WHEN BOTS USE AI
AND SO DO WE

THE FUTURE OF AUTOMATED DEFENSE

White Paper

Innovative ways for Bot Detection

EXECUTIVE SUMMARY

Bots are everywhere in today's digital world—some make life easier, but others are designed to exploit systems and commit fraud. Detecting these malicious bots is no longer optional; it's essential to protect trust, revenue, and user experience. The challenge is growing because fraudsters now use AI to build bots that look and behave like real users. These bots can bypass old defenses like CAPTCHAs and static rules, making traditional detection methods almost obsolete.

The good news is that AI can also be our strongest defense. By combining machine learning, data science, and behavioral analytics, we can create adaptive systems that spot anomalies in real time and respond intelligently. Instead of relying on rigid rules, modern bot detection uses dynamic models that learn and evolve as threats change. This approach turns AI from a curse into a lifesaver—helping businesses stay ahead of attackers and keep digital ecosystems safe without adding unnecessary friction for genuine users.

AUTHORS

Akhil Singhal

Senior Software Engineer at Microsoft, working with Commerce Core Trust platform team, with over a decade of experience in designing and implementing advanced AI-driven solutions for enterprise applications. As a key contributor to the CFAR, Akhil specializes in creating innovative, intelligent software systems that leverage Agentic AI, machine learning, graph analytics, and natural language processing to detect and prevent fraud across diverse sectors, including finance, supply chain, and compliance.

Sadhana Viswanathan

Principal Software Engineering Manager at Microsoft, overseeing the Trust platform team. She brings extensive expertise in guiding strategic technology initiatives and fostering innovation in fraud prevention and risk management. Under her leadership, team has advanced the development of scalable AI-driven platforms that integrate Agentic AI, machine learning, and advanced analytics to proactively detect and mitigate fraud across complex ecosystems.

Esther Amullen

Senior Data & Applied Scientist at Microsoft within the Trust Platform Data Science and Data Engineering organization, where she specializes in applying advanced AI and machine learning techniques to fraud detection and security risk mitigation. She brings deep expertise across fraud analytics, anomaly detection, predictive modeling, and responsible AI, with a strong focus on operationalizing models at scale through robust data pipelines and production-ready ML systems.

UNDERSTANDING BOTS

What is a Bot

A bot is an automated software application designed to perform tasks that mimic human interaction or execute repetitive actions without direct human input. Bots can range from simple scripts that send messages or scrape data to sophisticated, AI-powered agents capable of understanding context, generating responses, and adapting behavior in real time.

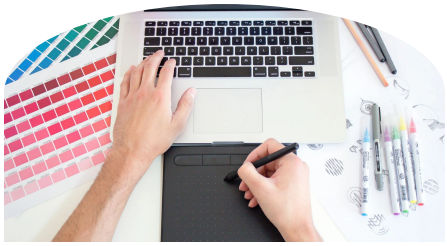
Characteristics of Bots

- **Automated Execution** – Bots run tasks without human intervention, following predefined scripts or logic.
- **Malicious or Goal-Oriented Intent** – Many bots are designed for harmful purposes like scraping, spamming, or fraud, though some serve legitimate automation needs.
- **Camouflage and Mimicry** – Bots often imitate human behavior (e.g., click timing, navigation patterns) to bypass detection.
- **Scalability** – Bots can operate at massive scale, sending thousands of requests or creating multiple accounts in seconds.
- **Consistency in Behavior** – Automated bots exhibit repetitive or aggregated patterns that differ from natural human variability.

NEED OF BOT DETECTION

1 Prevent Fraud and Abuse

Stop bots before they steal accounts, hoard inventory, or commit payment fraud.

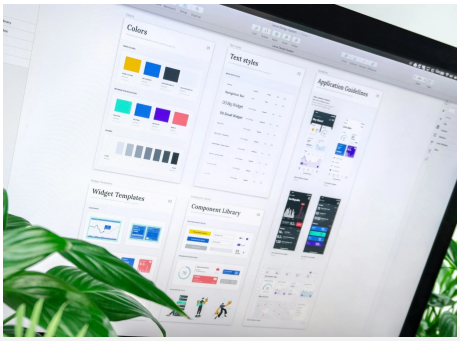


2 Protect System Performance

Keep your servers healthy by blocking traffic storms caused by automated scripts.

3 Safeguard Data and Privacy

Prevent bots from scraping sensitive information and violating user trust.



4 Maintain Trust and Compliance

Ensure your platform stays secure and meets regulatory standards.

5 Enable Accurate Analytics

Filter out fake traffic so your business decisions are based on real user behavior.

LEGACY WAYS
IN BOT DETECTION

User-Agent and Header Validation

Check for missing or suspicious HTTP headers and non-standard User-Agent strings.

IP Reputation and Rate Limiting

Block or throttle requests from known bad IPs or abnormal traffic spikes.

Captcha and Challenge Response

Use visual or interactive tests to separate humans from automated scripts.

Cookie and JavaScript Execution Checks

Require proper cookie handling and JavaScript execution to confirm real browsers.

DNS and Netwrok-Based Monitoring

Identify bots through anomalies in DNS queries, IP rotation, or reverse lookups.

Retrospective Remediation

Review logs post-event to flag and disable accounts linked to bot activity.

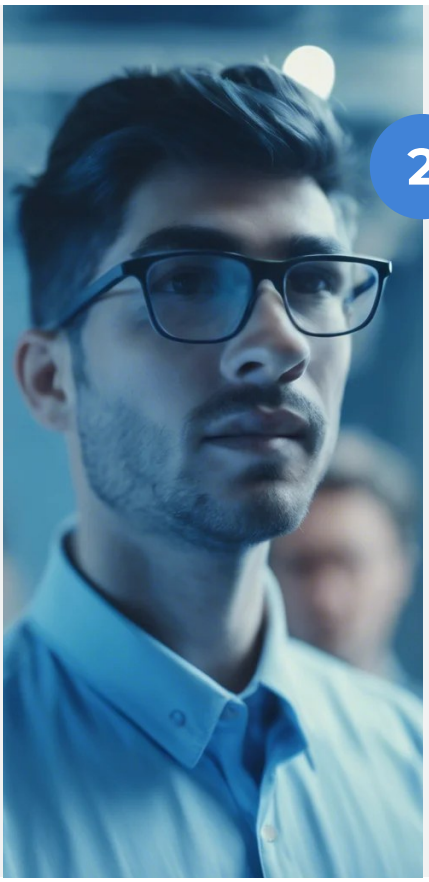
HOW AI DISRUPTING CLASSICAL BOT DETECTION TECHNIQUES

1

User-Agent and Header Validation

Legacy approach: Spot bots by checking for odd or missing headers.

AI impact: Modern bots use AI to perfectly mimic real browser headers, making this check almost useless.

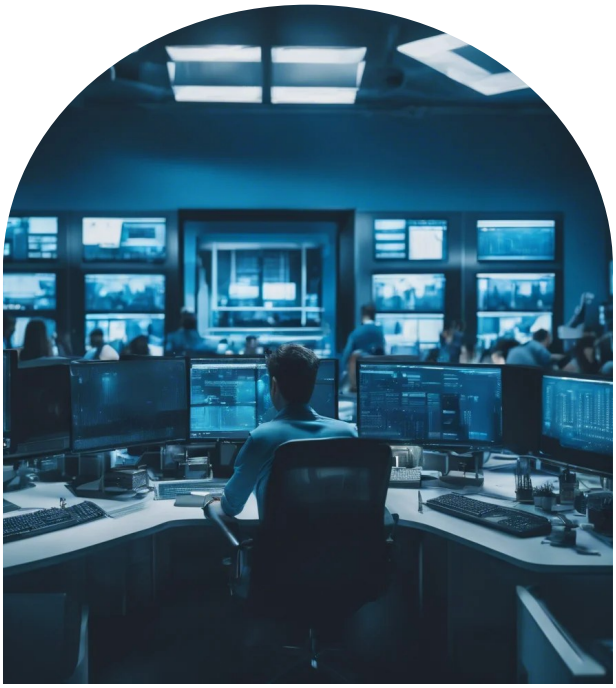


2

IP Reputation and Rate Limiting

Legacy approach: Block known bad IPs or throttle high-volume traffic.

AI impact: AI-driven bots rotate through residential proxies and learn throttling patterns, blending into normal traffic.



3 CAPTCHA and Challenge-Response

Legacy approach: Use puzzles to separate humans from bots.
AI impact: AI models now solve CAPTCHAs in seconds—or outsource them to human farms—rendering this defense weak.

6 Retrospective Remediation

Legacy approach: Analyze logs after the fact to catch bots.
AI impact: Bots adapt in real time, so by the time you review logs, the damage is done—and they've already evolved.

4 Cookie and JavaScript Execution Checks

Legacy approach: Require JS execution and cookie handling to confirm real browsers.
AI impact: AI bots run full browser automation stacks (like Puppeteer or Playwright) and pass these checks effortlessly.

5 DNS and Network-Based Monitoring

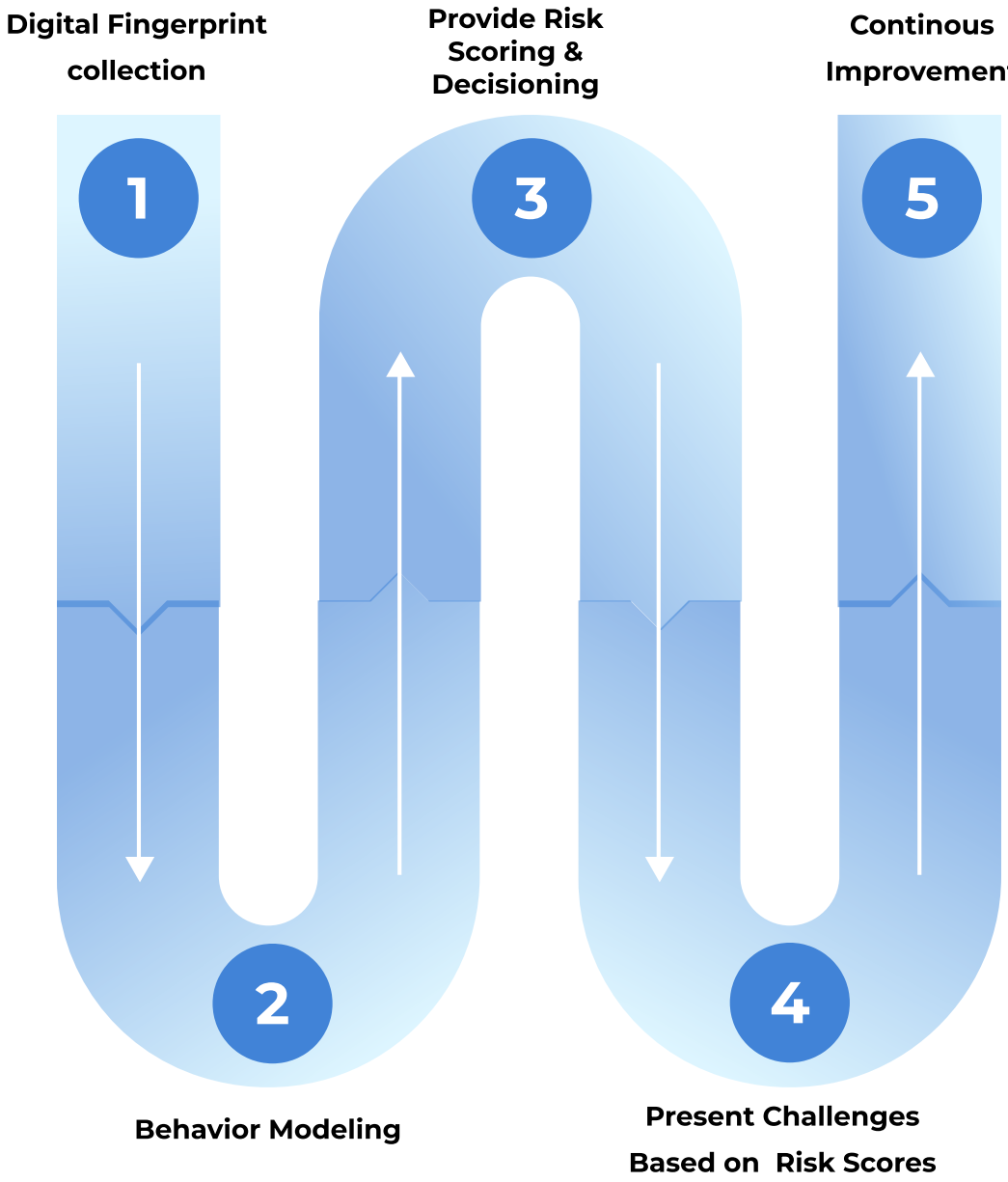
Legacy approach: Flag bots using IP rotation or suspicious DNS patterns.
AI impact: AI optimizes IP rotation strategies and uses residential IP pools, making detection far harder.

AI AND MACHINE LEARNING: THE NEW FRONTIER IN BOT DETECTION

Instead of relying on rigid rules, modern detection uses models that learn from thousands of signals—like device fingerprints, click patterns, and network activity—to spot what humans can't. These systems don't just react; they improve every day, adapting to new tricks as attackers invent them. By combining smart algorithms with real-time data, we can build defenses that stay one step ahead without slowing down genuine users.

This shift changes everything. Bot detection is no longer a static checklist—it's a living, learning process. AI helps us move from reactive fixes to proactive protection, keeping fraud out while preserving the smooth experience people expect. In a world where bots use AI, only AI-powered defenses can truly keep us safe.

STEPS IN AI/ML BASED DETECTION

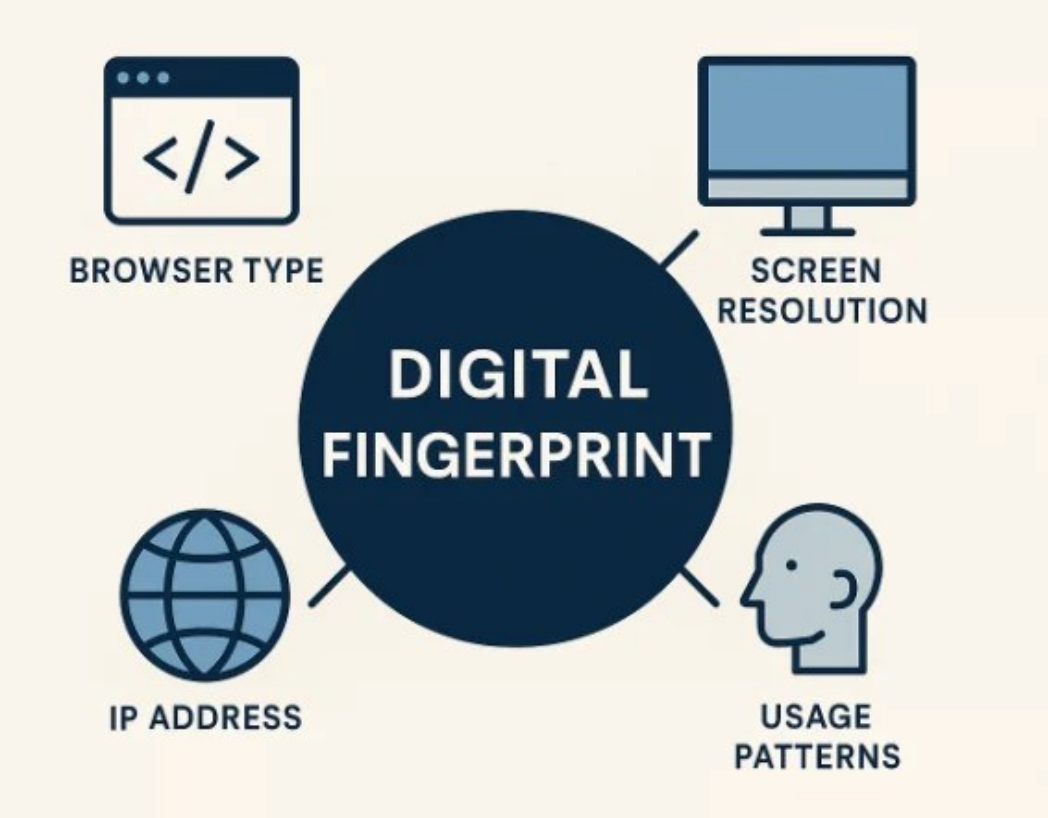


DIGITAL FINGERPRINT

What is a Digital Fingerprint

A digital fingerprint is like an online identity card for every device or session. It combines dozens of signals—browser type, screen resolution, IP address, usage patterns, and more—to create a unique signature that's hard to fake. Security systems use these fingerprints to recognize legitimate users and flag anomalies that might indicate bots or fraud.

Unlike passwords or CAPTCHAs, digital fingerprinting works quietly in the background, analyzing subtle details in real time. This makes it a powerful tool for modern bot detection, helping businesses stay ahead of attackers without adding friction for genuine users.



GATHERING INFORMATION

Digital Fingerprinting & User Behavioral pattern collection

The process begins by collecting digital fingerprint data along with user behavior on the webpage. One of the simplest ways to achieve this is by embedding a lightweight JavaScript snippet that runs when the page loads. This script observes interactions such as cursor movement, click patterns, and navigation flow, building a behavioral profile in real time.

Once the data is captured, it's sent to a backend API where machine learning models calculate a risk score—a measure of how likely the session is to be genuine or bot-driven.

Important Considerations

Privacy and Ethics: Browser fingerprinting raises significant privacy concerns. It should only be implemented with a clear understanding of legal and ethical implications, such as GDPR and CCPA, and with user consent where required.
Uniqueness and Persistence: While combining multiple attributes increases the likelihood of a unique fingerprint, it's not guaranteed to be perfectly unique or persistent across browser updates, setting changes, or anti-fingerprinting measures.
Hashing: For practical use, the collected data is typically hashed (e.g., using SHA-256) to produce a concise and manageable fingerprint string.

MACHINE LEARNING & AI FOR ADVANCED BOT DETECTION

Machine learning and artificial intelligence provide the analytical foundation needed to transform raw behavioral and environmental attributes into actionable bot detection intelligence. Rather than relying on individual signals in isolation, AI models learn how combinations of attributes—collected from client-side interaction telemetry, device and environment characteristics, network behavior, and contextual indicators—collectively distinguish automated activity from legitimate human behavior.

In this context, AI does not replace the attributes discussed in the previous section; instead, it amplifies their value. Machine learning models identify subtle patterns, temporal relationships, and inconsistencies that are difficult or impossible to encode using static rules. This capability enables detection systems to adapt continuously as bot behavior evolves, while supporting risk-based decisioning that balances security effectiveness with user experience.

CORE MODEL FAMILIES FOR BOT DETECTION

Tree-Based Models: Structured Learning from Multi-Signal Telemetry

Gradient-boosted tree models, such as XGBoost or LightGBM, extend the capabilities of basic tree ensembles by learning decision boundaries sequentially. Each successive tree focuses on correcting the errors of prior trees, allowing the model to capture complex non-linear relationships across attributes.

In bot detection systems, gradient-boosted models are often used as the primary risk scoring engine. They excel at combining behavioral variability, fingerprint stability metrics, network reputation signals, and historical challenge outcomes into a single probabilistic score. Rather than producing a binary outcome, these models generate a continuous risk estimate that supports graduated responses—such as allowing, challenging, or blocking traffic.

Gradient-Boosted Trees: High-Precision Risk Scoring at Scale

Tree-based models, including Decision Trees and Random Forests, represent one of the most widely adopted approaches for bot detection in production environments. These models learn a hierarchy of decision rules that partition traffic based on combinations of attributes rather than single indicators.

Applied to bot detection, tree-based models typically operate on structured representations of behavioral metrics, device fingerprints, and network features. They can learn interactions such as inconsistencies between browser characteristics and observed behavior, abnormal timing patterns in navigation flows, or repeated reuse of the same device or network characteristics across multiple sessions. By combining these attributes, tree-based models detect automation even when individual signals appear benign.

Classical Linear and Distance-Based Models: Baselines and Control Points

Classical models such as logistic regression, support vector machines, and distance-based classifiers play a supporting role in modern bot detection architectures. While they are rarely sufficient on their own for detecting sophisticated bots,
they remain valuable as baselines, control models, or components within ensemble systems.
However, as bot behavior becomes increasingly adaptive and multi-dimensional, linear decision boundaries often fail to capture the complexity required for high-confidence detection. As a result, these models are best positioned as benchmarks, explainability anchors, or fallback mechanisms rather than primary detectors.

DEEP LEARNING AS A GAME CHANGER FOR BEHAVIORAL ANALYSIS

Sequential Deep Learning for Interaction Dynamics

Deep learning introduces a fundamentally different capability: the ability to model sequences of behavior rather than static summaries. Recurrent neural networks, including LSTM and GRU architectures, learn directly from ordered event streams such as mouse movements, scrolling behavior, keystroke timing, and navigation sequences.

This approach is particularly powerful against bots that successfully mimic high-level behavioral statistics but fail to reproduce the nuanced dynamics of human interaction. Deep sequence models can detect unnatural consistency, unrealistic reaction times, or implausible transitions between actions—signals that are difficult to encode manually.

While deep learning models significantly enhance detection of advanced automation, they come with higher computational and latency costs. For this reason, they are often deployed selectively—either as secondary evaluators for high-risk sessions or as offline models that inform rule updates and feature refinement.

Language Understanding with NLP, Transformers, and Embeddings

As bots increasingly interact through natural language—whether in chat interfaces, forms, reviews, or social channels—text becomes a critical detection surface. Transformer-based models, such as those built on pre-trained language embeddings, enable detection systems to evaluate linguistic patterns at scale.

These models convert text into contextual embeddings that capture semantics, structure, and stylistic nuance. When applied to bot detection, embeddings can reveal templated responses, unnatural phrasing, low linguistic diversity, or inconsistencies between language use and claimed identity. Importantly, these insights are strongest when combined with behavioral and environmental attributes rather than used in isolation.

Transformer-based NLP models significantly reduce false negatives associated with “human-like” bots, but they also introduce additional privacy, latency, and cost considerations. As with deep behavioral models, they are most effective when incorporated into a layered detection strategy.

DEEP LEARNING AS A GAME CHANGER FOR BEHAVIORAL ANALYSIS CONT..

Generative Pre-Trained Models and Emerging Capabilities

Generative pre-trained models introduce both new risks and new defensive opportunities. On one hand, they lower the barrier for adversaries to generate convincing, context-aware content at scale. On the other, they provide defenders with powerful tools for anomaly detection, semantic consistency analysis, and adaptive learning.

In bot detection, generative models can be used to simulate adversarial behavior, augment training data, and identify semantic mismatches across interactions. They also support advanced reasoning over heterogeneous signals—connecting behavioral patterns, language use, and contextual cues into a unified assessment. While still emerging, these capabilities point toward more autonomous, self-improving detection systems.

FROM ATTRIBUTES TO FEATURES: FEEDING MODELS EFFECTIVELY

Regardless of model type, success depends on transforming raw attributes into features that capture intent, consistency, and deviation from expected norms. Behavioral telemetry may be converted into timing distributions, entropy measures, or sequence embeddings. Device and environment attributes become stability metrics, uniqueness indicators, and cross-session correlation features. Network signals evolve into churn rates, reputation aggregates, and anomaly scores.

Different model families consume these features differently. Tree-based and boosted models rely on well-structured, tabular representations. Deep learning models operate on sequences or embeddings derived directly from raw events or text. Hybrid architectures combine both approaches, allowing each model to contribute where it is strongest.

EVALUATING DETECTION PERFORMANCE BEYOND ACCURACY

Evaluating bot detection models requires more than traditional accuracy metrics. Because false positives degrade user experience and false negatives enable abuse, performance must be assessed through a cost-sensitive lens.

Core evaluation metrics include precision, recall, false positive rate, false negative rate, and F1 score, all derived from the confusion matrix. Risk-based systems additionally rely on ranking metrics such as ROC-AUC and precision–recall AUC to assess how well models separate benign from malicious activity across thresholds.

Equally important is evaluation design. Time-based validation ensures resilience to behavioral drift, while segmented analysis by traffic source, geography, device type, and challenge outcome reveals hidden failure modes. Post-deployment metrics—such as challenge success rates, abandonment rates, latency, and infrastructure cost—complete the picture.

END-TO-END AI PIPELINE FOR BOT DETECTION

An effective AI-driven bot detection system follows a continuous lifecycle. Behavioral and environmental data are ingested in real time, transformed into features, and scored by one or more models. The resulting risk score informs decisioning logic that determines whether traffic is allowed, challenged, or blocked.

Outcomes from these decisions—challenge results, confirmed fraud, and user feedback—are fed back into the training pipeline. This feedback loop enables continuous improvement, allowing the system to adapt as attackers evolve their tactics.

PERFORMANCE, LATENCY, AND FEATURE COST TRADEOFFS

Finally, model selection and feature richness must align with operational constraints. Boosted tree models offer an excellent balance of accuracy and latency for real-time decisioning. Deep learning and NLP models deliver higher detection power but require careful deployment to manage compute cost and response times.

Similarly, richer telemetry improves detection but increases ingestion, processing, and privacy considerations. Successful implementations deliberately balance these tradeoffs, using layered scoring strategies that apply the right level of intelligence at the right time.

CORE MODEL FAMILIES FOR BOT DETECTION

Tree-Based Models: Structured Learning from Multi-Signal Telemetry

Gradient-boosted tree models, such as XGBoost or LightGBM, extend the capabilities of basic tree ensembles by learning decision boundaries sequentially. Each successive tree focuses on correcting the errors of prior trees, allowing the model to capture complex non-linear relationships across attributes.

In bot detection systems, gradient-boosted models are often used as the primary risk scoring engine. They excel at combining behavioral variability, fingerprint stability metrics, network reputation signals, and historical challenge outcomes into a single probabilistic score. Rather than producing a binary outcome, these models generate a continuous risk estimate that supports graduated responses—such as allowing, challenging, or blocking traffic.

Gradient-Boosted Trees: High-Precision Risk Scoring at Scale

Tree-based models, including Decision Trees and Random Forests, represent one of the most widely adopted approaches for bot detection in production environments. These models learn a hierarchy of decision rules that partition traffic based on combinations of attributes rather than single indicators.

Applied to bot detection, tree-based models typically operate on structured representations of behavioral metrics, device fingerprints, and network features. They can learn interactions such as inconsistencies between browser characteristics and observed behavior, abnormal timing patterns in navigation flows, or repeated reuse of the same device or network characteristics across multiple sessions. By combining these attributes, tree-based models detect automation even when individual signals appear benign.

Classical Linear and Distance-Based Models: Baselines and Control Points

Classical models such as logistic regression, support vector machines, and distance-based classifiers play a supporting role in modern bot detection architectures. While they are rarely sufficient on their own for detecting sophisticated bots,
they remain valuable as baselines, control models, or components within ensemble systems.
However, as bot behavior becomes increasingly adaptive and multi-dimensional, linear decision boundaries often fail to capture the complexity required for high-confidence detection. As a result, these models are best positioned as benchmarks, explainability anchors, or fallback mechanisms rather than primary detectors.

DECIDE CHALLENGE BASED ON RISK SCORE

Challenge selection

Once the model gives us a risk score, we decide how to proceed. If the confidence is high that the person at the keyboard is genuine, we let them continue without any challenge—because adding unnecessary friction hurts the user experience. But if the risk score suggests uncertainty, we introduce adaptive challenges based on the level of risk.

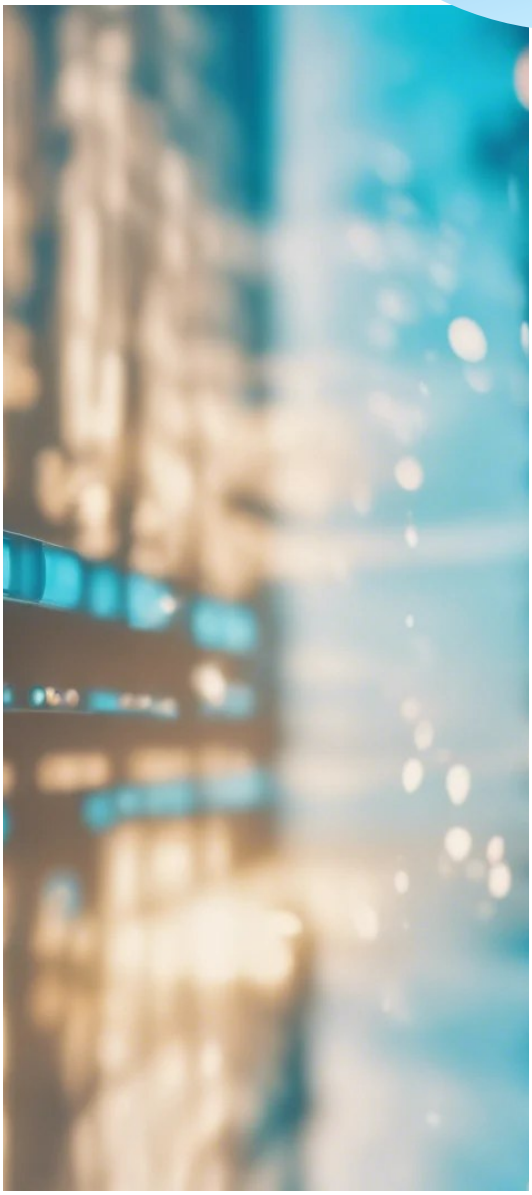
For low-risk cases (say around 10%), a simple CAPTCHA is enough. For medium risk, we can step it up with image-based pattern checks or quick math questions like "**What's 3 + 5**?". When the risk is high, we go further—asking the user to turn on their camera and perform a quick liveness check, such as moving their head or blinking. The principle is simple: **make it easy for genuine users and harder for suspicious ones** by scaling friction according to risk.

Feedback loop & Continuous Improvement

Bot detection shouldn't be static—it needs to learn and adapt. Every challenge outcome is a valuable signal. If the model predicts low risk but the user fails a challenge and turns out to be a bot, that insight should feed back into the system. The model can retrain on those patterns so next time, similar behavior triggers a higher risk score. Likewise, if a bot slips through and is later identified by the fraud team, that event should be logged and used to refine detection logic. This self-learning approach creates a continuous feedback loop, making the system smarter with every interaction and reducing the chances of repeat mistakes.

CONCLUSION

The battle against bots is no longer about static rules and simple CAPTCHAs—it's an arms race powered by Artificial Intelligence. As fraudsters use AI to create adaptive, human-like bots, traditional defenses fall short. The future of bot detection lies in embracing AI and Machine Learning not just as tools, but as intelligent systems that learn, adapt, and evolve. By combining digital fingerprinting, behavioral analytics, and risk-based adaptive challenges within a continuous feedback loop, organizations can build defenses that stay ahead of attackers without sacrificing user experience.



In this new era, success depends on speed, intelligence, and resilience. AI-driven detection transforms security from reactive to proactive, ensuring that every interaction strengthens the system. When bots use AI, so must we—because only intelligent defenses can protect trust at scale.