preprints [12]–[14], [45], [46]	Replicate and extend with peer-reviewed studies
G2 Binary formulations	Core detectors are human-vs-bot [2], [3], [15]	Evaluate multi-class label spaces
G3 Little three-class work	One explicit three-class study [14]	Test on broader data
G4 No standard benchmarks	Benchmarking lacking [22]; explanation benchmarks lacking [18]	Community datasets and protocols
G5 Small agent datasets	Instrumented, controlled collections [12], [45]	Real, verified-label traffic
G6 XAI as by-product	SHAP or importance as secondary analysis [12], [38]	Design explanation in from the start
G7 No analyst evaluation	No user-centric explanation evaluation [18]	Application-grounded studies [75]
G8 Generalisation	Mixed: cross-family success [45], task-shift drop [12]	Cross-task and cross-agent protocols
G9 Adaptive adversaries	Fixed attackers; replay and adaptation left to future work [2]	Adaptive adversarial evaluation
G10 Accessibility	CAPTCHA barriers [81]; narrow human samples [12]	Assistive-technology testing
G11 Privacy	Concerns noted [3]; no privacy-preserving detectors found	On-device and federated designs
G12 Longitudinal drift	Updates may alter fingerprints [12]; no longitudinal study found	Longitudinal data collection [63]
G13 Signal fusion with XAI	Multi-layer work emerging [13], [46]; not combined with analyst evaluation	Interpretable fusion architectures

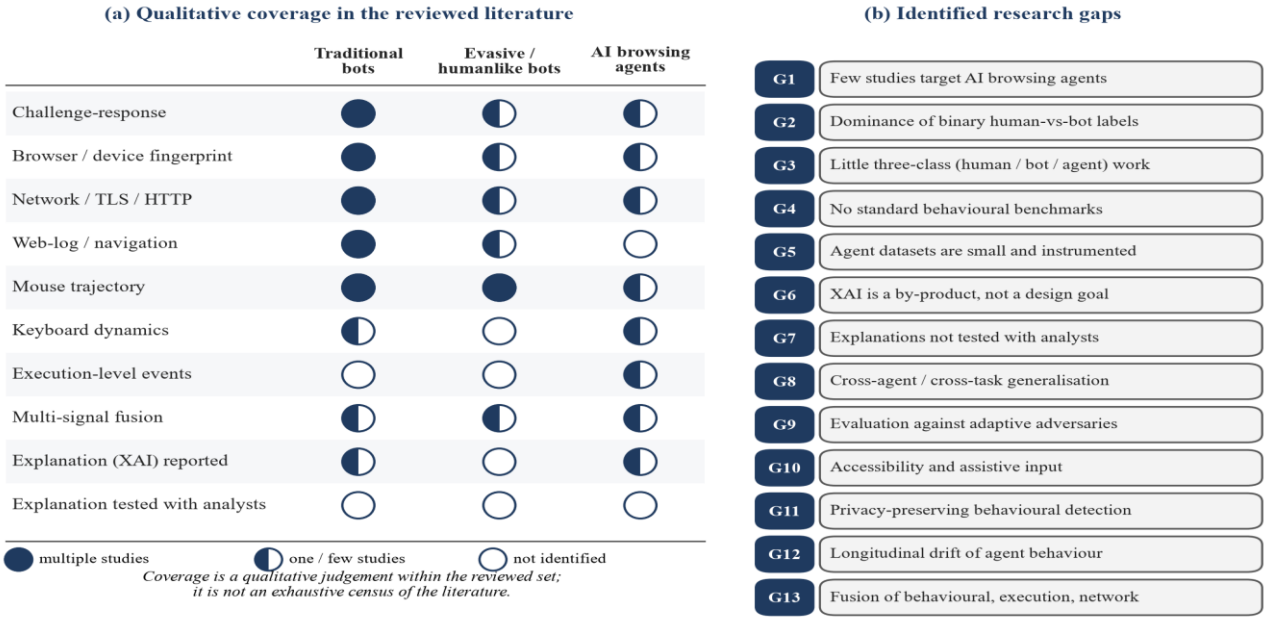


Figure 5. Research landscape and identified gaps: (a) qualitative coverage of signal layers across detection targets within the reviewed set; (b) research gaps G1–G13 (Section 11).

12. FUTURE RESEARCH DIRECTIONS

The directions in this section are research proposals that follow from the gaps in Section 11. They are not established findings, and none has been validated by this review.

Multi-modal behavioural detection and signal fusion. Combining pointer, keyboard, scroll, execution-level, fingerprint and network evidence is supported by emerging results [2], [46]. A proposal is to study fusion architectures that remain interpretable at the signal level, for example decision-level fusion with explicit confidence rules [2], and to measure how much each layer contributes against adaptive adversaries.

Human / bot / AI-agent classification. Three-class methods should be tested on larger data sets with various types of bots, stealth bots, and different classes of agents based on open-set recognition of unknown agents. The utility of three-class methods as opposed to binary classification methods using agent-based features is an empirical question [14].

Explainable and real-time XAI. Constrained models as well as global methods such as permutation feature importance [68] and local methods such as SHAP [66] can be benchmarked according to their faithfulness, robustness, and latency with respect to the streaming detection task. The idea of counterfactuals [17], [70] can be applied for description of the expected behavior of humans.

Robustness against synthetic and adaptive behaviour. It must include aspects of trajectory synthesis [3], evasion learning [78], [79], replaying [2] and humanization of the agent [47]. Adversarial training must be included when necessary [22].

Privacy-preserving and federated approaches. Feature extraction and minimization and federated learning approaches [82] could enable training of behavioral models without central aggregation of the actual interaction streams; the point is whether such an approach maintains accuracy and explainability.

Continual and drift-aware learning. Detection of distribution shift and continual updating [63] should be coupled with longitudinal data collection on how agent behaviour changes across versions.

Benchmarks and cross-agent evaluation. Standardised datasets with documented participants, tasks, bot and agent types and labelling procedures, and protocols for cross-dataset, cross-agent and cross-task evaluation, would address the benchmarking gap [18], [22].

Human-centred evaluation and accessibility. Studies with security analysts should assess whether explanations improve triage, using application-grounded designs [75] and established explanation-measurement frameworks [76]. Accessibility-aware detection requires evaluation with users of assistive technology and fairness analysis of false-positive rates.

Production-scale evaluation. Honeysite-style Deployment [1] could potentially be adapted to measure the movement of agents in verifiable labels, as long as it passes ethical approval.

Table 8 summarises the proposals, their rationale and the open questions they raise.

Table 8. Proposed future research directions (proposals, not established findings)

Direction (proposal)	Rationale from the literature	Open questions
Multi-modal behavioural detection and fusion	Complementary signals and robust fusion [2], [46]	Which layers matter against adaptive adversaries?
Human / bot / AI-agent classification	Binary labels misroute agents [14]	Open-set handling of unseen agents
Explainable and real-time XAI	XAI by-product; costly explainers [12], [66]	Fidelity and latency in streaming
Counterfactual explanations	Recourse and analyst reasoning [17], [70]	Avoiding disclosure of evasion paths
Robustness to synthetic behaviour	Synthesis and humanisation [3], [47]	Adversarial training; replay defence
Privacy-preserving and federated learning	Privacy concerns [3]; federated learning [82]	Accuracy and explanation under constraints
Continual and drift-aware learning	Drift [63]; agent updates [12]	Update policy and monitoring

Direction (proposal)	Rationale from the literature	Open questions
Cross-agent benchmarking and standard datasets	Benchmark gap [18], [22]	Shared tasks and labelling
Human-centred XAI evaluation	No user-centric evaluation [18]	Do explanations improve triage?
Accessibility-aware detection	Accessibility barriers [81]	False-positive rates for assistive input
Production-scale evaluation	Honeysite approach [1]	Ethics and label verification

13. DISCUSSION

In this section, the information is put together to address the questions asked in the literature review. The repetition of the previously mentioned information is done if it is needed to draw conclusions.

13.1 What recurs across the literature

Four families of signal recur: server-side request behaviour from web logs [1], [2], [33]; client-side pointer movement [2], [3], [15], [22]; browser and network fingerprints [1], [34], [38]; and, more recently, keyboard, scroll and execution-level traces [12], [14], [45]. Mouse kinematics and timing have received by far the most attention, in part because they have a long history in authentication research [22]. Keyboard and execution-level features are better represented in the recent agent-focused preprints than in earlier bot-detection work.

13.2 Static fingerprints versus behavioural signals

Static fingerprints are inexpensive to collect and effective against clients that do not conceal their nature: most clients that claimed to be mainstream browsers in the honeysite study were exposed by their TLS and header fingerprints [1]. Their limitations are that they can be changed or replaced by real browsers [2], depend on JavaScript execution [1], drift over time [35], and discriminate poorly when several agents share a browser environment [12]. Behavioural signals are harder to imitate and carry information about how the actor operates [3], [12], but have their own limitations: human variability and device dependence [22], small and non-representative datasets [2], [12], susceptibility to synthesis and humanisation [3], [47], and the accessibility and privacy concerns discussed in Section 10. They work together, and that is the reason why multi-channel detection assumes importance; even if the adversary is able to spoof one channel, he will have to spoof the other channels as well.

13.3 Where machine learning helps and where it obscures

Machine learning enhances detector effectiveness by way of feature interaction that cannot be captured in rules, processing data sequences, and generation of synthetic data for predicting attacks [2], [3], [61]. Opacity of machine learning results from capacity, where representations learned are devoid of any readable meaning to humans [2], [15] and usage of training data which does not represent bots [15]. Machine learning effects on its detectors under adversarial situations remain unclear [63], [80].

13.4 What XAI contributes and what it does not solve

The possibility to give feature-level explanation allows for validation and error analysis [18], [65]. For detection research, it additionally helps in showing that the discriminating features are the artifacts of automation rather than the artifacts of intelligence [14], which is an important point for determining the time period during which the detector remains valid. However, explainable AI does not replace the need for good data and a good evaluation methodology since explanations can be unstable and even manipulated [73], [74], dependent on data and models [68], and not often validated by its users [18], while the explanation may be used as a leakage of information for the adversary (see Section 7). From our perspective, the optimal solution for such a contradiction lies in using meaningful by design feature sets [3], [22] and treating explanations of machine's decisions as complementary to, but not a replacement for, evaluation against the adversary.

13.5 How autonomous AI browsing changes the detection problem

AI browser agents affect the problem's nature in four different ways. They are real browsers and identity mismatch is one of the cues which makes the HTTP-library agents more useless [1], [12]. Actions of the agents depend on the page content and this makes navigation and timing rules meaningless. They do not have to be malicious agents and therefore the detection problem turns into a classification problem where multiclass labels and explainability are beneficial [14]. They are active agents

and humanization and stealth are two of the domains of study [13], [47]. Current evidence indicates that agents leave detectable traces under controlled conditions [12], [14], [46]; whether these persist as agents and automation frameworks evolve is the central open question.

13.6 Illustrative example of a research direction (not a finding)

The extra literature that the authors provide for their accompanying seminar includes the following definition for a "tripwire": "a hidden object that is placed in the path of a pointer toward a control in such a way that a click without previous motion of the pointer can be detected; the method can be generalized to a pipelined approach." We just include it to show how these signals might be implemented. It is a conceptual framework and proof-of-concept idea; its behaviour against live human and agent traffic has not been validated in this review, and it is not presented as evidence for any claim above.

13.7 Conceptual taxonomy

Figure 6 integrates the review dimensions into a six-level taxonomy: detection target, signal type, feature type, ML method, explainability method and evaluation method. It is a conceptual taxonomy synthesised from the reviewed literature and is not an established standard. Its use is to locate any detector, and any study, as a path through the levels, and to make gaps visible, for example paths with no explainability or no adversarial evaluation.

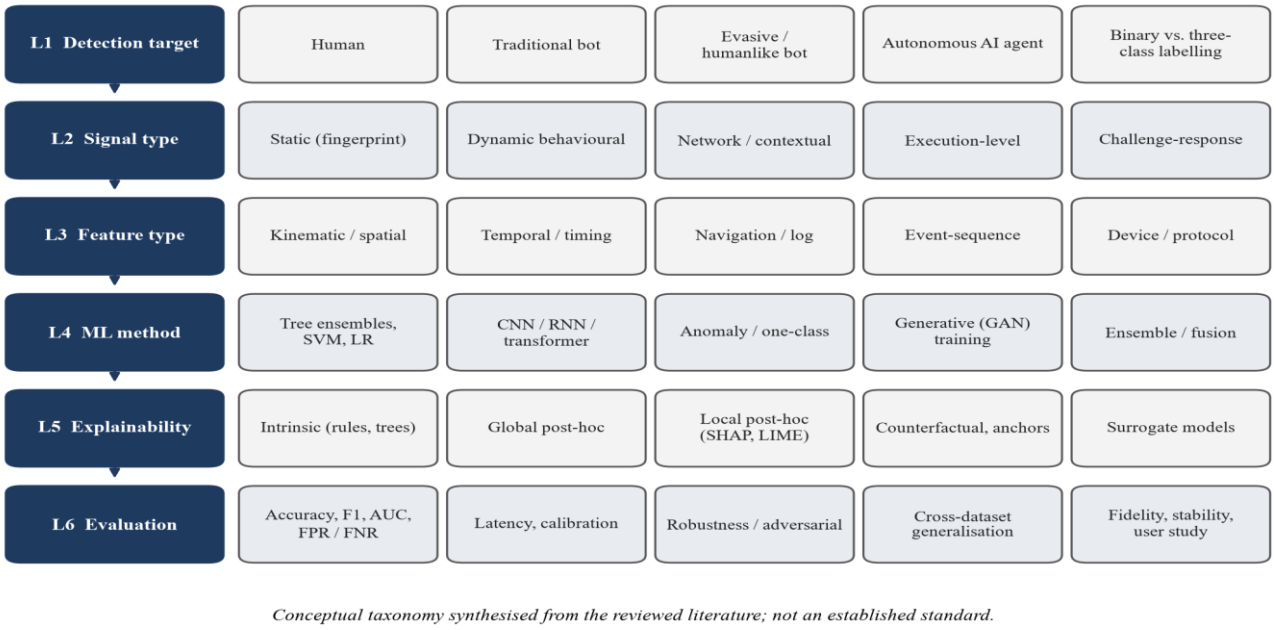


Figure 6. Conceptual taxonomy synthesised from the reviewed literature (six levels from detection target to evaluation method). It is not an established standard.

14. THREATS TO VALIDITY

Selection and coverage. The review is structured and comprehensive in aim but not systematic in method. Search was purposive, no counts of retrieved or screened records were recorded, screening was not performed independently by two reviewers, and database coverage was uneven. Relevant studies may therefore be missing. This also qualifies every statement about the absence of work in an area (Section 11).

Publication bias. Published studies are more likely to report successful detection than failure, and security results are often reported against the authors' own attackers [2], [3].

Terminology. The field lacks shared definitions. "Bot" ranges from simple scripts to advanced humanlike bots [2], and "AI agent" is defined differently across studies and is distinct from the earlier use of "agent" for crawlers. Studies using different terms may have been missed, and studies using the same term may not be comparable.

Rapid evolution and preprint status. The agent-detection evidence consists mostly of 2026 preprints that have not been peer reviewed and may change, and agents evolve faster than the publication cycle [12]–[14], [45], [46]. The results derived from their research are termed the results achieved by the researchers themselves.

Core-study access. The full text of the core study on sequence learning [15] was not available in the analysed material; statements about it are limited to publisher-level information, and it was not weighted in comparisons that require methodological detail.

Dataset availability and reproducibility. Availability of data and code was not verified for each resource, and several studies depend on bespoke testbeds [1], [2].

Heterogeneous evaluation. The research designs vary in terms of units of analysis, spatial designations, populations, and methodologies, hence cannot be quantitatively compared (see Section 9). Our qualitative judgements, such as the coverage assessment in Figure 5 and the table entries marked as conceptual, involve interpretation and could differ between reviewers.

Limited evidence. Evidence is limited regarding AI-agent behaviour in real traffic, explanation quality for analysts, accessibility effects, privacy-preserving detection and long-term drift. Conclusions in these areas should be regarded as provisional.

15. CONCLUSION

This review examined how autonomous AI browsing agents can be distinguished from humans and conventional bots using behavioural metrics and explainable machine learning. The literature establishes that behavioural biometrics, especially mouse dynamics, offer signals grounded in human neuromotor control [3], [22]; that web-log, fingerprint and network signals are effective against simple and naive automation but can be evaded or become less informative against clients that drive real browsers [1], [2], [12]; that combining signals raises the difficulty of evasion in the testbeds studied [2], [46]; and that explainable machine learning has so far been applied mainly to other biometric modalities and evaluated with model-centric measures [18]. Recent preprints indicate that AI browsing agents leave behavioural, execution-level and network-level traces and that a three-class formulation may be needed [12]–[14], [46].

The evidence base is narrow, however. Agent datasets are small and instrumented, evaluations differ and cannot be directly ranked, adaptive adversaries and drift are insufficiently studied, and explanations are rarely validated with analysts or examined for accessibility and privacy consequences. The research agenda proposed here, which comprises multi-signal fusion with interpretable components, three-class evaluation, human-centred XAI evaluation, robustness against humanised agents, privacy-preserving methods and standard benchmarks, is a set of proposals and not a set of established results. A reliable and responsible detector for autonomous AI browsing agents has not yet been demonstrated in the reviewed literature, and we do not claim that any single approach is universally effective.

REFERENCES

[1] X. Li, B. A. Azad, A. Rahmati, and N. Nikiforakis, "Good bot, bad bot: Characterizing automated browsing activity," in Proc. IEEE Symp. Security and Privacy (SP), 2021, pp. 1589–1605, doi: 10.1109/SP40001.2021.00079.

[2] C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and I. Kompatsiaris, "Detection of advanced web bots by combining web logs with mouse behavioural biometrics," Digit. Threats: Res. Pract., vol. 2, no. 3, pp. 1–26, 2021, doi: 10.1145/3447815.

[3] A. Acien, A. Morales, J. Fierrez, and R. Vera-Rodriguez, "BeCAPTCHA-Mouse: Synthetic mouse trajectories and improved bot detection," Pattern Recognit., vol. 127, Art. no. 108643, 2022, doi: 10.1016/j.patcog.2022.108643.

[4] S. Sivakorn, I. Polakis, and A. D. Keromytis, "I am robot: (Deep) learning to break semantic image CAPTCHAs," in Proc. IEEE European Symp. Security and Privacy (EuroS&P), 2016, pp. 388–403, doi: 10.1109/EuroSP.2016.37.

[5] I. Akrout, A. Feriani, and M. Akrout, "Hacking Google reCAPTCHA v3 using reinforcement learning," arXiv:1903.01003, 2019.

[6] A. Searles, Y. Nakatsuka, E. Ozturk, A. Paverd, G. Tsudik, and A. Enkoji, "An empirical study & evaluation of modern CAPTCHAs," in Proc. 32nd USENIX Security Symp., Anaheim, CA, USA, 2023.

[7] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," arXiv:2210.03629, 2022.

[8] S. Zhou et al., "WebArena: A realistic web environment for building autonomous agents," arXiv:2307.13854, 2023.

[9] X. Deng et al., "Mind2Web: Towards a generalist agent for the web," arXiv:2306.06070, 2023.

[10] B. Zheng, B. Gou, J. Kil, H. Sun, and Y. Su, "GPT-4V(ision) is a generalist web agent, if grounded," arXiv:2401.01614, 2024.

[11] H. He et al., "WebVoyager: Building an end-to-end web agent with large multimodal models," arXiv:2401.13919, 2024.

[12] E. Wang, Z. Shafiq, and Y. Vekaria, "FP-Agent: Fingerprinting AI browsing agents," arXiv:2605.01247, May 2026.

[13] I. Fayolle, S. Bouhenniche, S. Pélissier, P. Laperdrix, C. Maurice, and W. Rudametkin, "On the Internet,

nobody knows you're an LLM bot: Unmasking web agents with multi-layer fingerprinting," arXiv:2606.30119, Jun. 2026.

[14] V. Choudhary, L. Schmidt, A. Z. Kenntner, F. Skhab, M. Osswald, and J. Ernstberger, "What does it take to detect an AI agent? Minimal feature sets for behavioral detection under browser automation," arXiv:2607.26935, Jul. 2026.

[15] H. Niu, J. Chen, Z. Zhang, and Z. Cai, "Mouse dynamics based bot detection using sequence learning," in Biometric Recognition (CCBR 2021), Lecture Notes in Computer Science. Cham, Switzerland: Springer, 2021, pp. 49–56, doi: 10.1007/978-3-030-86608-2_6.

[16] European Parliament and Council, "Regulation (EU) 2016/679 (General Data Protection Regulation)," Official Journal of the European Union, L 119, pp. 1–88, May 4, 2016.

[17] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," Harv. J. Law Technol., vol. 31, no. 2, pp. 841–887, 2018.

[18] C. Tucci, A. Della Greca, G. Tortora, and R. Francese, "Explainable biometrics: A systematic literature review," J. Ambient Intell. Humaniz. Comput., 2024, doi: 10.1007/s12652-024-04856-1.

[19] H. Jonker, B. Krumnow, and G. Vlot, "Fingerprint surface-based detection of web bot detectors," in Computer Security – ESORICS 2019, Lecture Notes in Computer Science. Cham, Switzerland: Springer, 2019, pp. 586–605, doi: 10.1007/978-3-030-29962-0_28.

[20] A. Vastel, W. Rudametkin, R. Rouvoy, and X. Blanc, "FP-Crawlers: Studying the resilience of browser fingerprinting to block crawlers," in Proc. NDSS Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb), 2020, doi: 10.14722/madweb.2020.23010.

[21] B. A. Azad, O. Starov, P. Laperdrix, and N. Nikiforakis, "Web Runner 2049: Evaluating third-party anti-bot services," in Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), Lecture Notes in Computer Science. Cham, Switzerland: Springer, 2020, pp. 135–159, doi: 10.1007/978-3-030-52683-2_7.

[22] S. Khan, C. Devlen, M. Manno, and D. Hou, "Mouse dynamics behavioral biometrics: A survey," ACM Comput. Surv., vol. 56, no. 6, pp. 1–33, 2024, doi: 10.1145/3640311.

[23] M. Antal and E. Egyed-Zsigmond, "Intrusion detection using mouse dynamics," IET Biometrics, vol. 8, no. 5, pp. 285–294, 2019, doi: 10.1049/iet-bmt.2018.5126.

[24] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," J. Exp. Psychol., vol. 47, no. 6, pp. 381–391, 1954, doi: 10.1037/h0055392.

[25] W. E. Hick, "On the rate of gain of information," Q. J. Exp. Psychol., vol. 4, no. 1, pp. 11–26, 1952, doi: 10.1080/17470215208416600.

[26] R. Plamondon, "A kinematic theory of rapid human movements: Part I. Movement representation and generation," Biol. Cybern., vol. 72, no. 4, pp. 295–307, 1995, doi: 10.1007/s004220050131.

[27] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," IEEE Trans. Dependable Secure Comput., vol. 4, no. 3, pp. 165–179, 2007, doi: 10.1109/TDSC.2007.70207.

[28] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in Proc. ACM Workshop on Visualization and Data Mining for Computer Security, 2004, pp. 1–8, doi: 10.1145/1029208.1029210.

[29] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," Future Gener. Comput. Syst., vol. 16, no. 4, pp. 351–359, 2000, doi: 10.1016/S0167-739X(99)00059-X.

[30] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in Proc. IEEE/IFIP Int. Conf. Dependable Systems & Networks (DSN), 2009, pp. 125–134, doi: 10.1109/DSN.2009.5270346.

[31] A. Acien, A. Morales, J. V. Monaco, R. Vera-Rodriguez, and J. Fierrez, "TypeNet: Deep learning keystroke biometrics," IEEE Trans. Biometrics, Behav., Identity Sci., vol. 4, no. 1, pp. 57–70, 2022, doi: 10.1109/TBIOM.2021.3112540.

[32] Z. Chu, S. Gianvecchio, A. Koehl, H. Wang, and S. Jajodia, "Blog or block: Detecting blog bots through behavioral biometrics," Comput. Netw., vol. 57, no. 3, pp. 634–646, 2013, doi: 10.1016/j.comnet.2012.10.005.

[33] K. Park, V. S. Pai, K.-W. Lee, and S. Calo, "Securing web service by automatic robot detection," in Proc. USENIX Annual Technical Conf. (General Track), 2006, pp. 255–260.

[34] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine, "Browser fingerprinting: A survey," ACM Trans. Web, vol. 14, no. 2, pp. 1–33, 2020, doi: 10.1145/3386040.

[35] A. Vastel, P. Laperdrix, W. Rudametkin, and R. Rouvoy, "FP-STALKER: Tracking browser fingerprint evolutions," in Proc. IEEE Symp. Security and Privacy (SP), 2018, pp. 728–741, doi: 10.1109/SP.2018.00008.

[36] Z. Durumeric et al., "The security impact of HTTPS interception," in Proc. Network and Distributed System Security Symp. (NDSS), 2017, doi: 10.14722/ndss.2017.23456.

[37] B. Anderson and D. McGrew, "TLS beyond the browser: Combining end host and network data to understand application behavior," in Proc. Internet Measurement Conf. (IMC), 2019, pp. 379–392, doi: 10.1145/3355369.3355601.

[38] G. Jarad and K. Bicakci, "When handshakes tell the truth: Detecting web bad bots via TLS fingerprints," arXiv:2602.09606, Feb. 2026.

[39] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," Artif. Intell., vol. 267, pp. 1–38, 2019, doi: 10.1016/j.artint.2018.07.007.

[40] Z. C. Lipton, "The mythos of model interpretability," Commun. ACM, vol. 61, no. 10, pp. 36–43, 2018, doi: 10.1145/3233231.

[41] A. Barredo Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Inf. Fusion, vol. 58, pp. 82–115, 2020, doi: 10.1016/j.inffus.2019.12.012.

[42] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," Nat. Mach. Intell., vol. 1, no. 5, pp. 206–215, 2019, doi: 10.1038/s42256-019-0048-x.

[43] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," ACM Comput. Surv., vol. 51, no. 5, pp. 1–42, 2018, doi: 10.1145/3236009.

[44] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," IEEE Access, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[45] W. Lugoloobi, S. Marro, J. Magomere, J. Wright, and C. Russell, "Known by their actions: Fingerprinting LLM browser agents via UI traces," arXiv:2605.14786, May 2026.

[46] D. Kang, H. Jeong, J. Sheffey, P. Datta, and A. Houmansadr, "Whose agent are you? Multi-layer fingerprinting and attribution of autonomous web agents," arXiv:2606.20910, Jun. 2026.

[47] J. Zhu et al., "Turing test on screen: A benchmark for mobile GUI agent humanization," arXiv:2604.09574, 2026.

[48] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[49] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[50] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Ann. Stat., vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.

[51] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[52] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[53] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv:1406.1078, 2014.

[54] A. Vaswani et al., "Attention is all you need," arXiv:1706.03762, 2017.

[55] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, 2009, doi: 10.1145/1541880.1541882.

[56] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Comput., vol. 13, no. 7, pp. 1443–1471, 2001, doi: 10.1162/089976601750264965.

[57] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[58] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504–507, 2006, doi: 10.1126/science.1127647.

[59] I. J. Goodfellow et al., "Generative adversarial networks," arXiv:1406.2661, 2014.

[60] A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, and O. Delgado-Mohatar, "BeCAPTCHA: Behavioral bot detection using touchscreen and mobile sensors benchmarked on HuMIdb," Eng. Appl. Artif. Intell., vol. 98, Art. no. 104058, 2021, doi: 10.1016/j.engappai.2020.104058.

[61] S. T. K. Jan, Q. Hao, T. Hu, J. Pu, S. Oswal, and G. Wang, "Throwing darts in the dark? Detecting bots with limited data using neural data augmentation," in Proc.

IEEE Symp. Security and Privacy (SP), 2020, pp. 1190–1206, doi: 10.1109/SP40000.2020.00079.

[62] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," arXiv:1706.04599, 2017.

[63] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Comput. Surv., vol. 46, no. 4, pp. 1–37, 2014, doi: 10.1145/2523813.

[64] S. Jain and B. C. Wallace, "Attention is not explanation," in Proc. NAACL-HLT, 2019, pp. 3543–3556, doi: 10.18653/v1/N19-1357.

[65] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, "Explainable artificial intelligence in cybersecurity: A survey," IEEE Access, vol. 10, pp. 93575–93600, 2022, doi: 10.1109/ACCESS.2022.3204171.

[66] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 4765–4774.

[67] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[68] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," J. Mach. Learn. Res., vol. 20, no. 177, pp. 1–81, 2019.

[69] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," J. Comput. Graph. Stat., vol. 24, no. 1, pp. 44–65, 2015, doi: 10.1080/10618600.2014.907095.

[70] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in Proc. ACM Conf. Fairness, Accountability, and Transparency (FAT*), 2020, pp. 607–617, doi: 10.1145/3351095.3372850.

[71] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in Proc. AAAI Conf. Artificial Intelligence, vol. 32, no. 1, 2018, doi: 10.1609/aaai.v32i1.11491.

[72] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 11–20, doi: 10.18653/v1/D19-1002.

[73] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods," in Proc. AAAI/ACM Conf. AI, Ethics, and Society, 2020, pp. 180–186, doi: 10.1145/3375627.3375830.

[74] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in Proc. AAAI Conf. Artificial Intelligence, vol. 33, no. 1, 2019, pp. 3681–3688, doi: 10.1609/aaai.v33i01.33013681.

[75] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv:1702.08608, 2017.

[76] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Metrics for explainable AI: Challenges and prospects," arXiv:1812.04608, 2018.

[77] C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and Y. Kompatsiaris, "Towards a framework for detecting advanced web bots," in Proc. 14th Int. Conf. Availability, Reliability and Security (ARES), 2019, pp. 1–10, doi: 10.1145/3339252.3339267.

[78] C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and I. Kompatsiaris, "Web bot detection evasion using generative adversarial networks," in Proc. IEEE Int. Conf. Cyber Security and Resilience (CSR), 2021, pp. 115–120, doi: 10.1109/CSR51186.2021.9527915.

[79] C. Iliou, T. Kostoulas, T. Tsikrika, V. Katos, S. Vrochidis, and I. Kompatsiaris, "Web bot detection evasion using deep reinforcement learning," in Proc. 17th Int. Conf. Availability, Reliability and Security (ARES), 2022, pp. 1–10, doi: 10.1145/3538969.3538994.

[80] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," Pattern Recognit., vol. 84, pp. 317–331, 2018, doi: 10.1016/j.patcog.2018.07.023.

[81] W3C, "Inaccessibility of CAPTCHA: Alternatives to visual Turing tests on the web," W3C Working Group Note/Draft, 2021. [Online]. Available: https://www.w3.org/TR/turingtest/

[82] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2017.