CENTRAL ASIAN JOURNAL OF INNOVATIONS ON TOURISM MANAGEMENT AND FINANCE

https://cajitmf.casjournal.org/index.php/CAJITMF

Volume: 07 Issue: 03 | July 2026 ISSN: 2660-454X

Article

An Agent-Oriented Adaptive Machine Learning Framework for Bias-Aware DDoS Attack Identification

Bekov Sanjar Nigmandjanovich*[1]

1. Independent Researcher at Tashkent International University
*Correspondance: sanjar.bekov@gmail.com

Abstract: Distributed Denial-of-Service (DDoS) attacks constitute a persistent and significant threat to the availability of Internet services, cloud platforms, and Internet of Things infrastructures. Numerous machine-learning approaches conceptualize DDoS detection as a single classification task, in which a single model labels network traffic as either benign or malicious. While such systems may demonstrate strong performance on benchmark datasets, they remain susceptible to dataset bias, class imbalance, concept drift, suboptimally calibrated thresholds, and limited interpretability. This study introduces an agent-oriented, adaptive machine-learning framework that allocates traffic monitoring, feature extraction, supervised classification, anomaly detection, normal-behavior modeling, transparent rule analysis, evidence fusion, explanation, analyst feedback, and bias monitoring to specialized agents. The architecture amalgamates independent sources of evidence, thereby mitigating overreliance on any single model, dataset, or feature family. The proposed methodology employs CIC-DDoS2019 for primary offline training and incorporates cross-dataset testing, nested cross-validation, per-attack metrics, probability calibration, agent-ablation studies, and concept-drift monitoring. An event-driven Spring microservice implementation facilitates modular model replacement. Rather than asserting unverified experimental results, this paper provides a reproducible architecture and evaluation protocol, thereby laying the foundation for developing adaptive, interpretable, and bias-aware multi-agent DDoS detection systems.

Keywords: DDoS Detection, Multi-Agent Systems, Agent-Oriented Programming, Machine Learning, Evidence Fusion, Model Selection, Dataset Bias, Anomaly Detection, Explainable AI

Citation: Nigmandjanovich, B. S. An Agent-Oriented Adaptive Machine Learning Framework for Bias-Aware DDoS Attack Identification. Central Asian Journal of Innovations on Tourism Management and Finance 2026, 7(3), 394-405.

Received: 05th Apr 2026
Revised: 20th Apr 2026
Accepted: 15th May 2026
Published: 31st May 2026

1. Introduction

The ongoing expansion of digital services has established availability as a fundamental pillar of economic and societal activity. Domains such as banking, e-commerce, cloud platforms, public administration, education, and industrial control systems are increasingly reliant on the continuous availability of services, even amid rapidly evolving network conditions. Distributed Denial-of-Service (DDoS) attacks specifically compromise this availability by depleting bandwidth, connection states, processing capacity, or application resources. The consequences of such attacks include service interruptions, transaction losses, increased incident-response expenditures, reputational harm, and diminished user trust. Importantly, the challenge is not limited to high-volume flooding; protocol-based attacks, application-layer request floods, reflection and amplification techniques, and low-rate strategies produce varied traffic signatures and operational impacts [1].

Machine learning is a compelling approach to DDoS detection, owing to its ability to discern patterns in network data and use them to predict future labels or risk states.

From a probabilistic perspective, supervised detection entails estimating the distribution of the output label conditional on a given traffic feature vector, whereas anomaly detection assesses the degree to which an observation deviates from an anticipated distribution. Murphy underscores that machine learning extends beyond a mere aggregation of algorithms; it embodies a principled methodology involving model selection, parameter estimation from data, and evaluation of generalization capability. This distinction is particularly salient in the domain of cybersecurity, wherein high accuracy on a familiar dataset may coincide with suboptimal performance on novel networks, attack implementations, or temporal contexts.

Most published DDoS detection systems are structured around a singular, predominant model. Typically, a supervised classifier is trained to distinguish benign traffic from one or more attack classes, or an anomaly detector is calibrated to normal traffic and used to identify deviations. While single-model architecture offers implementation simplicity, it consolidates a range of risks. For instance, the model may memorize laboratory-specific identifiers, overly depend on packet volume, exhibit diminished performance on minority attack types, display overconfidence, or erroneously classify legitimate flash crowds as malicious activity. A model trained on CIC-DDoS2019, for example, may leverage dataset-specific artifacts that are absent from enterprise, cloud, or IoT environments. This issue manifests not only as statistical error but also as operational opacity: analysts may receive elevated risk scores without clarity regarding whether the decision was influenced by protocol signatures, anomalous source distributions, deviations from service baselines, or dataset-specific heuristics.[2]

Agent-oriented software engineering presents a compelling alternative to traditional monolithic detection architectures. An intelligent agent is conventionally defined by attributes such as autonomy, reactivity, proactiveness, and social ability [3]. Rather than consolidating all responsibilities within a single detector, an agent-oriented system delegates specialized objectives and local knowledge to collaborative components. Within the proposed framework, individual agents are designated to perform supervised classification, anomaly estimation, normal behavior modeling, and transparent analysis of protocol rules. The outputs generated by these agents are subsequently integrated by an evidence-fusion agent, while supplementary agents facilitate explanation, analyst feedback, and bias monitoring. This architectural approach thus partitions detection responsibilities while preserving a unified final verdict.

This architectural decomposition is informed by the no-free-lunch principle, which posits that no learning model achieves uniform optimality across the entire spectrum of possible problems, and that each model inherently incorporates assumptions regarding the data-generating process [4]. For instance, a random forest may effectively identify established attack patterns; an isolation forest may be responsive to novel deviations; a statistical baseline may discern service-specific surges from typical workloads; and a transparent rule may elucidate incomplete handshake patterns. The principal merit of architecture lies in its ability to synthesize these partial perspectives while maintaining each constituent score for auditing and subsequent analysis.[5]

The objective of this research is to establish a reproducible architecture and evaluation protocol for bias-aware, multi-agent DDoS detection. This study addresses five principal questions: (1) how detection responsibilities can be effectively distributed among cooperating agents; (2) which feature groups and models are optimally allocated to these

agents; (3) whether evidence fusion enhances robustness in comparison to a single classifier; (4) how dataset, feature, threshold, and temporal biases may be mitigated; and (5) how explanations and analyst feedback can be incorporated without compromising the integrity of independent testing. The contribution of this work is primarily methodological, rather than a declaration of empirical superiority. The study delineates the recommended implementation procedures, training methodologies, and requisite experimental protocols necessary to substantiate performance claims.[6]

Literature Review

Historically, research on Distributed Denial-of-Service (DDoS) attacks has classified them by the exploited resource, traffic mechanism, degree of distribution, and location of the defense. Foundational taxonomies of attack and defense mechanisms, along with comprehensive surveys of detection, source identification, filtering, and response strategies, have been well-established in the literature. These works collectively illustrate the inadequacy of employing a single traffic threshold for detection: for example, high-rate UDP floods, SYN floods, HTTP request floods, and reflective amplification attacks may each target distinct protocol layers and generate diverse feature profiles.[7]

The availability of public datasets has facilitated reproducible comparisons of machine learning algorithms; however, these datasets also influence the characteristics that models ultimately learn. The CIC-DDoS2019 dataset, for instance, provides packet captures and labeled flow records that encompass benign traffic and several DDoS families, including SYN, UDP, DNS, NTP, LDAP, SSDP, TFTP, and WebDDoS scenarios. CSE-CIC-IDS2018 extends this context to additional intrusion categories, while CAIDA traces provide packet-level DDoS observations, and Bot-IoT captures IoT-oriented attack traffic. Methodological evaluations demonstrate that network intrusion datasets exhibit substantial variation in terms of realism, labeling accuracy, feature availability, and attack coverage. Accordingly, evaluation based solely on a single random train-test split is insufficient to establish robustness for real-world deployment.[8]

Supervised learning persists as the principal methodology for labeled DDoS detection. Logistic regression serves as a transparent probabilistic baseline; decision trees offer interpretable data partitions; random forests attenuate the variance inherent in individual trees; and gradient-boosted models effectively capture complex nonlinear interactions. Support vector machines demonstrate strong performance in high-dimensional feature spaces, while neural networks can learn intricate representations with sufficient data and appropriate regularization. The primary limitation of these approaches lies in their reliance on representative labels. If an attack family or network condition is absent from the training data, a closed-set classifier may erroneously assign a familiar label with unwarranted confidence.[9]

Anomaly detection mitigates this limitation by modeling normal behavior or isolating outliers, rather than relying solely on attack labels. For example, the Isolation Forest algorithm identifies data points that can be separated by short random partitions; one-class support vector machines estimate the support of high-dimensional distributions; and autoencoders utilize reconstruction error as an indicator of abnormality. While these methods can respond to previously unseen behaviors, their outputs do not necessarily correspond to confirmed attacks. Routine workload fluctuations, the introduction of new applications, scheduled maintenance periods, and legitimate flash crowds may all be classified as anomalous. Accordingly, anomaly-based evidence must be interpreted within

its operational context and should not be employed as an autonomous basis for blocking decisions.[10]

The extant literature on machine learning for intrusion detection consistently cautions against extrapolating benchmark performance to operational effectiveness. Researchers have articulated the semantic gap between statistical anomalies and security-relevant events, emphasizing the challenges of acquiring representative labels and achieving stable base rates. Comprehensive surveys look at the diverse array of machine-learning and data-mining methodologies employed in cyber analytics; however, methodological diversity does not obviate the necessity for realistic evaluation. Device-specific behavioral patterns can facilitate IoT DDoS detection, and distributed deep learning approaches offer alternative solutions. Collectively, these studies underscore the rationale for hybrid and distributed detection architectures while reinforcing the imperative to rigorously assess generalization across diverse operational environments.

The conceptualization of agent-oriented intrusion detection systems predates contemporary microservice architectures. Agent-oriented programming has been formalized as a paradigm centered on computational entities endowed with beliefs, capabilities, and commitments. Autonomous agent architectures for intrusion detection distributed low-level data collection and analytical tasks among multiple agents, thereby promoting flexibility and fault isolation. Subsequently, the JADE framework established a Java-based infrastructure for constructing communicating agents that adhere to standardized interaction protocols. These foundational approaches remain relevant, as modern detection pipelines similarly require autonomous components, asynchronous communication, and clearly delineated responsibility boundaries.

Contemporary research on DDoS mitigation has increasingly integrated agent-oriented paradigms with machine learning techniques. Intelligent agent-based systems have introduced automated feature extraction and selection on datasets like CIC-DDoS2019. This line of research substantiates the feasibility of linking agent-based decomposition with data-driven detection; however, several critical questions persist: the mechanisms by which multiple heterogeneous detectors should exchange evidence, the formal representation of inter-detector disagreement, the strategies for monitoring model bias, and the validation of composite risk scores. The framework proposed herein addresses these challenges through explicit fusion of four sources of evidence and independent bias monitoring.[11]

Model selection and evaluation constitute fundamental components of the bias-aware research objective. Cross-validation and bootstrap methodologies have been systematically evaluated for classifier selection, and it has been demonstrated that employing cross-validation for both model tuning and final error estimation can introduce optimistic bias. Consequently, nested cross-validation or an entirely independent test set is essential to obtain unbiased estimates. Class imbalance presents an additional challenge: overall accuracy may remain artificially elevated, even when minority attack classes are systematically overlooked. In such contexts, metrics such as precision, recall, macro-F1, false positive rate, and precision-recall curves offer more informative assessments of model performance.

Operational deployment necessitates both calibrated confidence and adaptive mechanisms. For a numerical risk score to effectively inform threshold-based decisions, it must exhibit reasonable correspondence to empirical event frequencies. Probability

calibration techniques can enhance the interpretability and practical utility of classifier outputs, particularly for complex models. Furthermore, it has been demonstrated that concept drift can alter the mapping between input features and output labels over time. While interpretability tools such as LIME and SHAP can aid analysts in evaluating individual model decisions, they do not constitute definitive proof of model correctness. Collectively, these insights underscore the need for a framework that integrates heterogeneous detectors, maintains version control, monitors concept drift, and regards explanations as decision-support tools rather than as sources of ground truth.[12]

2. Research Methodology

Research Design

This study employs a design-science paradigm in conjunction with a controlled experimental methodology. The primary artifact is an agent-oriented DDoS detection framework, whose evaluation involves a comparative analysis of increasingly sophisticated systems applied to consistent data partitions and traffic windows. System A comprises a single supervised classifier; System B incorporates independent anomaly detection, baseline modeling, and rule-based agents with evidence fusion; and System C extends the architecture to include feedback learning, explanation, and bias monitoring. This incremental design facilitates isolating architectural advantages from performance improvements attributable solely to enhanced base classifiers.

Data Sources and Experimental Boundaries

The CIC-DDoS2019 dataset is designated as the primary development dataset owing to its inclusion of multiple DDoS families and labeled flow records amenable to multiclass learning. CSE-CIC-IDS2018, Bot-IoT, UNSW-NB15, and selected CAIDA traces are reserved for robustness assessments or cross-dataset experiments, contingent upon feature compatibility. The study relies exclusively on public datasets and authorized passive telemetry. Scenario metadata is preserved for evaluation purposes but is systematically excluded from feature vectors to prevent models from inferring outcomes based on filenames, correlation identifiers, source addresses, or laboratory-specific topology.[13]

Data Preparation

The preprocessing pipeline eliminates duplicate records, infeasible or infinite values, missing labels, and features that explicitly encode the data-collection scenario. Numerical attributes are standardized or robustly scaled as necessitated by the chosen algorithm. Categorical protocol fields are encoded in a consistent manner, and all feature transformations are fitted exclusively on training data. Sampling, class weighting, or synthetic minority oversampling techniques are implemented within training folds rather than prior to data partitioning, thereby preventing information leakage from the validation or test sets into the model training process.

Traffic Representation

Each observation is mapped to a fixed analysis window or a completed network flow. The Feature Extraction Agent generates multiple complementary feature families. Volume features encompass packets per second, bytes per second, flow rates, and packet-size statistics. Protocol features include counts of SYN, ACK, RST, UDP, and ICMP packets, as well as ratios such as SYN-to-ACK. Distribution features include unique source counts, source entropy, destination-port entropy, and measures of destination concentration.

Temporal features capture metrics such as duration, inter-arrival statistics, and burstiness. Application-level features may include HTTP request rates and response code distributions, contingent upon the availability of lawful metadata. Baseline features quantify deviations from the normative profiles of hosts, services, destinations, or specific time periods.[14]

Agent Responsibilities

The Traffic Monitoring Agent ingests flow records or dataset events and aggregates them into correlated analysis windows. The Feature Extraction Agent generates a versioned feature vector for each window. The Supervised Classification Agent estimates class probabilities for benign traffic and recognized attack categories. The Anomaly Detection Agent outputs a normalized novelty score, while the Normal Behavior Agent evaluates each window against a locally learned baseline. The Rule Evidence Agent applies transparent protocol and rate-based rules. Subsequently, the Evidence Fusion Agent integrates the outputs of the preceding four agents. The Explainability Agent documents influential features and activated rules, the Feedback Learning Agent archives analyst-provided labels, and the Bias Monitoring Agent calculates per-class and temporal diagnostic metrics.

Candidate Models

The Supervised Agent initially evaluates logistic regression, decision trees, random forests, gradient boosting methods, support vector machines, and a compact multilayer perceptron. Logistic regression serves as an interpretable baseline, while random forests and gradient boosting offer robust nonlinear alternatives; the neural model is incorporated only when the volume of training data and calibration metrics warrant its inclusion. The Anomaly Agent assesses the performance of Isolation Forest, one-class SVM, and autoencoder-based approaches. The Baseline Agent employs rolling medians, exponentially weighted statistics, and robust z-scores, given their transparency and computational efficiency. Model complexity is escalated exclusively in response to clear improvements observed during validation.

3. Results and Discussion

Evidence Normalization

Each agent is required to produce a scalar value in the unit interval, along with metadata specifying the model version and the applied transformation. Classification probabilities are calibrated using validation datasets. Anomaly and baseline scores are mapped to the unit interval through training-distribution quantiles or monotonic calibration techniques. Rule-based evidence is derived from bounded contributions, as opposed to an unbounded tally of activated rules. While this normalization process does not render the signals statistically identical, it establishes a well-documented common scale for the initial evidence fusion experiment.[15]

Evidence Fusion. For a given analysis window, denoted as w, let c(w) represent the supervised DDoS probability, a(w) the anomaly score, b(w) the baseline-deviation score, and r(w) the rule-evidence score. The fused risk is subsequently defined as a convex combination:

$$f(w) = \omega c\ \mathrm{c(w)} + \omega a\ \mathrm{a(w)} + \omega b\ \mathrm{b(w)} + \omega r\ \mathrm{r(w)}$$

$$\omega c + \omega a + \omega b + \omega r = 1,\ \omega i \geq 0$$

where all weights are constrained to be non-negative and collectively sum to one. An explicit initial configuration assigns weights of 0.40 to classification, 0.25 to anomaly detection, 0.20 to baseline deviation, and 0.15 to rule-based evidence. These values serve as design defaults rather than asserted optima. The weights are selected and subjected to stress testing on validation data, with the independent test set remaining unaltered. The weighted fusion approach is evaluated alongside a logistic stacking model to assess whether learned fusion strategies yield sufficient performance gains to justify diminished transparency.

Risk Mapping and Decision Policy

The fused risk score is categorized as NORMAL for values below 0.40, SUSPICIOUS for values from 0.40 to below 0.70, PROBABLE_DDOS for values from 0.70 to below 0.85, and HIGH_CONFIDENCE_DDOS for values of 0.85 and above. Thresholds are calibrated with respect to explicit operational cost considerations, rather than being selected solely to maximize accuracy. During the initial deployment phase, the response agent is configured to generate alerts only during calibration; automatic blocking is intentionally excluded. This human-in-the-loop policy is designed to minimize collateral damage during model calibration.

Offline Learning and Online Adaptation

All initial model training and hyperparameter optimization are conducted offline. Following deployment, the system archives predictions, model versions, explanations, and analyst feedback. Confirmed false positives and false negatives are collected for scheduled retraining cycles; however, no model is replaced until it successfully passes the same suite of validation and regression tests as the incumbent version. Drift monitoring encompasses analysis of changes in feature distributions, score distributions, class proportions, and error rates. This controlled update process is favored over unrestricted online adaptation, particularly in security-sensitive contexts where poisoned or erroneous feedback could otherwise compromise the integrity of the detection system.

Implementation Architecture

Each agent is implemented as a Java Spring Boot microservice that uses asynchronous, event-driven communication. Spring Cloud Stream and Kafka facilitate publish-subscribe messaging, while shared schemas ensure that correlation identifiers persist across agents. PostgreSQL is used to store alerts, feedback, and model metadata; Redis supports ephemeral baseline state management. Classical models may be developed in Python and exported to ONNX for Java-based inference or alternatively implemented using Java libraries such as Tribuo. Spring Batch is utilized for offline preprocessing and retraining workflows. The architecture maintains its agent-oriented character, as each service is defined by a distinct objective, localized state, independent lifecycle, and an explicit communication protocol.[16]

Validation Protocol

Data are partitioned into training, validation, and independent test sets based on capture session, day, or scenario whenever feasible, thereby minimizing the risk of random row-level leakage between highly correlated flows. Hyperparameter tuning and feature selection are conducted within an inner cross-validation loop, while performance estimation is performed in an outer cross-validation loop or on a held-out test set. The evaluation metrics reported include per-class precision, recall, F1 score, macro-F1, weighted-F1, false positive rate, false negative rate, precision-recall area, ROC area, Brier

score, calibration error, and confusion matrices. Bootstrap confidence intervals are employed to quantify and communicate statistical uncertainty.

Cross-Dataset and Temporal Evaluation

A model that demonstrates success exclusively on its development dataset fails to meet the research objective. Accordingly, feature-compatible models are evaluated on external datasets both without retraining and subsequent to controlled domain adaptation. Time-ordered evaluations are conducted to quantify performance degradation as the temporal gap between training and evaluation widens. Benign flash-crowd scenarios are incorporated to ensure that the detection system does not erroneously classify all substantial traffic surges as attacks, thereby maintaining operational utility.[17]

Agent-Ablation Experiments

The complete system is systematically compared with modified versions in which classification, anomaly detection, baseline modeling, or rule-based evidence is omitted individually. Variations in macro-F1, attack-specific recall, false-positive rate, calibration, and latency are used to quantify each agent's marginal contribution. Analysis of pairwise disagreement rates and score correlations elucidates whether the agents provide genuinely complementary information or predominantly redundant signals. Such experimental evaluations are essential, as the visual plausibility of a multi-agent architecture alone does not constitute empirical validation of its effectiveness.

Bias Controls

Dataset bias is mitigated through cross-dataset and temporal evaluation protocols. Feature bias is investigated via feature-group ablation studies and importance analyses. Threshold bias is managed through validation-based calibration procedures and cost curve analyses. Class-imbalance bias is addressed by employing per-class performance metrics and restricting resampling or weighting strategies to the training phase. Environmental bias is assessed by withholding laboratory-specific identifiers and evaluating across diverse network topologies. Feedback bias is reduced by segregating analyst reviews from the untouched test set and systematically recording reviewer confidence and instances of disagreement.

Reproducibility and Safety

Each experiment systematically documents the dataset version, preprocessing configuration, feature schema, random seed, model parameters, fusion weights, threshold settings, and software version. The Figure 1. implementation exclusively uses public datasets, synthetic records, or authorized passive telemetry and does not generate harmful traffic directed at external systems. This constraint serves as both an ethical imperative and a methodological advantage, ensuring the repeatability of experiments without relying on uncontrolled offensive activities.[18]

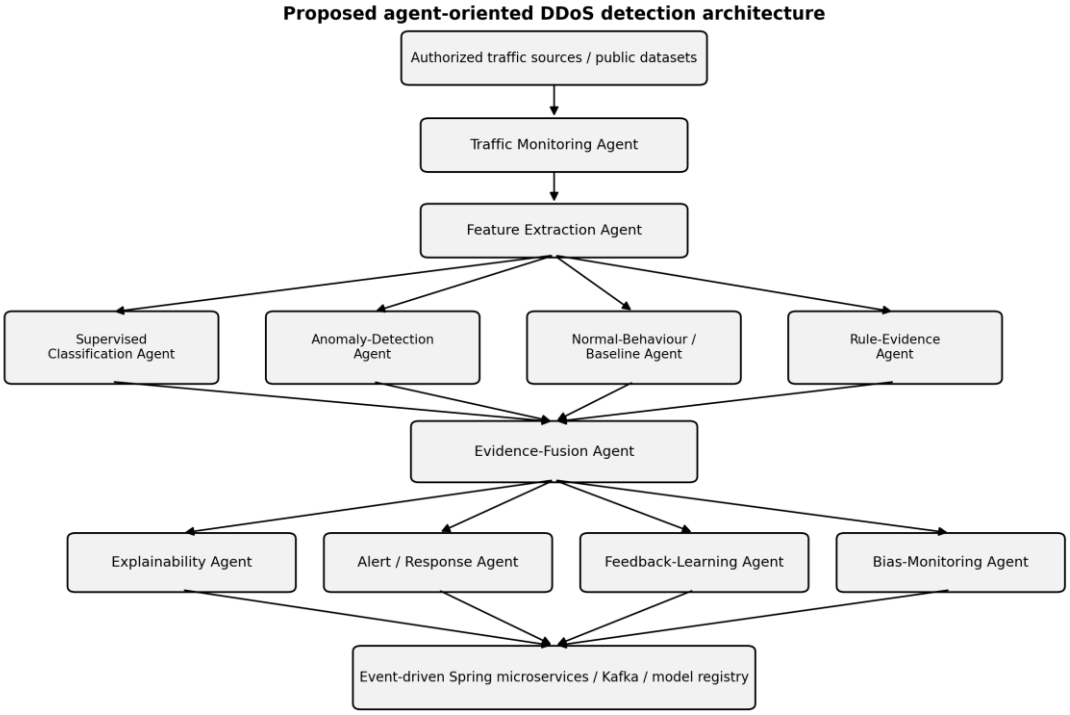

Figure 1. Proposed agent-oriented adaptive DDoS detection architecture.

The proposed architecture establishes a testable intermediary between a monolithic classifier and an unconstrained ensemble of detectors. The four evidence agents are intentionally endowed with distinct inductive biases. Supervised classification is expected to provide the most salient signal for attack families present in the training data. The anomaly detection agent is designed to respond to traffic that falls outside the learned representational space. The baseline agent contributes service- and time-specific contextual information, whereas the rule-based agent supplies deterministic evidence that can be directly audited. Consequently, evidence fusion is not a simple model-averaging procedure; rather, it constitutes a structured approach to integrating complementary explanations of the same observation.

The primary mechanism for bias reduction is diversification in conjunction with independent evaluation. A single model may retain bias despite achieving high average accuracy. Within the multi-agent framework, the occurrence of an anomalously high classification probability, coupled with low anomaly, baseline, and rule scores, manifests as inter-agent disagreement rather than as an unchallenged verdict. Such a pattern may suggest model overconfidence, training-set artifacts, or a legitimate yet familiar traffic surge. Conversely, elevated anomaly and baseline scores, combined with limited evidence of classification, may reflect the emergence of a novel attack, a benign workload fluctuation, or concept drift. The system does not resolve these ambiguities autonomously; instead, it preserves them for subsequent validation and analyst review.[19]

The Normal-Behavior Agent plays a critical role in mitigating flash-crowd-induced false positives. Global thresholding mechanisms fail to account for the distinct baseline traffic rates and temporal patterns characteristic of services such as online marketplaces, payment gateways, and internal administrative systems. A destination-specific baseline allows the same packet rate to yield different evidentiary interpretations, contingent on

historical context. Nonetheless, the baseline agent is susceptible to bias if its learning process coincides with an ongoing attack or if normal behavioral patterns shift abruptly. Consequently, the implementation of robust update rules, the exclusion of confirmed incidents from the training data, and the deployment of drift-detection alarms are essential.

Transparent rule-based evidence enhances accountability but must remain appropriately constrained. Rules predicated on elevated SYN-to-ACK ratios, excessive UDP concentration, or pronounced source diversity encapsulate protocol-level knowledge that statistical models may fail to capture. However, fixed rules are inherently brittle and may embody outdated assumptions. Within the proposed framework, the primary function of such rules is to provide interpretable evidence, rather than to unilaterally determine the final decision. Rule weights and thresholds are systematically versioned and evaluated in a manner analogous to model parameters.

Evidence fusion introduces distinct modeling risks

A fixed weighted sum offers interpretability and facilitates auditing; however, it presumes stable relative contributions across diverse attack types and operational environments. Learned stacking approaches can dynamically adjust these weights, yet they risk overfitting and may obscure the relationship between input features and final decisions. Accordingly, the experimental design adopts transparent weighted fusion as the primary baseline and benchmarks it against a calibrated meta-classifier. Sensitivity analyses are conducted to document how conclusions vary as weights are perturbed within plausible ranges.

The microservice-based implementation confers operational advantages that are directly pertinent to research endeavors. Each agent may be independently replaced, scaled, disabled, or evaluated. For instance, a high-throughput classification service may require deploying additional replicas compared to the explanation service, whereas a baseline agent may retain localized state for a specific network segment. Event-driven communication architecture enables the replay of identical traffic windows across multiple model versions, thereby enhancing experimental reproducibility. However, this flexibility introduces greater architectural complexity: message ordering, schema evolution, failure recovery, and cross-agent latency must be empirically quantified rather than presumed to be negligible.

The bias-monitoring component is deliberately segregated from the detection process. Its function is not to exert direct influence on live verdicts, but rather to assess whether errors are disproportionately concentrated within specific attack classes, temporal intervals, services, or model versions. This component is tasked with generating per-class confusion matrices, precision-recall summaries, calibration plots, drift indicators, and ablation analyses. While an analyst console may subsequently aggregate evidence by attack type, such descriptive visualizations pertain to separate interpretability studies and should not be conflated with formal accuracy measurement.

Several hypotheses are derived from the proposed architecture

First, the complete multi-agent system is expected to reduce false positives from benign surges compared with a volume-dominated classifier. Second, the inclusion of anomaly and baseline agents should enhance recall for attacks that diverge from the supervised training distribution. Third, rule-based evidence is anticipated to improve interpretability, even if its direct impact on aggregate F1 score remains modest. Fourth,

while a learned fusion model may yield improved average performance, it may also compromise calibration or robustness under dataset shift. These hypotheses require validation through controlled experiments; they cannot be substantiated by architectural design alone.

The most compelling evidence for the proposed architecture will come from consistent performance improvements across distinct data partitions and diverse datasets, rather than from isolated instances of high accuracy. Should the multi-agent system enhance performance on CIC-DDoS2019 while degrading on Bot-IoT or CSE-CIC-IDS2018, the purported robustness would not be substantiated. Similarly, if ablation studies reveal that two agents provide no discernible benefit, architectural simplification would be warranted. Accordingly, this research conceptualizes modularity as a methodological tool for rigorous comparative evaluation, rather than as evidence that increased architectural complexity is inherently advantageous.[20]

4. Conclusion

This paper has introduced an agent-oriented, adaptive machine learning framework for bias-aware Distributed Denial-of-Service (DDoS) attack detection. The proposed architecture decomposes traffic monitoring, feature extraction, supervised classification, anomaly detection, normal-behavior modeling, rule-based evidence, evidence fusion, explanation, feedback, and bias monitoring into independently testable agents. The final risk score is derived from four normalized sources of evidence, rather than relying on a single opaque classifier.

The framework systematically addresses multiple limitations inherent to conventional benchmark-oriented detection methodologies. Dataset bias is mitigated by employing cross-dataset and temporal evaluation protocols; feature bias is assessed via feature-group ablation studies; class imbalance is managed through per-class and precision-recall performance metrics; threshold bias is controlled through calibration techniques and cost-sensitive validation; and temporal bias is addressed via drift monitoring and controlled retraining procedures. Explanatory outputs and analyst feedback are preserved as resources for auditing and iterative learning, yet neither is permitted to influence the integrity of the independent testing process.

The proposed Spring-based microservice implementation offers a pragmatic transition from agent-oriented theory to deployable software systems. Asynchronous event handling preserves loose coupling among components, while model adapters enable the integration of classifiers trained in disparate ecosystems within a unified pipeline. This architectural approach facilitates replication, replacement of individual agents, controlled traffic replay, and precise measurement of system latency. It also introduces engineering imperatives, such as schema governance, secure messaging, model version control, and robust fault management.

Future research should focus on implementing the three proposed systems, conducting nested model selection procedures, evaluating cross-dataset generalization, and performing agent-ablation experiments. The primary objective is to establish robust single-model baselines, as the absence of such benchmarks precludes meaningful attribution of improvements to the multi-agent architecture. The secondary objective involves calibrating evidence scores and fusion thresholds. The tertiary objective is to assess system performance under benign flash crowd scenarios and temporal drift, as these

conditions are critical for determining whether the system has genuinely learned attack behaviors or is simply reacting to elevated traffic volumes.

In conclusion, agent orientation does not inherently confer unbiasedness or accuracy upon a DDoS detection system. Its principal value lies in establishing explicit boundaries of responsibility, preserving competing sources of evidence, and facilitating the measurement of model dependence. When integrated with rigorous validation protocols, transparent evidence fusion, and controlled feedback mechanisms, the proposed framework provides a robust foundation for adaptive and explainable DDoS detection in cloud, enterprise, and IoT environments.

REFERENCES

[1] R. Abu Bakar, X. Huang, M. S. Javed, S. Hussain, and M. F. Majeed, "An intelligent agent-based detection system for DDoS attacks using automatic feature extraction and selection," *Sensors*, vol. 23, no. 6, p. 3333, 2023.

[2] J. S. Balasubramaniyan, J. O. Garcia-Fernandez, D. Isacoff, E. Spafford, and D. Zamboni, "An architecture for intrusion detection using autonomous agents," in *Proc. 14th Annual Computer Security Applications Conf.*, 1998, pp. 13–24.

[3] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Chichester, UK: Wiley, 2007.

[4] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[5] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[6] 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.

[7] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.

[8] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Security and Privacy Workshops*, 2018, pp. 29–35.

[9] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, Art. 44, 2014.

[10] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Machine Learning*, 2017, pp. 1321–1330.

[11] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[12] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int. Joint Conf. Artificial Intelligence*, 1995, pp. 1137–1144.

[13] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of a realistic botnet dataset in IoT: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[14] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 413–422.

[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, 2017.

[16] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[17] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. Military Communications and Information Systems Conf.*, 2015, pp. 1–6.

[18] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[19] 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.*, 2016, pp. 1135–1144.

[20] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial-of-service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.