Multi-Agent Crowd-Risk Prioritization for Large-Scale Events: A Hybrid ML–Agent Approach

Najwa Samrgandi
Department of Data Science-College of Computing, Umm Al-Qura University, Makkah, Saudi Arabia

***Abstract*—Managing crowd safety during large-scale gatherings such as Hajj requires timely assessment of multiple risk factors, including crowd conditions, environmental factors, and health-related indicators. Many existing approaches address these aspects separately, making it difficult to obtain a unified operational view of risk. This study presents a multi-agent framework that combines machine learning with domain-specific agent coordination for crowd-risk prioritization. The framework includes five agents responsible for crowd monitoring, health monitoring, environmental sensing, coordination, and alert generation. To reduce data leakage, variables used in risk-label construction and post-event outcomes were excluded before model training, resulting in 17 real-time observable features. The framework was evaluated using the Hajj and Umrah Crowd Management dataset containing 10,000 simulated records. On the independent test set, XGBoost+SMOTE achieved 47.30% accuracy and 46.22% F1-score; the XGBoost baseline achieved 57.15% accuracy and 45.96% F1-score, and a majority-class baseline that always predicts the dominant class achieved 58.70%, underscoring that overall accuracy is an insufficient metric for this task. Although accuracy decreased after class rebalancing, minority-class sensitivity improved: XGBoost+SMOTE achieved partial High-risk recognition (recall = 0.14, F1 = 0.12), whereas the XGBoost baseline and Cost-Sensitive XGBoost failed to identify any High-risk cases (recall = 0.00). McNemar's test confirmed a statistically significant difference between the XGBoost baseline and the SMOTE-enhanced model ($\chi^2 = 69.218$, $p < 0.001$). Cross-validation on the SMOTE-balanced training data produced 59.89% ± 0.20% accuracy and 58.94% ± 0.17% F1-score. A streaming evaluation on 500 held-out test records showed that the Full Framework generated 93 High-risk alerts, compared with 74 from the ML-only configuration and 23 from the Agents-only configuration, a 25.7% increase relative to ML-only operation. A precision check against the internal risk-score proxy found that none of the 19 agent-driven escalations beyond ML-only predictions corresponded to a High-risk label under this proxy; these alerts should accordingly be read as operational risk-prioritization signals rather than confirmed incident detections. All reported figures should further be interpreted in light of the dataset's synthetic origin: because features were algorithmically generated rather than collected from real sensors or incidents, the results are illustrative of the framework's behavior under these specific simulated conditions rather than predictive of real operational deployment. Average processing latency was 4.127 ms per record under sequential single-stream conditions. In addition to the proposed framework, the study presents a diagnostic evaluation approach that combines leakage control, dual association analysis, and direct precision auditing. This approach may also prove useful for evaluating similar ML–agent systems in other safety-critical alerting applications.**

***Keywords*—Multi-agent systems; crowd-risk prioritization; intelligent crowd management; real-time IoT monitoring; SMOTE; XGBoost; safety-critical systems**

I. Introduction

Large public gatherings represent one of the most operationally complex environments for safety management. The annual Hajj pilgrimage, which draws more than two million participants to confined geographic zones over a compressed timeframe, exemplifies the challenges that arise when crowd density, environmental stress, and health vulnerability intersect. The scale and operational complexity of Hajj continue to motivate the development of intelligent crowd-management systems capable of supporting timely risk assessment and decision-making [1], [2].

The past decade has seen considerable progress in AI-driven crowd analysis. Vision-based systems using YOLO and CNN architectures have achieved strong performance in density estimation and abnormal behavior detection [3], [4], [5]. IoT sensor networks have extended monitoring beyond camera fields of view by capturing environmental and operational indicators associated with crowd conditions [6], [7]. Multi-agent frameworks have demonstrated value in evacuation simulation and distributed coordination [8], [9]. Yet these advances have largely evolved in parallel rather than in concert: visual monitoring systems rarely incorporate health indicators; IoT platforms seldom feed predictive risk models; and agent-based systems tend toward simulation rather than real-time operational deployment [10], [11].

This fragmentation carries real operational cost. A monitoring system that detects crowd compression via camera feeds cannot independently account for the additional risk posed by extreme heat, an aging pilgrim population, or extended waiting times at transport checkpoints. Conversely, rule-based alert systems operating without predictive components lack the capacity to differentiate between momentary crowd fluctuations and genuinely escalating risk trajectories; predictive agent-based architectures have been shown to address this limitation more effectively [12], [13].

This study addresses this gap through a framework designed around five principles: 1) asynchronous agent independence: each agent processes its own data stream without blocking on others; 2) leakage-free feature engineering: the feature space used for ML prediction is strictly isolated from the variables used to construct risk labels; 3) operationally grounded escalation: the coordination policy prioritizes sensitivity over specificity, reflecting the asymmetric cost of missing High-risk events; 4) statistical accountability: model improvements are

validated through McNemar's test; and 5) real-time feasibility: all predictions are generated with sub-10 ms latency per record under the evaluated experimental conditions.

The study is guided by three research questions: (RQ1) How can asynchronous agent coordination improve crowd-risk prioritization beyond ML prediction alone? (RQ2) What feature-selection strategy best prevents leakage in operationally constructed risk labels? (RQ3) How does the proposed framework perform under realistic class imbalance without artificially inflating reported performance?

II. Related Work

Crowd safety research has matured across three intersecting streams. The first concerns AI-based crowd analysis. Vision-based architectures, particularly YOLOv8 and YOLOv9, have achieved near-real-time detection of density anomalies and abnormal behaviors [3], [14], [15], [16]. CNN-based approaches extended this to behavioral trajectory prediction [17], [18]. Cyber-physical integrations further embedded visual monitoring within broader operational sensing architectures [19], [20]. However, these systems remain primarily reactive: they detect what is already happening rather than flagging converging risk conditions.

The second stream concerns IoT-enabled monitoring. Distributed sensor networks capture environmental variables that co-occur with crowd incidents but fall outside camera fields of view [21], [22], [23]. Zhang et al. proposed an IoT-based public safety alert system that combines distributed multi-sensor inputs with lightweight edge-deployed machine learning models to generate alerts, demonstrating the feasibility of automated, low-latency alert pipelines for emergency response support [7]. While [7] incorporates basic predictive classification at the sensor level, it does not address multi-domain risk fusion across crowd, health, and environmental indicators simultaneously, nor does it consider data-leakage risks in label construction. The broader limitation across this stream of work is a continued reliance on per-sensor threshold or single-model logic rather than coordinated multi-agent risk fusion across heterogeneous domains [24].

The third stream concerns multi-agent coordination. Keykhaei et al. developed a situation-aware multi-agent evacuation model for earthquake scenarios, demonstrating that distributed agent coordination enables more adaptive response than static centralized planning in dynamic crowd conditions [9]. Dumitrescu et al. used multi-agent modeling to simulate crowd panic propagation [8]. More recent work has explored LLM-based agents for smart city coordination [12] and multi-agent machine learning frameworks combining classification with coordinated decision-making for real-time safety response [25]. Despite this breadth, most multi-agent applications remain simulation-oriented, with few validated deployments in operational real-time monitoring systems [26], [27], [28], [29].

The most directly comparable work is CIRS [25], a five-agent machine learning-based framework for real-time accident detection and emergency response, whose architecture (perception, classification, description, communication, and decision-making agents) closely parallels the five-agent design adopted in the present framework (Crowd Monitoring, Health Monitoring, Environmental Sensing, Coordinator, and Alert agents). The present framework differs from [25] in several respects. First, [25] is applied to a structurally distinct domain (road-traffic accident detection from CCTV video) rather than multi-domain crowd-risk prioritization, and accordingly integrates a vision-language model for incident description rather than tabular sensor-derived risk classification. Second, [25] reports classifier performance (YOLOv11: precision 0.866, recall 0.855) without an explicit discussion of label-construction leakage, since its ground truth (manually annotated accident frames) is independent of its input features by design, whereas the present framework addresses a leakage risk that is structurally specific to operationally derived risk labels. Third, [25] does not report a formal statistical test of agreement between its decision-making agent and its underlying classifier, an aspect the present framework addresses directly through McNemar's test (Section IV-C). A complementary comparison point is provided by Alshamrani et al. [30], who developed a wearable-sensor-based system for monitoring pilgrim stress and fatigue during Hajj, combining physiological, geospatial, and self-reported data; unlike the present framework, [30] is oriented toward descriptive analysis of collected pilgrim data rather than real-time multi-agent risk coordination, and does not incorporate a predictive classification component or an agent-based architecture. Taken together, [25] offers the closest architectural parallel (multi-agent coordination with an embedded ML classifier) while [30] offers the closest domain parallel (Hajj-specific physiological and environmental monitoring); neither addresses both dimensions simultaneously, which motivates the integrated evaluation presented in this study. These comparisons should be interpreted as descriptive positioning relative to the closest available published work rather than as a direct empirical benchmark. Table I summarizes the major research streams and their core limitations.

TABLE I. Summary of Major Research Streams and their Limitations.

Research Stream	Core Limitation	Key References
AI/deep learning crowd monitoring	Reactive; limited integration with health or environmental sensing	[3]-[5],[14]-[20]
IoT-enabled sensing and alerting	Threshold-based; no adaptive learning; no multi-source fusion	[6],[7],[21]-[24]
Multi-agent coordination frameworks	Simulation-oriented; limited real-time operational deployment	[8],[9],[12],[25]-[29]

Theoretical perspectives on multi-agent decision-making [31] and a broader multi-expert analysis of AI agentic systems and their deployment characteristics [32] further contextualize the coordination architecture adopted in this framework. Distinct from this general perspective, a multi-source information fusion technique specifically targeting abnormal crowd behavior detection [33] reinforces the operational complementarity of data-driven and rule-based sensing that underpins the proposed design. The review of prior studies suggests that current approaches address specific dimensions of crowd safety but rarely combine them within a single real-time architecture. This gap motivates the development of a framework that integrates predictive machine learning,

heterogeneous sensing, and coordinated agent-based responses to support proactive risk management.

III. METHODOLOGY

A. Dataset and Representativeness

The Hajj and Umrah Crowd Management dataset (Kaggle) comprises 10,000 records across 30 features capturing spatial coordinates, environmental readings, operational waiting times, health indicators, and pilgrim demographic profiles. The dataset contains no missing values and no duplicate records. All features were generated under simulated operational conditions designed to reflect common Hajj crowd management scenarios; their ability to fully represent real-world crowd behavior remains a study limitation.

Risk labels were derived through a weighted operational scoring function: Crowd_Density (0.25), Temperature (0.20), Fatigue_Level (0.15), Stress_Level (0.15), Queue_Time_minutes (0.10), Emergency_Event (0.10), and Distance_Between_People_m (0.05). The resulting score was discretized into three thresholds: Low (below 1.8), Medium (1.8–2.4), and High (above 2.4), yielding 3,121 Low (31.2%), 5,869 Medium (58.7%), and 1,010 High (10.1%) records. These labels represent operationally constructed risk proxies rather than verified emergency records. The framework is accordingly best characterized as a risk-prioritization tool.

B. Feature Engineering and Leakage Prevention

Data leakage represents one of the most consequential validity threats in applied machine learning research, capable of inflating reported accuracy without reflecting genuine predictive capacity [34], [35]. In this dataset, the risk is structurally elevated: the seven features used to compute the risk label were excluded from model input. Post-event outcome variables (Satisfaction_Rating, Perceived_Safety_Rating, Pilgrim_Experience, AR_Navigation_Success) were also excluded as they are unavailable at operational prediction time.

The resulting clean feature set comprises 17 real-time observable indicators: spatial (Location_Lat, Location_Long), movement (Movement_Speed), environmental (Weather_Conditions, Sound_Level_dB), operational (Activity_Type, Event_Type, Waiting_Time_for_Transport, Security_Checkpoint_Wait_Time, Time_Spent_at_Location_minutes, Interaction_Frequency), pilgrim profile (Age_Group, Health_Condition, Transport_Mode), and contextual indicators (Incident_Type, Crowd_Morale, AR_System_Interaction).

C. Data Partitioning and Class Balancing

The dataset was divided into 80% training (8,000 records) and 20% test (2,000 records) using stratified random splitting with seed 42. The test set was held out entirely throughout training, tuning, and oversampling. SMOTE [36] was applied exclusively to the training set following the split to address class imbalance [35], expanding the minority High and Low classes to match the majority Medium class at 4,695 samples each. Cross-validation on the SMOTE-augmented training set produces accuracy estimates under balanced class conditions (33.3% per class), which naturally exceed test-set accuracy under the original imbalanced distribution. Both metrics are reported with their interpretations distinguished in Section IV.

D. Prediction Models and Experimental Setup

Seven configurations were evaluated: Random Forest [37] (n=150, depth=10), XGBoost baseline [38] (n=200, depth=6, lr=0.05), LightGBM baseline (n=200, depth=6, lr=0.05), RF+SMOTE, XGBoost+SMOTE, LightGBM+SMOTE, and Cost-Sensitive XGBoost (scale_pos_weight calibrated to the High-to-Medium class ratio). LightGBM was included to assess whether gradient-boosting performance generalizes across implementations; Cost-Sensitive XGBoost was included to test class-weighting as an alternative to oversampling for minority-class detection. Cost-sensitive learning addresses class imbalance by assigning higher penalties to misclassification of minority-class instances, thereby encouraging the model to place greater emphasis on underrepresented classes [39]. All models used seed 42. Weighted-average and per-class precision, recall, and F1-score were computed to expose behavior on the minority High-risk class. McNemar's test with continuity correction assessed statistical significance between XGBoost baseline and XGBoost+SMOTE at alpha = 0.05, following established recommendations for comparing classifier performance on a single shared test set [40].

E. Multi-Agent Framework Architecture

The framework comprises five agents implemented as independent Python threads communicating through dedicated in-memory message queues, consistent with the reactive agent architecture described by Wooldridge [41] and the general multi-agent coordination workflow surveyed by Maldonado et al. [26]. The Crowd Monitoring Agent observes Movement_Speed, Interaction_Frequency, and Crowd_Morale. It applies threshold rules synthesized from crowd dynamics simulation literature [8], [9]: a Medium signal is triggered when movement speed falls below 0.5 m/s, consistent with documented pre-compression crowd conditions, or when interaction frequency exceeds 7; a High signal is triggered when speed falls below 0.3 m/s and frequency exceeds 8 simultaneously. These thresholds represent expert-informed approximations derived from simulation findings and are subject to empirical calibration against real incident data.

The Health Monitoring Agent observes Health_Condition, Age_Group, and Waiting_Time_for_Transport. Pilgrims presenting dehydration or fatigue, those aged 70 or above, or those waiting more than 90 minutes for transport receive elevated health risk signals. These thresholds were informed by health management frameworks for crowded events [42] and constitute expert-informed approximations subject to empirical calibration. The combination of dehydration and age above 70 triggers High. The Environmental Sensing Agent observes Weather_Conditions, Sound_Level_dB, and Security_Checkpoint_Wait_Time. Adverse weather or sound levels above 80 dB trigger Medium; sandstorm with sound above 85 dB triggers High. All agent thresholds are expert-informed approximations pending empirical validation.

The Coordinator Agent blocks on three message queues until signals from all monitoring agents arrive, then queries the XGBoost+SMOTE model with the encoded feature vector. The final risk decision applies a maximum-escalation policy: the

highest risk level across all four inputs is selected. The Alert Agent converts the final decision into an operational notification.

The proposed architecture differs structurally from a conventional centralized pipeline in several respects. Table II summarizes the key architectural distinctions.

TABLE II. ARCHITECTURAL COMPARISON: CENTRALIZED PIPELINE VS. PROPOSED FRAMEWORK.

Centralized Pipeline	Proposed Framework
Single processing module	Five domain-specific agents
Shared decision logic	Independent domain-specific assessment
Tightly coupled components	Modular, independently operable components
Limited extensibility	New agents can be added without modifying existing ones
Sequential domain processing	Asynchronous domain-specific monitoring

The present framework does not claim decentralized negotiation or autonomous multi-agent planning. Rather, it adopts a modular, coordinator-based multi-agent architecture in which independent domain-specific agents perform local assessment before a coordinator integrates their outputs through a conservative escalation policy [13]. The architecture was selected for modularity, extensibility, and separation of domain-specific reasoning rather than to claim computational superiority over an equivalent centralized implementation. This design allows each agent to be developed, tested, and replaced independently, and permits the addition of new sensing domains without restructuring the coordination logic.

In terms of computational complexity, each monitoring agent evaluates threshold conditions over a fixed three-feature subset, contributing O(1) time per agent and per record. The Coordinator Agent then performs a single XGBoost inference over 17 features. Gradient-boosted tree inference requires $O(T \cdot D)$ time per record, where T is the number of trees and D is the maximum tree depth. Because T and D are fixed hyperparameters in the present implementation ($T = 200$, $D = 6$), the inference cost is constant with respect to dataset size and the number of monitoring agents. Inter-agent communication overhead scales as O(A), where A denotes the number of monitoring agents. The overall per-record complexity is therefore $O(A + T \cdot D)$; with A, T, and D fixed in the current five-agent architecture, it is O(1) with respect to dataset size, while increasing linearly if the number of monitoring agents is expanded. The framework architecture is illustrated in Fig. 1.

In the architecture of the proposed multi-agent framework, Independent Crowd, Health, and Environmental agents publish risk signals through dedicated message queues. The Coordinator Agent integrates these signals with XGBoost+SMOTE predictions using a maximum-escalation policy to generate the final risk assessment. Results are based on 500 test records held out from the training data (see Section III-F for the relationship between this streaming subset and the broader test set used in Section IV-A).

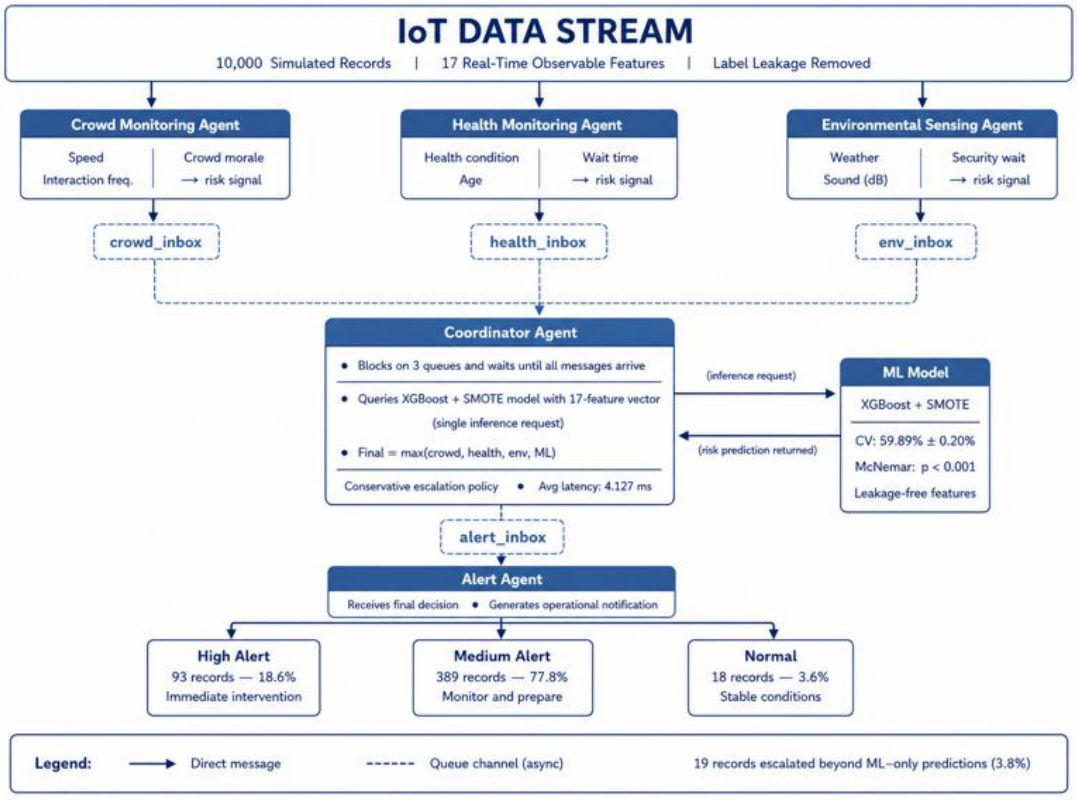

Fig. 1. Proposed multi-agent framework for crowd-risk prioritization.

F. Evaluation Protocol

Model performance was assessed on the held-out test set using overall and per-class metrics. Robustness was evaluated through 3-fold stratified cross-validation on the SMOTE-augmented training set, reporting Mean plus or minus Std. McNemar's test with continuity correction assessed statistical significance. The multi-agent framework was evaluated in three configurations (ML-only, Agents-only, and Full Framework) on 500 records drawn exclusively from the held-out test set, ordered by Timestamp to simulate a streaming environment; these 500 records are a subset of the same 2,000-record test set used for the overall and per-class results reported in Section IV- A, rather than an independently drawn fresh sample, and the ablation results should accordingly be read as a re-examination of the existing held-out test set from a streaming perspective rather than as validation on new, previously unseen data. Sensitivity analysis was conducted by varying all agent thresholds uniformly by plus or minus 20% from baseline values and computing High-risk alert counts experimentally across the three threshold configurations.

IV. Results and Discussion

A. Model Performance

Table III presents overall test-set results for seven configurations. Table IV presents per-class results for the XGBoost and LightGBM SMOTE-enhanced configurations. On the test set, XGBoost+SMOTE achieved 47.30% accuracy (F1: 46.22%); cross-validation on the SMOTE-balanced training data yielded 59.89% plus or minus 0.20% accuracy and 58.94% plus or minus 0.17% F1 for XGBoost+SMOTE. The gap between CV accuracy (59.89%) and test-set accuracy (47.30%) reflects evaluation under balanced (33.3% per class) versus imbalanced (Medium 58.7%) class conditions, respectively, and is reported explicitly to avoid misrepresentation of operational performance. To quantify the stability of test-set performance beyond the single fixed split, non-parametric bootstrap resampling (1,000 resamples with replacement, drawn from the test-set predictions) was used to construct 95% confidence intervals: accuracy 47.35% [45.20%, 49.55%], F1-score 46.26% [43.96%, 48.59%], and High-risk recall 13.76% [9.23%, 18.60%]. These intervals indicate that the reported point estimates are reasonably stable under resampling, while the width of the High-risk recall interval reflects the limited number of High-risk test instances available for evaluation.

Table IV reveals a critical finding: the XGBoost baseline and Cost-Sensitive XGBoost detected zero High-risk cases (precision = 0.00, recall = 0.00, F1 = 0.00), while XGBoost+SMOTE achieved partial High-risk detection (recall = 0.14, F1 = 0.12) and LightGBM+SMOTE achieved comparable performance (recall = 0.12). The baseline model's single false positive prediction for the High-risk class (1 of 1,174 true Medium-risk records misclassified as High) was insufficient to register a non-zero precision value at two decimal places, confirming that the model produced no usable High-risk signal whatsoever on the independent test set. The inability of baseline models to detect any High-risk instance is not a model failure per se but reflects the intrinsic difficulty of the task: after leakage removal, the observable features provide insufficient discriminative signal to separate High- from Medium-risk without class rebalancing. This finding directly motivates the domain-aware agent escalation layer, which captures High-risk configurations through rule-based domain knowledge that the baseline ML classifier does not detect under the current feature set. This underscores that overall accuracy is an insufficient metric for safety-critical systems. McNemar's test confirmed a statistically significant improvement of XGBoost+SMOTE over the baseline (chi-squared = 69.218, p less than 0.001), indicating a statistically significant shift in minority-class sensitivity, mechanistically explained in Section V as a decision-boundary effect rather than a genuine signal-recovery effect on the independent test set.

To contextualize these results against trivial reference points, two simple baselines were also evaluated on the same test set: a majority-class baseline that always predicts the dominant class (Medium) and a stratified-random baseline that predicts classes according to their training-set proportions. The majority-class baseline achieved 58.70% accuracy and 43.42% F1-score (Table III); the stratified-random baseline achieved 42.20% accuracy and 42.31% F1-score. Notably, the majority-class baseline's accuracy (58.70%) exceeds that of the XGBoost baseline (57.15%), although the XGBoost baseline achieves a higher F1-score (45.96% vs. 43.42%) by producing non-trivial, if ultimately unsuccessful, predictions for the Low-risk class. This comparison reinforces rather than undermines the study's central argument: it provides direct, model-agnostic confirmation that overall accuracy is an insufficient and potentially misleading metric for this task, since a model that makes no use of any observable feature can match or exceed the accuracy of a trained classifier, consistent with the near-absence of linear and nonlinear feature–label association documented in Section V. The comparison is included here, prior to that analysis, specifically to establish this point empirically before it is explained mechanistically. The headline XGBoost+SMOTE figures reported above were further confirmed to be stable, rather than dependent on the specific random seed used, via a repeated-splits analysis across 10 independent seeds reported in Section VII (Statistical Validity).

TABLE III. Overall Model Performance on Independent Test Set

Model	Acc.%	Prec.%	Rec.%	F1%
Random Forest	58.75	50.10	58.75	43.64
XGBoost Baseline	57.15	45.12	57.15	45.96
RF + SMOTE	48.50	45.81	48.50	44.62
XGBoost + SMOTE	47.30	45.90	47.30	46.22
LightGBM Baseline	57.00	44.40	57.00	45.72
LightGBM + SMOTE	46.55	44.93	46.55	45.34
XGBoost Cost-Sensitive	57.15	45.12	57.15	45.96
Majority-Class Baseline	58.70	34.46	58.70	43.42
Stratified-Random Baseline	42.20	42.31	42.20	42.31

TABLE IV. PER-CLASS PERFORMANCE: XGBOOST MODELS (TEST SET)

Model	Class	Prec.	Recall	F1
XGBoost Baseline	High	0.00	0.00	0.00
	Low	0.34	0.07	0.11
	Medium	0.59	0.94	0.72
XGBoost + SMOTE	High	0.11	0.14	0.12
	Low	0.31	0.22	0.26
	Medium	0.60	0.67	0.63
LightGBM + SMOTE	High	0.10	0.12	0.11
	Low	0.31	0.21	0.25
	Medium	0.59	0.66	0.62

Table IV shows that the baseline XGBoost model failed to identify any High-risk cases, with a recall of 0.00. After applying SMOTE, the model began to detect this class, although performance remained limited, with a recall of 0.14 and an F1-score of 0.12. SMOTE also increased Low-risk recall from 0.07 to 0.22. However, this came at the cost of reducing Medium-risk recall from 0.94 to 0.67. These results indicate that rebalancing reduced the model's dependence on the dominant Medium-risk class and allowed better, though still imperfect, detection of minority risk levels.

B. Feature Importance and Interpretability

Feature importance scores from the XGBoost+SMOTE model show a balanced distribution: the highest-ranked feature (Crowd_Morale, 0.0995) contributes less than 10% of total importance. This contrasts with analyses that included label-derived variables, where a single feature dominated, indicating leakage. Crowd_Morale, Event_Type, and Weather_Conditions are operationally interpretable: crowd morale reflects collective behavioral tension; event type determines spatial organization and density thresholds; weather conditions directly affect crowd movement. Fig. 2 illustrates the top 10 feature importance scores.

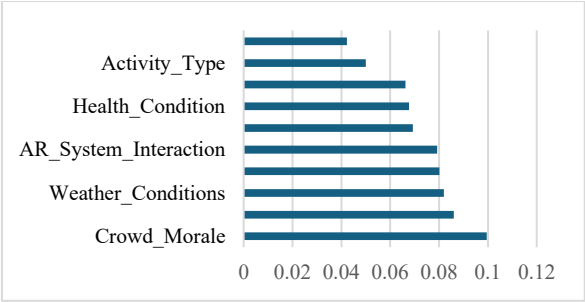

Fig. 2. Top 10 feature importance scores from the XGBoost+SMOTE model.

Because Crowd_Morale emerged as the single most important feature, its operational measurability warrants clarification. In the present dataset, Crowd_Morale is recorded as a discrete categorical indicator (Positive, Neutral, Negative); it is not derived from facial-expression or sentiment analysis, which would introduce separate privacy and accuracy concerns in dense crowd settings. In an operational deployment, this indicator would instead be inferred indirectly from signals already collected by the framework's other sensing channels (for example, aggregate crowd-noise level (Sound_Level_dB), movement irregularity (Movement_Speed variance), and local interaction density (Interaction_Frequency)) rather than measured directly. This indirect-inference approach is consistent with the framework's real-time observability requirement, but it also means that Crowd_Morale should be understood as a composite behavioral proxy rather than a directly sensed physical quantity, and its operational construction from these underlying signals remains an item for future validation.

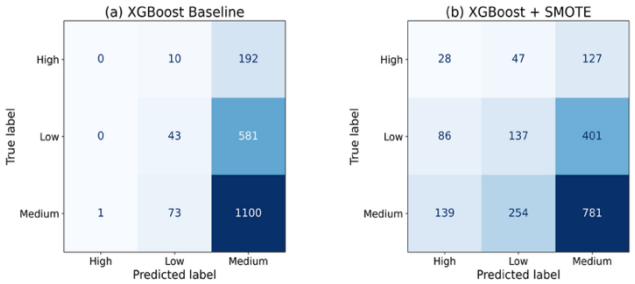

Fig. 3. Confusion matrices for (a) XGBoost baseline and (b) XGBoost+SMOTE on the independent test set.

Fig. 3 shows the confusion matrices for both models. A review of the confusion matrices reveals notable differences between the two models. The original XGBoost classifier predominantly predicted the Medium-risk class, leading to no correct predictions for High-risk observations. After incorporating SMOTE, the classifier identified a small portion of the High-risk cases. Although this increased sensitivity to the minority class, it also resulted in a lower overall accuracy. This pattern is in agreement with the performance metrics reported in Table IV.

To assess discriminative performance across the full range of decision thresholds rather than at the single fixed operating point reported in Table IV, a precision-recall curve was constructed for the High-risk class using the predicted class probabilities from XGBoost+SMOTE on the full 2,000-record independent test set described in Section IV-A (Fig. 4); this is distinct from, and substantially larger than, the 500-record streaming subset used in the ablation and escalation-precision analyses in Section IV-C below. The curve achieved an average precision (AP) of 0.1016, compared with an AP of 0.1010 expected under random ranking at this class's prevalence (10.1% of test-set records). The observed AP exceeds the random-baseline value by only 0.0006, and Fig. 4 shows the model's curve tracking the random-baseline reference line closely across nearly the entire recall range, with the exception of a brief, high-variance fluctuation at very low recall where the small number of underlying records makes individual precision estimates unstable. This result provides a third, independent line of evidence supporting the same conclusion reached via the majority-class baseline comparison above and the mutual information analysis in Section V: the trained classifier's ranking of records by predicted High-risk probability carries negligible discriminative value beyond what would be expected from the class's base rate alone, across the entire space of possible decision thresholds rather than only at the single threshold used to generate Table IV.

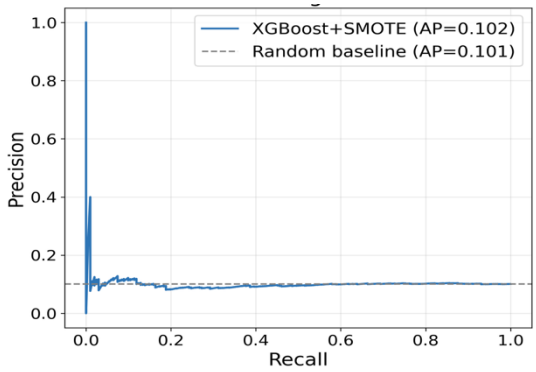

Fig. 4. Precision-Recall curve for High-risk class detection (XGBoost+SMOTE) versus the random-baseline reference line at class prevalence.

C. Multi-Agent Coordination Value and Streaming Performance

Table V presents the ablation results across three configurations on 500 held-out test records ordered by Timestamp. The ML-only configuration generated 74 High-risk alerts, while the Agents-only configuration generated 23 High-risk alerts. The Full Framework generated 93 High-risk alerts, representing a 25.7% increase relative to the ML-only configuration and substantially exceeding either component operating independently. Because the Full Framework applies a maximum-escalation policy (its High-risk count can never fall below either individual component's, by construction; see Table V below), this difference alone does not establish symmetric, bidirectional disagreement between the two components. With this structural property in mind, the pattern is nonetheless consistent with the ML model and the agent layer each surfacing High-risk configurations that the other does not independently flag, suggesting complementary, if asymmetric, risk-assessment mechanisms. The ML model performs probabilistic classification across all 17 observable features, whereas the agents apply domain-specific escalation rules to targeted feature subsets, allowing each component to identify elevated-risk situations that may not be emphasized by the other.

TABLE V. ABLATION: HIGH-RISK ALERT GENERATION ACROSS THREE CONFIGURATIONS.

Configuration	High-Risk Alerts	Records Escalated	Avg. Latency
ML Only (XGBoost+SMOTE)	74	—	4.127 ms
Agents Only (rules, no ML)	23	N/A	N/A
Full Framework (ML + Agents)	93	19 (3.8%)	4.127 ms

Because the Full Framework configuration cannot, by construction, generate fewer High-risk alerts than either individual component on any given record, the discordance pattern between configurations is necessarily one-directional rather than a conventional bidirectional disagreement (see Table VI for full contingency tables). With this structural constraint in mind, the 70 additional High-risk alerts generated by the Full Framework beyond the Agents-only configuration (93 vs. 23) were confirmed via McNemar's test to represent a statistically significant difference ($\chi^2 = 68.01$, $p < 0.001$), and the 19 additional alerts generated beyond the ML-only configuration (93 vs. 74) were likewise confirmed to represent a statistically significant difference ($\chi^2 = 17.05$, $p < 0.001$); in both paired comparisons, all discordant records favored the Full Framework configuration ($b > 0$, $c = 0$). These alerts arise from records where the ML model identified elevated-risk configurations that did not satisfy agent escalation thresholds, and, conversely, from records where domain-specific agent rules flagged conditions the classifier had not independently identified as High-risk. Consistent with the note accompanying Table VI, the statistical tests above should be interpreted as confirming that a nonzero number of one-directional escalations occurred in each comparison, rather than as evidence of mutual, symmetric correction between the two components or of a strong discriminative advantage for either component individually. The underlying architectural rationale for combining ML and agent inputs nonetheless reflects a conceptually bidirectional design: the ML model and the agent rules are each capable of flagging risk configurations that the other does not independently classify as High-risk. A statistically significant increase in alert volume should not, on its own, be read as an operational improvement. This concern is well established in clinical alerting research: a systematic qualitative review of general practitioners' experiences with clinical decision support alerts found that a high volume of alerts with limited specificity or relevance was consistently reported as a driver of alert fatigue, reducing practitioner engagement with and trust in subsequent alerts, including clinically important ones [43]. While this concept originates in primary-care clinical alerting research rather than crowd-safety monitoring, the underlying mechanism, desensitization to frequent, low-value notifications reducing responsiveness to genuinely critical signals, is domain-general and transfers directly to the present context: an operational alerting system that increases alert volume without a corresponding increase in precision risks producing the same desensitization effect among crowd-safety monitoring personnel. As discussed below, the precision of the escalated alerts examined in this study was found to be low against the operational risk proxy; the alert-volume increase reported here should therefore be weighed against this risk rather than treated as an unambiguous benefit.

A separate check tested whether the agents' three-tier escalation logic (each agent first raising a record from Low to Medium, then separately from Medium to High) is actually necessary, or whether a single, simpler rule would produce the same alerts. The simpler version skips the intermediate Medium step entirely: it flags a record as High whenever any one of the three agents' original High-level conditions is met, with no per-agent Medium-escalation logic in between. On the same 500-record streaming evaluation, this simplified single-rule version produced exactly the same number of High-risk alerts as the full three-agent hierarchy: 93 in both cases. In other words, the extra Medium-level escalation step built into each agent did not add any High-risk alerts beyond what the three original High-level conditions already produced on their own, at least on this dataset and this evaluation window. This does not mean the five-agent design is unnecessary it remains useful for modularity and for adding more nuanced intermediate-risk logic later but it does mean that the specific benefit of the current three-tier hierarchy has not yet been demonstrated, and should not be assumed

without testing on data where Medium-risk conditions are more common or more closely tied to eventual High-risk outcomes.

TABLE VI. MCNEMAR'S TEST CONTINGENCY TABLES FOR ABLATION COMPARISONS (N=500)

Comparison	Both High	b (Comparator High, Full Not)	c (Comparator Not, Full High)	Both Not High
Full vs. ML-only	74	0	19	407
Full vs. Agents-only	23	0	70	407

Note: b and c are zero-cell-constrained by the maximum-escalation policy (Full Framework can never report fewer High-risk alerts than either individual component on a given record), which forces a one-directional discordance pattern in both comparisons. McNemar's test is designed for paired nominal comparisons and is widely recommended for comparing classifiers on a shared test set [40]; however, its standard interpretation assumes that discordant pairs can in principle occur in either direction. Because the maximum-escalation policy used here rules out one of the two discordant cells by construction, the resulting test is closer to confirming that a nonzero number of one-directional escalations occurred than to demonstrating conventional, symmetric disagreement between the compared configurations, and the reported statistics should be read with this structural constraint in mind. Continuity-corrected McNemar statistics are computed as $\chi^2 = (|b-c|-1)^2/(b+c)$.

To assess the stability of the framework with respect to threshold selection, a sensitivity analysis was conducted by varying all agent thresholds by ±20% from their baseline values. Reducing thresholds by 20% yielded 83 High-risk alerts (16.6%), whereas increasing thresholds by 20% yielded 105 High-risk alerts (21.0%). The baseline configuration produced 93 alerts (18.6%), representing an intermediate operating point within this range. This non-monotonic behavior reflects interactions among agent-specific escalation rules under the framework's maximum-escalation policy, where changes in individual thresholds can alter escalation pathways across the combined decision process. Despite these variations, the Full Framework consistently generated more High-risk alerts than the ML-only configuration (74 alerts), indicating that the reported contribution is not dependent on a single threshold setting.

The 500 streaming records were drawn exclusively from the held-out test set, ensuring complete independence from training data (this subset was drawn from the same 2,000-record test set evaluated in Section IV-A rather than from a separate sample), while preserving the original class distribution of approximately 10.1% High-risk, 31.2% Low-risk, and 58.7% Medium-risk. The increase from 74 High-risk alerts in the ML-only configuration to 93 alerts in the Full Framework is consistent with the ability of agent rules to surface risk configurations that the ML classifier may underweight due to feature overlap. Under the streaming evaluation, the average processing latency was 4.127 ms per record, indicating that the framework is suitable for single-stream operational monitoring scenarios. To assess scalability, per-record latency increased from 1.918 ms under sequential processing to 4.571 ms at 5 concurrent streams, 10.284 ms at 10 concurrent streams, and 15.538 ms at 20 concurrent streams, representing an approximately 8.1-fold increase at the highest concurrency level evaluated. These results suggest that large-scale multi-stream deployment would likely require a distributed or otherwise more scalable execution architecture.

To assess the operational precision of agent-driven escalations, the 19 records escalated by the agent layer beyond ML-only predictions (i.e., classified as Medium or Low by XGBoost+SMOTE but elevated to High by agent rules) were cross-checked against their underlying continuous risk scores and original risk labels. None of these 19 records (0%) corresponded to a true High-risk label; 14 (73.7%) were true Medium-risk and 5 (26.3%) were true Low-risk. Only 4 records (21.1%) had an underlying risk score within 0.2 of the High-risk threshold (2.4), indicating that these escalations predominantly targeted records that were not borderline cases. This finding indicates that, under the current label definition, agent-driven escalations beyond ML predictions exhibit low precision against the operational risk proxy, and should be interpreted strictly as sensitivity-oriented prioritization signals rather than validated risk detections. This result motivates recalibrating agent thresholds against verified incident data as a priority for future work, rather than treating the increased alert count alone as evidence of improved detection; such recalibration could draw on recent proposals to incorporate net benefit directly as an objective during model and threshold development, rather than applying it only as a post-hoc evaluation metric, which would allow the relative cost of missed High-risk cases versus false escalations to inform threshold selection explicitly [44].

V. DATA QUALITY, LIMITATIONS, AND FUTURE DIRECTIONS

The most fundamental limitation concerns label validity. Risk categories were derived from a weighted scoring function applied to operational indicators, not from verified incident records. The models therefore predict a computed proxy for risk rather than observed outcomes. One practical consequence is the near-zero separation between Medium- and High-risk feature distributions, resulting in zero High-risk recall for the XGBoost baseline and Cost-Sensitive XGBoost, and only partial recall (0.14) for XGBoost+SMOTE on the independent test set. In addition, agent-generated escalations could not be validated against confirmed incidents because the dataset does not include verified real-world outcomes. The additional High-risk alerts are therefore best understood as prioritization cues for further review, not as verified detections of actual incidents. Future work should evaluate these alerts against real incident logs or expert-reviewed operational records.

The weak observed performance also follows a near-deterministic statistical expectation given the dataset's construction. A post hoc correlation analysis between the 17 retained features and the continuous Operational_Risk_Score (prior to threshold discretization) found a maximum absolute Pearson correlation of 0.026 and a mean absolute correlation of 0.010 across all retained features; for n = 10,000 records, the critical correlation magnitude for statistical significance at $\alpha = 0.05$ is approximately 0.0196, meaning that while a small number of these correlations are technically distinguishable

from zero, all are practically negligible, a well-documented consequence of large sample sizes inflating the detectability of statistically significant but practically meaningless effects [45]. Because Pearson correlation captures only linear association and the models evaluated in this study (XGBoost, LightGBM, Random Forest) are explicitly capable of exploiting nonlinear and interaction effects, this analysis was complemented with a mutual information estimate, a nonparametric measure sensitive to nonlinear dependence, computed using the Kraskov–Stögbauer–Grassberger k-nearest-neighbor estimator suited to mixed continuous and discrete feature types; following the originating methodological reference, which recommends k = 2–4 for estimating dependence (with larger k trading reduced statistical variance for increased systematic bias) [46], the primary analysis used k = 3, the lower end of this recommended range, in order to favor reduced systematic bias given the relatively small number of High-risk instances available in the full dataset (1,010 of 10,000 records) for estimating this association. Mutual information between the 17 retained features and the continuous risk score ranged from 0.000 to 0.016 nats (mean 0.003 nats) at k = 3, compared with a mutual information of 0.232 nats between Crowd_Density (one of the seven excluded label-constructing variables) and the same risk score, computed as a reference point for what a genuinely informative relationship looks like in this dataset under the same measure. To verify that this conclusion does not depend on the specific neighbor-count choice, the analysis was repeated at k = 5 and k = 10: the maximum mutual information among retained features decreased further (0.008 nats at k = 5; 0.007 nats at k = 10), while the reference value for the excluded variable remained essentially unchanged (0.238 nats at k = 5; 0.241 nats at k = 10), confirming that the retained features' near-absence of nonlinear signal is stable across this range of estimator settings rather than an artifact of the k = 3 configuration. Several of the reported values are exactly 0.000; a recent systematic benchmarking of mutual information estimators, including KSG-type k-nearest-neighbor approaches, found that estimator reliability varies considerably across distributional conditions and that no single estimator performs uniformly well across all settings, particularly under sparse or weakly-informative dependence structures [47].

Consistent with this, at this sample size and neighbor parameter, the present estimator returns an exact-zero output whenever the underlying dependence falls below its practical detection floor, rather than necessarily implying a perfectly null relationship in a mathematical sense, and these values should accordingly be interpreted as "below detectable dependence at this estimator resolution" rather than as a precise quantification of zero association. The retained features therefore carry negligible nonlinear, as well as negligible linear, signal about the risk label: the highest-ranked retained feature captures roughly one-fifteenth of the mutual information carried by a single excluded label-constructing variable. This indicates that the 17 leakage-free features carry little to no signal about the risk label beyond the noise floor, under either a linear or a nonlinear association measure, consistent with the seven label-constructing variables (Crowd_Density, Temperature, Fatigue_Level, Stress_Level, Queue_Time_minutes, Emergency_Event, and Distance_Between_People_m) having been generated largely independently of the retained features in this simulated dataset. Under this condition, classification performance approaching that of the majority-class baseline is the statistically expected outcome rather than an unexplained finding, and the intrinsic-difficulty framing offered above should be read as a description of this near-independence rather than as evidence of a deeper, unmodeled relationship between observable crowd conditions and operational risk. This consideration also implies that the degree of performance degradation observed here is specific to this dataset's generation process and may not generalize to settings where label-constructing and observable variables are more strongly associated in practice.

Given this near-absence of linear and nonlinear signal in the retained features, the partial recovery of High-risk recall observed for XGBoost+SMOTE and LightGBM+SMOTE (0.14 and 0.12, respectively, versus 0.00 for the unbalanced baselines) requires its own explanation rather than being attributed to the features becoming more informative after rebalancing. The more plausible mechanism is that SMOTE alters the effective decision boundary learned by the classifier rather than increasing the information content of the feature space: by synthetically inflating the representation of the High-risk class to match the majority class during training, SMOTE shifts the model's decision threshold to favor High-risk predictions in regions of feature space that are only weakly associated with the label, at the cost of reduced precision (0.11 for XGBoost+SMOTE, Table IV) and reduced Medium-risk recall (0.67, down from 0.94 for the baseline). This interpretation is consistent with general principles of oversampling behavior in weakly separable feature spaces established in [35], [36], though the specific framing offered here is the authors' own synthesis rather than a direct claim made in either source; it favors the decision-boundary-shift account over the alternative explanation that SMOTE somehow recovers predictive signal that the correlation and mutual information analyses above show is largely absent from the underlying features. Under this interpretation, the gain in High-risk recall after SMOTE reflects a deliberate, engineered shift in operating point along an unfavorable precision-recall trade-off, not a genuine improvement in the model's ability to discriminate High-risk from Medium-risk conditions using the available observable indicators.

The dataset represents a single simulated operational context. Generalization to other crowd environments requires separate validation. Agent threshold values are expert-informed approximations not empirically calibrated against historical incident data. SMOTE generates synthetic observations from existing samples; in this study, differences between baseline and SMOTE-enhanced models remained modest, and McNemar's test confirmed significant improvement ($\chi^2 = 69.218$, $p < 0.001$). Although statistically significant, the practical effect size remained modest because the improvement was primarily observed in High-risk recall rather than overall classification accuracy. Python threading constrains true parallelism for CPU-bound workloads such as model inference, and production-scale multi-stream deployment would likely require a distributed agent framework such as JADE or Mesa. To quantify the practical impact of this constraint, per-record inference latency was measured under concurrent load using a thread pool with 1,

5, 10, and 20 simultaneous prediction streams, following a model warm-up phase to eliminate first-call overhead. As reported in Section IV-C, per-record latency increased from 1.918 ms under sequential processing to 15.538 ms at 20 concurrent streams, an approximately 8.1-fold increase at the highest concurrency level evaluated. This increase reflects the combined effect of several factors that this thread-based experiment does not separately isolate (including Python-level interpreter contention, thread-scheduling and pool-management overhead, and CPU cache contention among concurrently executing threads) rather than being attributable to the Global Interpreter Lock alone; the underlying C++ implementation of the gradient-boosting library used here may release the interpreter lock during some internal numerical computation, so the GIL constraint described above should be understood as one contributing factor among several rather than the sole explanation for the measured degradation. Regardless of the precise decomposition of these factors, the empirical result establishes that the framework's favorable latency figures reported elsewhere in this study (e.g., Table V) reflect single-stream, sequential operation and do not extend proportionally to concurrent multi-stream conditions representative of simultaneous monitoring across many pilgrims; production deployment at scale would require either a multi-process or distributed architecture, and a more detailed profiling study isolating these individual contributing factors is identified as a direction for future work.

Future work will focus on validating the framework using real-world operational datasets, calibrating agent thresholds against historical incident records, and evaluating alert precision using independently verified ground-truth events.

VI. Conclusion

This study proposed a multi-agent approach for crowd-risk prioritization in large-scale events that combines machine learning with crowd, health, and environmental agents within a unified operational workflow. The study also addressed data leakage by removing variables used to construct the risk labels, as well as post-event outcome indicators, before model training.

The results show that leakage-free prediction is difficult in this dataset. The XGBoost baseline and Cost-Sensitive XGBoost did not identify any High-risk cases (recall = 0.00). XGBoost+SMOTE detected part of this class, but performance remained limited (recall = 0.14, F1 = 0.12). McNemar's test showed a statistically significant difference between the baseline and SMOTE-enhanced model ($\chi^2 = 69.218$, $p < 0.001$), although the practical gain was mainly in minority-class sensitivity rather than overall accuracy.

The agent layer increased High-risk alert volume but, on the data examined, did so with low precision relative to the internal risk-score proxy. Of the 19 records escalated by the agent layer beyond the ML-only predictions, none (0%) corresponded to a High-risk label under this proxy, while the majority (73.7%) corresponded to Medium-risk labels. In the 500-record streaming evaluation, the Full Framework produced 93 High-risk alerts, compared with 74 from ML-only and 23 from Agents-only, representing a 25.7% increase over the ML-only configuration.

These additional alerts should therefore be interpreted as prioritization cues rather than confirmed incidents. Overall, the results suggest that ML and rule-based agents can play complementary roles in crowd-risk monitoring. However, none of the 19 agent-driven escalations beyond the ML-only predictions corresponded to a High-risk label under the internal risk-score proxy. This finding further reinforces the need for threshold calibration and validation against verified real-world incidents before operational deployment. ML provides broad probabilistic assessment across observable features, while agents add domain-specific escalation rules for conditions that the model may under-prioritize. With an average latency of 4.127 ms per record, the framework is suitable for single-stream monitoring experiments. Future work should validate the framework using real operational data and evaluate alert precision against verified ground-truth labels.

VII. Threats to Validity

A. Internal Validity

SMOTE was applied only to the training set, so synthetic samples never reached the test set; cross-validation on the SMOTE-balanced data likely makes generalization look slightly better than on the original imbalanced distribution. All models used the same fixed random seed (42) for reproducibility, and the agent thresholds are expert estimates rather than values calibrated against real incident data. As reported in Section IV- C, a simplified single-rule version of the agent logic produced the same number of High-risk alerts as the full three-agent hierarchy, showing that the added complexity of the hierarchical escalation structure has not yet been shown to add value on this dataset. Additionally, the evaluation does not quantify inter-agent communication overheads such as queue contention, synchronization latency, or message-passing delays. These overheads are expected to have minimal impact under the single-stream evaluation considered in this study but may become more relevant under higher concurrency levels.

B. External Validity

The dataset was generated under simulated conditions specific to the Hajj context, so findings may not generalize to other crowd environments (sporting events, concerts, transport hubs) without separate validation. The framework has not been tested on live IoT streams under real deployment, and Python threading limits parallelism at production scale. Furthermore, the evaluation assumes complete and reliable sensor observations and does not assess the impact of sensor noise, missing values, delayed measurements, or inconsistent data, all of which are common in practical IoT deployments and should be investigated in future work. Because the dataset was algorithmically generated rather than collected from real sensors or incidents, the statistical relationships reported here, including the near-zero feature–label correlation in Section V, the resulting recall ceiling for High-risk, and the agent-escalation precision profile in Section IV-C, reflect this synthetic generation process rather than real Hajj crowd dynamics. This is consistent with broader evaluation frameworks for synthetic tabular data, which distinguish fidelity from downstream utility and find that strong utility does not guarantee the underlying relationships match real-world data [48]. A real-world dataset could plausibly show stronger or weaker associations than

measured here, so the performance gaps, alert volumes in Table V, and precision figures in Section IV-C should be read as illustrative of this synthetic setting rather than predictive of real operational performance.

C. Construct Validity

Risk labels come from a weighted scoring function rather than verified incident records, making this a risk-prioritization tool rather than a confirmed incident predictor. Agent-alert precision against real incidents cannot be assessed without ground-truth annotations, which this dataset lacks. A check against the internal risk-score proxy (Section IV-C) found that none of the 19 agent-driven escalations beyond ML-only predictions corresponded to a true High-risk label, so High-risk alert counts should accordingly be read as proxy-based output, not confirmed real-world events.

D. Statistical Validity

Evaluation used a single dataset with one fixed train/test split. McNemar's test confirms the XGBoost+SMOTE improvement is statistically significant ($\chi^2 = 69.218$, $p < 0.001$), but a single split may not generalize across seeds or partitions. Bootstrap resampling (1,000 resamples) gave 95% confidence intervals of 47.35% [45.20%, 49.55%] for accuracy and 13.76% [9.23%, 18.60%] for High-risk recall, confirming the point estimates are stable under resampling, though this does not address variation across different training seeds. To address that directly, the full pipeline was repeated across 10 independent seeds (0, 1, 2, 3, 4, 7, 13, 21, 42, 99): accuracy averaged 47.08% (SD = 0.55%), F1 averaged 45.97% (SD = 0.46%), and High-risk recall averaged 0.134 (SD = 0.017). The seed = 42 result reported throughout this study falls within one standard deviation of this mean on every metric, confirming it is representative rather than a favorably chosen outlier. The mutual information analysis in Section V was similarly stable across k = 3, 5, and 10, confirming the near-absence of nonlinear association is not an artifact of a single parameter choice.

Declaration on the Use of AI-Assisted Tools

The author used AI-assisted tools to support language refinement and manuscript drafting during the preparation of this work. All scientific ideas, research design, experimental implementation, data analysis, results interpretation, and conclusions were carried out by the human author. The author takes full responsibility for the accuracy, originality, and integrity of the manuscript.

References

[1] T. Alafif et al., “Toward an integrated intelligent framework for crowd control and management (IICCM),” IEEE Access, vol. 13, pp. 58559-58575, 2025, doi: 10.1109/ACCESS.2025.3555154.

[2] W. Halboob, H. Altaheri, A. Derhab, and J. Almuhtadi, “Crowd management intelligence framework: Umrah use case,” IEEE Access, vol. 12, pp. 6752-6767, 2024, doi: 10.1109/ACCESS.2024.3350188.

[3] M. Khel et al., “Realtime crowd monitoring - Estimating count, speed and direction of people using hybridized YOLOv4,” IEEE Access, vol. 11, pp. 56368-56379, 2023, doi: 10.1109/ACCESS.2023.3272481.

[4] R. Nasir et al., “An enhanced framework for real-time dense crowd abnormal behavior detection using YOLOv8,” Artif. Intell. Rev., vol. 58, 2025, doi: 10.1007/s10462-025-11206-w.

[5] M. Qaraqe et al., “PublicVision: A secure smart surveillance system for crowd behavior recognition,” IEEE Access, vol. 12, pp. 26474-26491, 2024, doi: 10.1109/ACCESS.2024.3366693.

[6] S. Ramesh, B. Rohith, K. Chetanand, and R. Manikandan, “Resilient Multi-Modal IoT Crowd Monitoring System with Self-Healing Sensor Architecture and Intelligent Alert Prioritization,” in Proc. 2nd Int. Conf. Multi-Agent Syst. Collaborative Intell. (ICMSCI), Erode, India, 2026, pp. 1249-1254, doi: 10.1109/ICMSCI67830.2026.11469199.

[7] H. Zhang, R. Zhang, and J. Sun, “Developing real-time IoT-based public safety alert and emergency response systems,” Sci. Rep., vol. 15, 2025, doi: 10.1038/s41598-025-13465-7.

[8] C. Dumitrescu et al., “Crowd panic behavior simulation using multi-agent modeling,” Electronics, vol. 13, no. 18, p. 3622, 2024, doi: 10.3390/electronics13183622.

[9] M. Keykhaei et al., “A situation-aware emergency evacuation model using multi-agent-based simulation,” Geo-spatial Inf. Sci., vol. 27, pp. 1800-1823, 2023, doi: 10.1080/10095020.2023.2270017.

[10] D. Costa et al., “A survey of emergencies management systems in smart cities,” IEEE Access, 2022, doi: 10.1109/ACCESS.2022.3180033.

[11] N. Nezamoddini and A. Gholami, "A survey of adaptive multi-agent networks and their applications in smart cities," Smart Cities, 2022, doi: 10.3390/smartcities5010019.

[12] A. Kalyuzhnaya et al., “LLM agents for smart city management: Enhancing decision support through multi-agent AI systems,” Smart Cities, 2025, doi: 10.3390/smartcities8010019.

[13] C. Berceanu, I. Banu, B. Husebo, and M. Patrascu, “Predictive agent-based crowd model design using decentralized control systems,” IEEE Syst. J., vol. 17, pp. 1383-1394, 2023, doi: 10.1109/JSYST.2022.3188339.

[14] S. Veesam et al., “Multi-camera spatiotemporal deep learning framework for real-time abnormal behavior detection in dense urban environments,” Sci. Rep., vol. 15, 2025, doi: 10.1038/s41598-025-12388-7.

[15] D. Mabrouk, M. Abdel-Fattah, and A. Taha, “Robust crowd anomaly detection via hybrid ensemble learning for real-world surveillance,” Sci. Rep., vol. 15, 2025, doi: 10.1038/s41598-025-27408-9.

[16] A. Alsabei, T. Alsubait, and H. Alhakami, “Enhancing crowd safety at Hajj: Real-time detection of abnormal behavior using YOLOv9,” IEEE Access, vol. 13, pp. 37748-37761, 2025, doi: 10.1109/ACCESS.2025.3545256.

[17] Y. Wu, L. Qiu, J. Wang, and S. Feng, “The use of convolutional neural networks for abnormal behavior recognition in crowd scenes,” Inf. Process. Manage., vol. 62, p. 103880, 2025, doi: 10.1016/j.ipm.2024.103880.

[18] A. Leelamony et al., “DeepTrackNet: Advanced crowd dynamics estimation with YOLO and Deep SORT integration,” in Proc. CSASE, 2025, pp. 80-86, doi: 10.1109/CSASE63707.2025.11053996.

[19] S. Hu et al., “Integrating embedded cyber-physical systems in smart energy for AI-enhanced real-time crowd monitoring and threat detection,” IEEE Trans. Consum. Electron., vol. 71, pp. 8363-8373, 2025, doi: 10.1109/TCE.2025.3576383.

[20] F. Zeng, C. Pang, and H. Tang, “Sensors on Internet of Things systems for the sustainable development of smart cities,” Sensors, vol. 24, 2024, doi: 10.3390/s24072074.

[21] T. Wiangwiset et al., “Design and implementation of a real-time crowd monitoring system based on public Wi-Fi infrastructure,” Smart Cities, 2023, doi: 10.3390/smartcities6020048.

[22] A. Fascista, “Toward integrated large-scale environmental monitoring using WSN/UAV/crowdsensing,” Sensors, vol. 22, 2022, doi: 10.3390/s22051824.

[23] T. Narayana et al., “Advances in real time smart monitoring of environmental parameters using IoT and sensors,” Heliyon, vol. 10, 2024, doi: 10.1016/j.heliyon.2024.e28195.

[24] B. Prabhu Shankar et al., “Smart crowd: AI-driven crowd density monitoring and management in Indian public hotspots,” in Proc. ICSCDS, 2025, pp. 651-657, doi: 10.1109/ICSCDS65426.2025.11167575.

[25] S. Ayesha, A. Aslam, M. H. Zaheer, and M. B. Khan, “CIRS: A multi-agent machine learning framework for real-time accident detection and

emergency response," Sensors, vol. 25, no. 18, p. 5845, 2025, doi: 10.3390/s25185845.

[26] D. Maldonado et al., "Multi-agent systems: A survey about its components, framework and workflow," IEEE Access, vol. 12, pp. 80950-80975, 2024, doi: 10.1109/ACCESS.2024.3409051.

[27] D. Sako, C. Igiri, E. Bennett, and F. Deedam, "ESARS: A situation-aware multi-agent system for real-time emergency response management," Eur. J. Inf. Technol. Comput. Sci., 2024, doi: 10.24018/compute.2024.4.1.83.

[28] A. Takahashi, K. Yasufuku, and M. Hegazy, "Applicability of multi-agent simulation for crowd control after a large-scale event," Japan Archit. Rev., 2025, doi: 10.1002/2475-8876.70063.

[29] T. Alshammari and M. Bennis, "Logic-driven semantic communication for resilient multi-agent systems," IEEE Open J. Commun. Soc., vol. 7, pp. 630-654, 2026, doi: 10.1109/OJCOMS.2026.3653924.

[30] M. Alshamrani, S. Alhazmi, T. Alafif, T. M. Qadah, M. Aljabri, W. Al-Eidarous, and A. Alhawsawi, "From data to insights: A comprehensive analysis of pilgrims stress and fatigue during Hajj using wearable remote sensing systems," J. King Saud Univ. Comput. Inf. Sci., vol. 37, p. 94, 2025, doi: 10.1007/s44443-025-00095-2.

[31] A. Lama, M. Di Bernardo, and S. Klapp, "Nonreciprocal field theory for decision-making in multi-agent control systems," Nat. Commun., vol. 16, 2025, doi: 10.1038/s41467-025-63071-4.

[32] L. Hughes et al., "AI agents and agentic systems: A multi-expert analysis," J. Comput. Inf. Syst., vol. 65, pp. 489-517, 2025, doi: 10.1080/08874417.2025.2483832.

[33] A. Hamid, A. Monadjemi, and B. Shoushtarian, "ABDviaMSIFAT: Abnormal crowd behavior detection utilizing a multi-source information fusion technique," IEEE Access, vol. 13, pp. 74996-75015, 2025, doi: 10.1109/ACCESS.2024.3436007.

[34] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," ACM Trans. Knowl. Discov. Data, vol. 6, no. 4, pp. 1-21, 2012, doi: 10.1145/2382577.2382579.

[35] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263-1284, 2009, doi: 10.1109/TKDE.2008.239.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321-357, 2002, doi: 10.1613/jair.953.

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

[38] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD), San Francisco, CA, USA, 2016, pp. 785-794, doi: 10.1145/2939672.2939785.

[39] I. Araf, A. Idri, and I. Chairi, "Cost-sensitive learning for imbalanced medical data: A review," Artificial Intelligence Review, vol. 57, no. 4, 2024, doi: 10.1007/s10462-023-10652-8.

[40] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," Neural Comput., vol. 10, no. 7, pp. 1895-1923, 1998, doi: 10.1162/089976698300017197.

[41] M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed. Chichester, UK: Wiley, 2009.

[42] N. Bahbouh, S. Sendra, and A. Sen, "Designing a comprehensive framework for health management in crowded events," Int. J. Inf. Technol., vol. 17, pp. 1641-1651, 2024, doi: 10.1007/s41870-024-02051-1.

[43] I. Gani, I. Litchfield, D. Shukla, G. Delanerolle, N. Cockburn, and A. Pathmanathan, "Understanding "alert fatigue" in primary care: Qualitative systematic review of general practitioners' attitudes and experiences of clinical alerts, prompts, and reminders," J. Med. Internet Res., vol. 27, p. e62763, 2025, doi: 10.2196/62763.

[44] A. Vickers, A. Hollingsworth, A. Bozzo, A. Chatterjee, and S. Chatterjee, "Hypothesis: Net benefit as an objective function during development of machine learning algorithms for medical applications," Int. J. Med. Inform., vol. 197, p. 105844, 2025, doi: 10.1016/j.ijmedinf.2025.105844.

[45] M. Lin, H. C. Lucas, and G. Shmueli, "Research commentary—Too big to fail: Large samples and the p-value problem," Inf. Syst. Res., vol. 24, no. 4, pp. 906-917, 2013, doi: 10.1287/isre.2013.0480.

[46] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," Phys. Rev. E, vol. 69, no. 6, p. 066138, 2004, doi: 10.1103/PhysRevE.69.066138.

[47] P. Czyż, F. Grabowski, J. E. Vogt, N. Beerenwinkel, and A. Marx, "Beyond normal: On the evaluation of mutual information estimators," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 36, 2023, pp. 16957-16990.

[48] M. Hernandez, P. A. Osorio-Marulanda, M. Catalina, L. Loinaz, G. Epelde, and N. Aginako, "Comprehensive evaluation framework for synthetic tabular data in health: Fidelity, utility and privacy analysis of generative models with and without privacy guarantees," Front. Digit. Health, vol. 7, p. 1576290, 2025, doi: 10.3389/fdgth.2025.1576290.