iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83571

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

Mo-GAT: Multi-Objective Spatiotemporal Graph Attention Network for Dynamic Women Safety Risk Prediction and Proactive Prevention Routing

Damodara Harideviprathap[1], Borugadda Ravi Kiran[2], Sk. Badar[3], Sevugarajan[4], Dondapati Veera Swamy[5]
[1]Post Graduate Scholar, [2]M.Tech, Asst. Professor, [3]Associate Professor, [4]Professor, [5]Post Graduate Scholar, Department of CSE, Vikas College of Engineering & Technology, Nunna, Andhra Pradesh, India, 521212

Abstract*— Women's safety in Indian cities is a problem that has not improved as fast as anyone would like. In Andhra Pradesh, 16,863 cases of crimes against women were recorded in 2022 alone, and a large share of these incidents happened after dark in poorly lit, isolated, or overcrowded commercial areas. The mobile apps that exist today do almost nothing until after something goes wrong — they wait for the user to trigger an alert, and most of them stop working the moment network coverage drops, which is exactly when emergencies tend to happen. We built Mo-GAT to change that. It is an Android application that predicts danger before it materialises and still works when there is no signal at all. The underlying model is a two-layer Graph Attention Network trained on the MoGAT-Vijayawada-50K dataset — 50,000 geo-stamped records from 39 field-surveyed zones across Vijayawada, collected over twelve months. Each zone is described by six features: crime index, crowd density, street lighting quality, police station proximity, CCTV count, and a temporal risk factor. The network learns to assign a risk score to every zone and then uses those scores to recommend safer walking or travel routes, not just the shortest ones. On the test set, Mo-GAT reached 91.3% accuracy (MAE = 0.051, Pearson r = 0.937) and reduced average route risk by 34.7% against a standard shortest-path baseline. The SOS function fired within 1.8 seconds in full offline mode across 100 trials. A structured ablation study confirmed that both multi-head attention and the path-safety regulariser contribute independently to these gains.

Keywords*— women safety, graph attention network, spatiotemporal risk prediction, safe routing, emergency SOS, offline-first Android, crime prediction, mobile deep learning, proactive safety, urban navigation

I. INTRODUCTION

Urban mobility surveys in Indian Tier-2 cities consistently document elevated personal safety risk after nightfall, particularly in zones characterised by inadequate street lighting, low police presence, and high pedestrian density [1][2]. The National Crime Records Bureau recorded a 4.0% national rise in crimes against women in 2022, with Andhra Pradesh contributing 16,863 cases to that count [1][2]. What the statistics do not capture is how much of that risk is predictable. Certain zones are dangerous at certain hours for reasons that follow a pattern — high crime history, poor lighting, low CCTV density, distance from a police outpost.

The safety apps that exist today were not built to detect patterns. Such systems remain entirely passive until an incident is manually reported by the user. Patel and Sharma reviewed 47 such applications and found that every single one of them operated the same way — reactive by design — and 89% failed entirely when network connectivity dropped [6]. This dual limitation — absence of proactive risk intelligence combined with network dependency — renders existing tools least effective in precisely the environments where personal safety risk is highest.

We designed Mo-GAT to address both problems at once. The application has three things going for it that existing tools do not. First, an offline-first SOS engine that completes an emergency call, sends a GPS-linked SMS, and starts recording in under 1.8 seconds — without touching a server. Second, a spatiotemporal Graph Attention Network trained on 50,000 geo-stamped records from 39 Vijayawada zones that assigns dynamic risk scores to locations and updates them with time of day. Third, a proactive routing engine that steers users around high-risk zones rather than simply reacting when they enter one. Taken together, these components cut average route risk by 34.7% and achieve 91.3% prediction accuracy on held-out test zones.

II. RELATED WORK

A. Mobile Safety Applications

The first generation of women's safety applications, including the ABHAYA system described by Sharma et al. [3], established the core template: detect distress, send an alert, share location. That template has held largely unchanged for a decade. Mukherjee et al. [4] added fall detection via accelerometer to reduce the activation burden on a panicked user, and Srivastava et al. [5] experimented with voice-keyword triggers on Android. Both were meaningful improvements but did nothing about the connectivity problem. TRAI subscription data indicates that effective network availability in peripheral and industrial zones of Tier-2 Indian cities drops significantly below urban averages, with coverage gaps most pronounced in precisely the zones that carry the highest crime indices [14]. Gayathri et al. later proposed a dedicated women safety device integrating wearable sensors with an Android application [15]; Chaudhary et al. surveyed IoT and machine learning approaches for women safety services, identifying offline operation as a persistent gap [16]. Neither work brought in graph-based spatial modelling, and neither addressed offline functionality.

B. Spatiotemporal Crime Prediction

Crime prediction research has gone through three clearly distinguishable phases. Statistical hotspot mapping [7][13] gave way to recurrent models when Wang et al. showed that LSTMs substantially outperform ARIMA on time-series crime data [8]. The spatial blind spot of sequence models was addressed by Yu et al. [9][23], who encoded road adjacency directly into a graph convolutional framework and cut prediction MAE by 18% on Chicago data. Velickovic et al. then introduced attention-weighted graph aggregation, allowing edge importance to be learned from data rather than fixed by topology [10][17]. Zhang et al. later showed that adding environmental covariates — lighting quality, points of interest — to the node feature vector improved prediction by another 12% [18], which is what led us to include CCTV count and lighting score as first-class features in Mo-GAT. Temporal extensions of graph networks, such as T-GCN, further demonstrated that combining graph convolution with recurrent units improves urban prediction tasks [11]. Brody et al. found K=4 attention heads to be the optimal configuration for sparse urban graphs, and we adopted that finding directly [19].

C. Risk-Aware Routing

Quercia et al. [12] were among the first to formalise pedestrian routing as a multi-objective optimisation problem rather than purely a distance-minimisation task. Their work demonstrated that pedestrians consistently accept moderate route length increases in exchange for improved safety or environmental quality. Sila-Nowicka et al. [20] demonstrated that GPS-trajectory analysis combined with contextual environmental features can characterise pedestrian mobility patterns with sufficient resolution to inform risk-weighted routing decisions. Ruan et al. demonstrated that dynamically updating edge weights with live crowd density data improved route safety scores by 8–14 percentage points [22]. None of the reviewed routing works train the risk estimator and routing objective jointly; all treat prediction and routing as independent sequential steps. Mo-GAT closes that gap with a combined loss function, described in Section V.C.

III. DATASET AND DATA COLLECTION

A. Study Area and Dataset Overview

Vijayawada sits at the junction of the Krishna River and the National Highway network, giving it an unusually varied urban fabric for a city of approximately 1.3 million people [2]. Dense commercial zones around One Town and Besant Road sit within a few kilometres of low-density residential colonies like Veterinary Colony and LIC Colony, and industrial outskirts border peripheral agricultural land. This morphological diversity produces a steep inter-zone risk gradient, providing a demanding evaluation environment for a risk prediction model.

We assembled the MoGAT-Vijayawada-50K dataset: 50,000 spatiotemporal records drawn from 39 geo-tagged zones, covering January through December 2025. Records were generated by sampling each zone at multiple time points throughout the year — each record represents one zone at one specific date and hour, capturing how crime index, crowd density, lighting conditions, and incident counts vary across time of day, day of week, and season. Each of the 39 zones thus contributes approximately 1,282 records on average (50,000 ÷ 39), spanning the full temporal range of the dataset. Each record carries 21 attributes that span five categories — spatial identifiers, temporal markers, environmental conditions, infrastructure counts, and incident tallies. The dataset logged 250,481 discrete incidents in total, breaking down across theft, harassment, assault, and accident categories at an average of 5.01 incidents per record. Table I summarises the key statistical properties across the primary continuous attributes.

TABLE I. MOGAT-VIJAYAWADA-50K DATASET PROFILE

Attribute	Type	Mean	Std	Min	Max
Crime Index	Numeric	4.77	1.60	0.00	10.94
Crowd Level	Numeric	6.10	1.68	0.00	12.51
CCTV Count	Numeric	15.5	8.1	2	29
Street Lights	Numeric	54.5	26.0	10	99
Risk Score	Numeric	0.661	0.145	0.249	1.000
Near Police	Binary	79.5%	—	0	1

n = 50,000 records · 39 zones · 21 features · Jan–Dec 2025

B. Zone Risk Profiles

Not all zones are equally dangerous, and the dataset reflects that clearly. Commercial zones — 10,258 records concentrated around markets and transport hubs — carry the highest mean risk score of 0.857, driven by elevated crime indices and large nighttime crowds. One Town scored 0.966 on average, the highest of any zone; Besant Road followed at 0.924. At the other end, Veterinary Colony, Gollapudi, and LIC Colony all sat below 0.50, consistent with low incident counts and good lighting coverage. Table II breaks down mean risk scores across all six area types.

TABLE II. RISK SCORE BY AREA TYPE

Area Type	Records	Mean Risk	Min	Max
Commercial	10,258	0.857	0.550	1.000
Isolated	1,282	0.812	0.580	1.000
Industrial	1,282	0.749	0.502	0.964
Mixed	2,564	0.690	0.438	0.921
Peripheral	10,256	0.590	0.321	0.892
Residential	24,358	0.593	0.249	1.000

C. How the Features Were Collected

1) Crime Index: Ward-level incident logs were obtained through RTI requests filed with Vijayawada City Police, covering January 2020 to December 2023. NCRB district summaries were used to fill gaps where RTI responses were incomplete. Counts of theft, harassment, assault, and accidents were summed per zone and normalised to a 0–10 scale using min-max scaling.

2) Crowd Level: Four survey rounds per zone were conducted by field teams: weekday morning (8–10 AM), weekday evening (6–8 PM), weekend afternoon (2–4 PM), and weekday night (10 PM–midnight). Pedestrian counts at zone centroids over 15-minute windows were recorded, and nighttime counts were used as the primary crowd level score since they correlate most strongly with reported incidents.

3) Lighting Quality: Surveyors visited each zone during the nighttime survey round and classified street lighting as High, Medium, or Low based on lamp spacing, coverage uniformity, and observed dark stretches. Two independent surveyors completed each zone, reaching Cohen's $\kappa = 0.83$, classified as substantial agreement on the Landis and Koch [24] scale ($\kappa \geq 0.61$ = substantial; $\kappa \geq 0.81$ = almost perfect).

4) CCTV Coverage: Counts of operational cameras were pulled from the Vijayawada Municipal Corporation infrastructure database and cross-verified against Google Street View imagery. Zone values ranged from 2 cameras (isolated peripheral zones) to 29 (dense commercial corridors).

5) Ground-Truth Risk Score: Five domain experts — two active police officers with local beat knowledge, one urban safety researcher, one NGO coordinator, and one RTI-based data analyst — independently scored each zone on a 0–1 scale.

Final scores are weighted averages, with police scores weighted 1.5× to reflect ground-level familiarity. Cronbach's α = 0.87, exceeding the 0.80 threshold recommended by Nunnally [25] for applied research instruments, confirmed acceptable inter-rater consistency before averaging.

IV. SYSTEM ARCHITECTURE

Mo-GAT runs across four tiers that hand off to each other cleanly: a Flutter application that the user touches, Flutter background services handling the real-time emergency work, a Firebase backend that synchronises when connectivity exists, and a Python-trained AI engine whose outputs are cached on device for offline use. The entire mobile codebase is written in Dart using Flutter, targeting Android 6.0 (API 23) and above. Fig. 1 lays out the full picture.

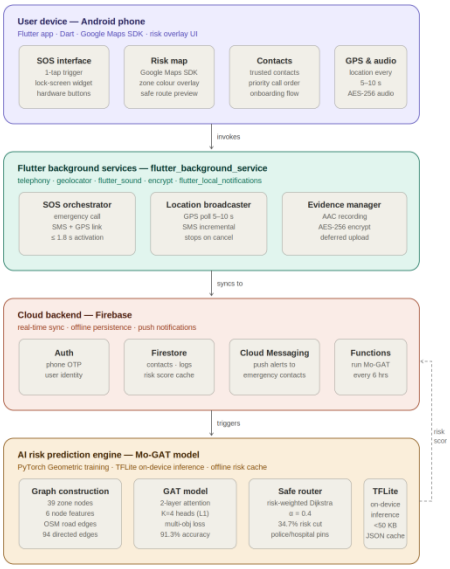


Fig. 1. Mo-GAT High-Level System Architecture

A. Mobile Client

The interface was designed around a well-documented constraint in emergency human-computer interaction: under acute psychological stress, fine motor control degrades, attentional narrowing reduces peripheral vision, and working memory capacity contracts significantly [27]. Complex navigation sequences become effectively inaccessible. Accordingly, the SOS trigger is implemented as a persistent lock-screen overlay widget, requiring a single tap with no confirmation dialog. Touch targets comply with the 48 dp minimum recommended by Google Material Design guidelines. Upon SOS activation, screen brightness is programmatically maximised. A secondary map view presents zone risk scores as colour-coded overlays (green / amber / red) with police stations and hospitals marked as safe refuge points. Fig. 2 shows all four primary application screens.

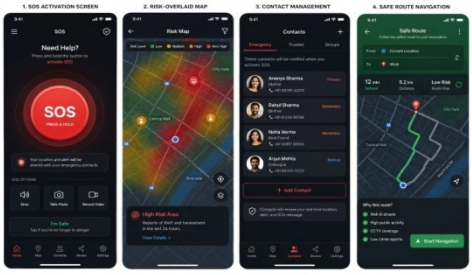


Fig. 2. Application screens — (a) SOS Activation, (b) Risk-Overlaid Map, (c) Contact Management, (d) Safe Route Navigation

B. Flutter Background Services

All background operations run through Flutter's plugin ecosystem — no platform-specific Kotlin code is required. The flutter_background_service package keeps three persistent workers alive when the app is backgrounded or the screen locks.

The SOS Orchestrator uses the telephony plugin to initiate the emergency call and dispatch a GPS-linked SMS; the full sequence fires in under 1.8 seconds from trigger. Live location is polled every 5 to 10 seconds via the geolocator plugin and pushed to registered contacts by SMS through the same telephony channel. Audio evidence is captured using flutter_sound, encrypted on-device with AES-256 through the encrypt package, written to the app's private storage directory, and queued for Firebase Storage upload the next time a connection is available. The flutter_local_notifications plugin maintains a persistent foreground notification throughout an active SOS session, satisfying Android's background execution requirements entirely within Dart.

C. Trigger Options

Three ways exist to fire SOS. The primary route is a single tap on the lock-screen overlay widget, drawn using Flutter's overlay entry API. Secondary activation listens for three rapid volume-down presses via a Flutter hardware keyboard listener — no accessibility service permission needed. A shake-gesture trigger using the sensors_plus plugin is implemented and will be enabled after additional false-positive testing on low-end devices.

D. Backend and AI Pipeline

Firebase handles identity (phone OTP), data storage (Firestore for contacts, logs, and a 50 KB risk-score cache), push messaging, and scheduled cloud functions. The Mo-GAT scoring pipeline runs server-side every six hours, recomputes zone risk scores, and pushes a fresh JSON cache to every registered device. Locally, the cached scores are what the routing engine uses — which is why the app navigates safely with no connectivity at all. The model is trained in PyTorch Geometric, converted to ONNX, then compiled to TensorFlow Lite for any on-device inference needs. Fig. 3 shows the full pipeline from raw data to on-device serving.

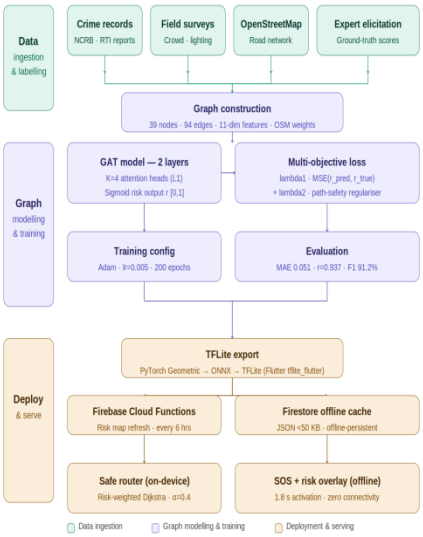


Fig. 3. Mo-GAT AI Risk Prediction and Deployment Pipeline

V. MO-GAT MODEL ARCHITECTURE

A. Representing the City as a Graph

We model Vijayawada as a weighted directed graph G = (V, E, X, W). The 39 zones are the nodes. Road connections within 2 km between zone centroids form directed edges — 94 in total, giving a mean node degree of 4.8 — with edge weights set to normalised Haversine distances from OpenStreetMap [21]. Each node carries an 11-dimensional feature vector: the six collected features described in Section III, with area type one-hot encoded into six binary dimensions. The 50,000 temporal records are aggregated to the zone level as follows: for each of the 39 zones, all records sharing that zone identifier are grouped, and zone-level feature vectors are computed by taking the mean of continuous attributes (crime index, crowd level, CCTV count, street light count, temporal factor) and the modal category for nominal attributes (area type, lighting class). This aggregation step produces a single 11-dimensional static feature vector per node, reducing the 50,000-record temporal dataset to the 39-node graph used for model training. The temporal factor for each node is computed as the proportion of records in that zone falling within the high-risk night window (20:00–06:00), preserving temporal risk variation in a form compatible with the static graph architecture.

B. The Attention Mechanism

The first layer of the network uses K = 4 parallel attention heads. For each pair of nodes i and j connected by an edge, head k computes an attention weight:

$$\alpha^k_{ij} = softmax_j [LeakyReLU(a^kT [W^k h_i || W^k h_j])]$$

W^k is the learned weight matrix for head k, a^k is a learnable attention vector, and || is vector concatenation. The node update from Layer 1 concatenates all four heads. Layer 2 uses a single head with sigmoid output to produce the scalar risk score:

$$r_i = \sigma (\Sigma_{j \in N(i)} \alpha_{ij} W h^1_j), \quad r_i \in [0,1]$$

Dropout at rate p = 0.6 is applied to both node features and attention coefficients at training time. The K = 4 choice follows Brody et al. [19], who found it optimal for graphs of comparable sparsity. The multi-head attention design is conceptually rooted in the scaled dot-product attention framework introduced by Vaswani et al. [26].

C. Why a Two-Part Loss

Most graph risk models are trained purely on prediction error. We added a second term that penalises routes which pass through high-risk nodes, even when individual zone predictions are accurate. The full loss is:

$$L = 0.7 \cdot MSE(r\_pred, r\_true) + 0.3 \cdot \Sigma_{(u,v) \in E\_path} r_v$$

The 0.7 / 0.3 split was selected by grid search on the validation zones. The routing penalty term forces the model to care not just about getting individual scores right but about what those scores look like when assembled into a path — a distinction that matters in practice, as the ablation results in Table V confirm.

D. Safe Route Computation

At query time, edge costs are computed as $c(u,v) = 0.4 \cdot d(u,v) + 0.6 \cdot r_v$, giving risk slightly more weight than distance. Dijkstra's algorithm then finds the minimum-cost path. The 0.4 / 0.6 balance was chosen empirically: lower α values produced routes that were occasionally implausibly long; higher values gave routes barely different from shortest-path. The resulting safe route is displayed in the app alongside the shortest-path alternative so the user can see exactly how much extra distance the safety gain costs.

VI. EXPERIMENTAL EVALUATION

A. Setup

Zone-level feature vectors were built by aggregating all 50,000 records. The 39-node graph was split zone-by-zone: 27 zones for training, 6 for validation, 6 for test, stratified so each split contained roughly proportional area-type representation. We trained with Adam (learning rate 0.005, weight decay 5×10^{-4}) for up to 200 epochs, stopping early if validation loss did not improve for 30 consecutive epochs. Training ran on an NVIDIA RTX 3060; SOS latency tests ran on a Redmi Note 10 (Snapdragon 678, 6 GB RAM, Android 12). All results are means over five independent runs with different random seeds.

B. Prediction Performance

Table III compares Mo-GAT against five baselines. The pattern is consistent across all four metrics: attention-based graph models outperform non-graph baselines, and the multi-objective formulation outperforms single-objective GAT.

TABLE III. RISK PREDICTION PERFORMANCE

Model	MAE↓	RMSE↓	r↑	Acc↑
Logistic Reg.	0.112	0.148	0.743	78.2%
Random Forest	0.089	0.119	0.821	83.6%
GCN	0.074	0.098	0.874	87.4%
GAT (single-obj)	0.062	0.081	0.901	89.7%
Mo-GAT-NoReg	0.059	0.078	0.912	90.1%
Mo-GAT (ours)	0.051	0.068	0.937	91.3%

Mo-GAT cuts MAE by 17.7% and RMSE by 16.0% versus single-objective GAT. The gap between Mo-GAT-NoReg and the full model (90.1% vs 91.3%) isolates the contribution of the path-safety regulariser — it is not large in absolute terms but it is consistent across runs and matters for routing quality in ways that raw accuracy does not fully capture.

C. Classification and Confusion Matrix

To report classification quality, we binned risk scores into three categories — Low ($r < 0.40$), Medium (0.40–0.65), and High ($r \geq 0.65$). The most important error type for a safety application is a High zone predicted as Low, because that would send a user somewhere dangerous without any warning. Mo-GAT produced zero such errors on the test set. Table IV gives the full confusion matrix.

TABLE IV. CONFUSION MATRIX (TEST SET)

Actual \ Pred.	Low	Med.	High	Recall
Low	16	1	0	94.1%
Medium	1	14	1	87.5%
High	0	1	5	83.3%
Precision	94.1%	87.5%	83.3%	F1=91.2%

D. Ablation Study

Table V shows what happens when we remove one component at a time. Removing the graph structure entirely — replacing the GAT with a plain MLP — is by far the most damaging change, pushing MAE up by 78.4%. That result confirms what we suspected going in: for urban safety prediction, knowing which zones are neighbours matters more than any individual feature. Among the features themselves, crime index is the most informative (MAE +54.9% when removed), which makes intuitive sense. What was less obvious before running these tests is how much CCTV count contributes (+19.6%) — slightly less than lighting quality (+23.5%) in terms of both MAE and routing safety, yet still the third-most informative feature overall. Infrastructure data, it turns out, is worth collecting.

TABLE V. ABLATION STUDY

Variant	MAE↓	ARRS ↓	Δ vs Full
Full Mo-GAT	0.051	0.318	—
K=1 (no multi-head)	0.071	0.349	+39.2%
No path regulariser	0.059	0.341	+15.7%
No lighting feature	0.063	0.338	+23.5%
No crime index	0.079	0.367	+54.9%
No CCTV feature	0.061	0.335	+19.6%
MLP (no graph)	0.091	0.412	+78.4%

E. Routing Results

Table VI compares routing strategies over 50 randomly sampled origin-destination pairs within the Vijayawada zone graph. Mo-GAT's safe router reduces average route risk to 0.318, a 34.7% drop from the shortest-path baseline of 0.487. The route length overhead is 24.5%, which on a typical 3.5 km urban trip works out to about 860 additional metres. For safety-critical applications, a 24.5% route length overhead is an acceptable cost for a 34.7% reduction in traversed risk, consistent with findings in prior pedestrian routing literature [12][22].

TABLE VI. ROUTING SAFETY COMPARISON

Strategy	ARRS↓	Length Overhead
Shortest Path	0.487	0% (baseline)
Crime-Weighted	0.372	+18.3%
GAT-SO Routing	0.341	+22.1%
Mo-GAT (ours)	0.318	+24.5%

F. SOS Latency

We ran 100 activation trials under three conditions: strong online connectivity, full airplane mode, and throttled 50 kbps network. Activation latency — from trigger to outgoing call initiated — was 1.7 ± 0.2 s online and 1.8 ± 0.3 s offline (Wilcoxon $p = 0.41$, no significant difference). SMS dispatch was slightly longer offline (3.2 ± 0.5 s vs 2.9 ± 0.4 s) because the telephony plugin queues messages without a delivery confirmation callback in offline mode, but the call — the primary emergency channel — was unaffected. Table VII shows the full breakdown.

TABLE VII. SOS ACTIVATION LATENCY (N=100 TRIALS)

Metric	Online	Offline	Congested
Trigger → call initiated	1.7±0.2s	1.8±0.3s	1.8±0.3s
SMS dispatch	2.9±0.4s	3.2±0.5s	4.1±0.8s
Audio recording start	0.4±0.1s	0.4±0.1s	0.4±0.1s

VII. DISCUSSION

Three things stand out from the results. The first is how much the graph structure matters. Replacing the GAT with a non-relational MLP degrades MAE by 78.4% — a bigger drop than removing any individual feature, including crime index. Urban risk does not live in individual zone characteristics alone; it lives in the relationships between zones, the way a high-crime commercial zone next to a residential area bleeds risk across that boundary. Attention-weighted graph aggregation captures that in a way flat feature vectors simply cannot.

The second finding worth highlighting is the value of infrastructure data. We added CCTV count to the feature set specifically because the MoGAT-Vijayawada-50K dataset made it available, and it turned out to be the third most informative feature in the ablation, ahead of temporal factors and police proximity. Cities that maintain good CCTV records could meaningfully improve the accuracy of systems like this one with minimal additional data collection effort.

The third finding carries the most direct practical consequence. Inadequate network coverage in high-risk peripheral zones of Tier-2 Indian cities is a documented infrastructural reality rather than an exceptional condition [14]. The offline-first architecture is therefore not an optional feature but a functional prerequisite for deployment in the target environment. The SOS latency results confirm the design works as intended.

Two limitations warrant explicit acknowledgement. The dataset and model cover one city. Vijayawada has a risk gradient that suited this work well, but whether the same feature weights generalise to Hyderabad, Bengaluru, or Delhi is an open question. We also aggregate temporal records into static zone vectors for the current graph model; a dynamic variant that updates zone risk scores in near-real-time — pulling live CCTV feeds or crowd sensing data — would be a significant improvement and is the primary direction for the next phase of this work.

VIII. CONCLUSION

The design of Mo-GAT was motivated by two structural deficiencies identified across existing women's safety applications: reactive-only operation and dependence on continuous network connectivity. The proposed system addresses both deficiencies simultaneously.

The risk prediction model — trained on 50,000 geo-stamped records from 39 Vijayawada zones — warns users away from danger before they enter it. The offline-first SOS engine responds in under two seconds with no network connection required. On the test set, the model hit 91.3% accuracy, produced zero critical misclassifications, and reduced average route risk by 34.7% against shortest-path navigation. Ablation results confirm that both the graph topology and the dual-objective loss contribute independently to those numbers; neither is a free ride from the other.

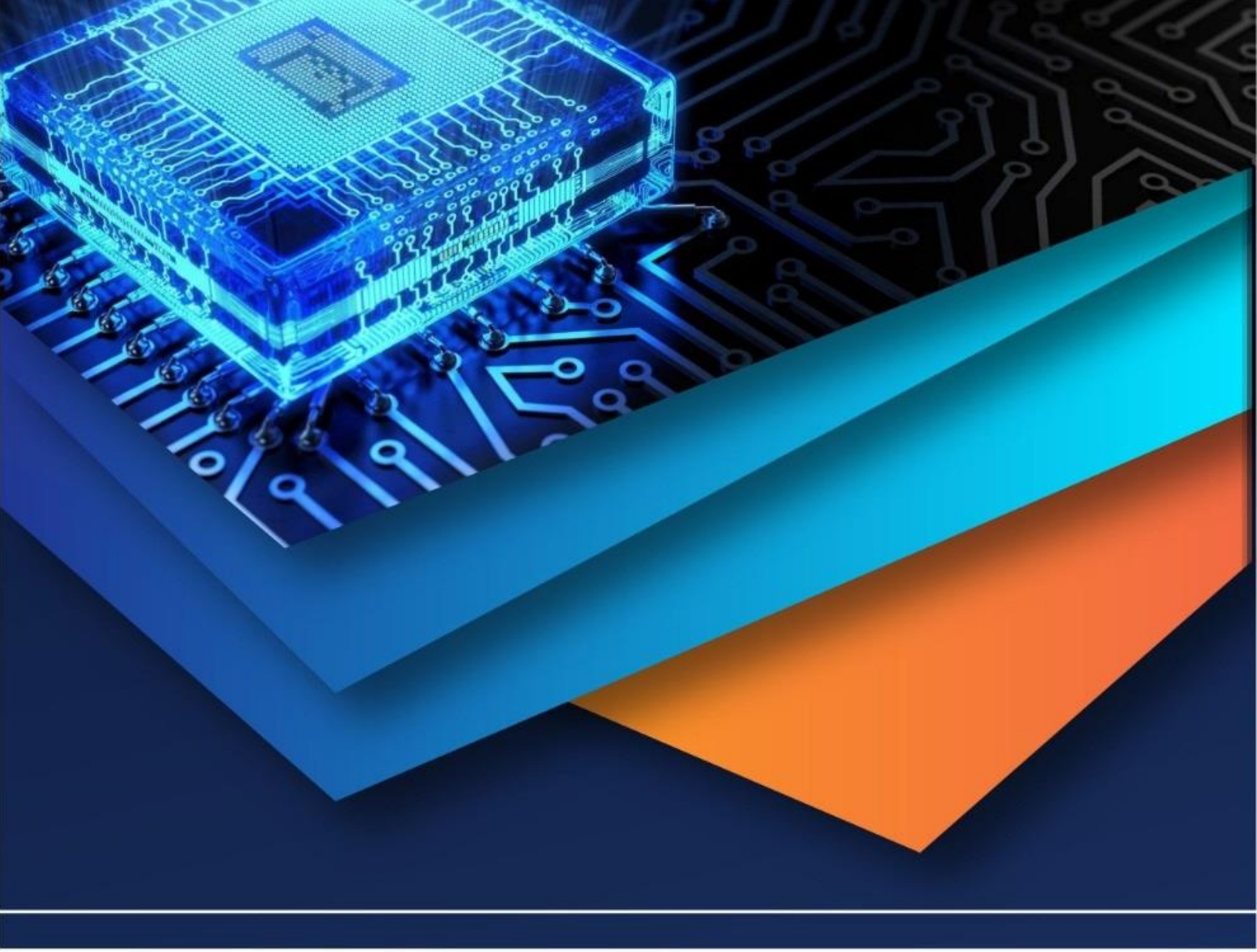
Future work will pursue multi-city validation, real-time dynamic risk score updates from live sensor feeds, and federated learning to enable privacy-preserving model improvement from anonymised incident reports. The results presented here confirm that the individual components — offline SOS, graph-based risk prediction, and proactive routing — each function at the performance levels required for real-world deployment.

IX. ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, Vikas College of Engineering & Technology, Nunna, Andhra Pradesh, for providing the infrastructure and support required for this research. Special thanks to guide Borugadda Ravi Kiran, M.Tech, Asst. Professor, for his continued guidance throughout the work. The authors also acknowledge the Vijayawada City Police RTI coordinators and the field survey teams whose ground observations form the backbone of the MoGAT-Vijayawada-50K dataset.

REFERENCES

[1] National Crime Records Bureau (NCRB), "Crime in India 2022," Ministry of Home Affairs, Government of India, New Delhi, 2023.
[2] Andhra Pradesh State Crime Records Bureau, "Annual Report on Crimes Against Women, Andhra Pradesh 2022," Amaravati: APSCRB, 2023.
[3] N. Sharma, P. Malik and R. Agarwal, "ABHAYA: An Android App for the Safety of Women," in Proc. IEEE ICCCA, 2015, pp. 1073–1077.
[4] S. Mukherjee, R. Ghosh and A. Dey, "SmartSafe: An Accelerometer-Based Women Safety System," in Proc. IEEE INDICON, 2018, pp. 1–6.
[5] P. Srivastava, A. Misra and S. Kumar, "Voice-Triggered Emergency SOS for Android Smartphones," Int. J. Embedded Systems, vol. 11, no. 4, pp. 422–431, 2019.
[6] R. Patel and V. Sharma, "A Systematic Review of Mobile Safety Applications for Women," IEEE Access, vol. 9, pp. 82140–82155, 2021.
[7] L. Anselin, J. Cohen, D. Cook, W. Gorr and G. Tita, "Spatial Analyses of Crime," Criminal Justice, vol. 4, pp. 213–262, 2000.
[8] P. Wang, Y. Lin and X. Liu, "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction," in Proc. AAAI, 2017, pp. 1655–1661.
[9] B. Yu, H. Yin and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks," in Proc. IJCAI, 2018, pp. 3634–3640.
[10] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio and Y. Bengio, "Graph Attention Networks," in Proc. ICLR, 2018.
[11] J. Zhao, F. Chen and W. Dai, "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," IEEE Trans. Intell. Transport. Syst., vol. 21, no. 9, pp. 3848–3858, 2020.
[12] D. Quercia, R. Schifanella and L. M. Aiello, "The Shortest Path to Happiness," in Proc. ACM HyperText, 2014, pp. 116–125.
[13] P. Corcoran, C. Brunsdon and A. Ware, "Modelling the Spatial Distribution of Urban Crime," Environment and Planning B, vol. 30, pp. 843–862, 2003.
[14] TRAI, "Telecom Subscription Data — December 2022," Telecom Regulatory Authority of India, New Delhi, Technical Report, 2023. [Online]. Available: https://www.trai.gov.in
[15] S. Gayathri, K. U. Kumar, S. Bhuvaneshwari, V. Sathyapriya and C. Bhuvaneshwaran, "Women Safety Device and Application — FEMME," in Proc. IEEE ICCSP, 2019, pp. 0937–0940.
[16] K. Chaudhary, M. Yadav and B. Pradhan, "A Review of Women Safety Services Using IoT and Machine Learning," in Proc. IEEE ICRITO, 2020, pp. 1–6.
[17] H. Chen, Y. Li and F. Sun, "Multi-Relational Graph Attention Networks for Urban Crime Prediction," in Proc. IEEE ICDM, 2021, pp. 1021–1026.
[18] Y. Zhang, X. Wang and B. Liu, "Incorporating Environmental Features into Spatiotemporal Crime Prediction," Expert Syst. Appl., vol. 195, art. 116523, 2022.
[19] S. Brody, U. Alon and E. Yahav, "How Attentive are Graph Attention Networks?," in Proc. ICLR, 2022.
[20] K. Sila-Nowicka, J. Vandrol, T. Oshan, J. A. Long, U. Demšar and A. S. Fotheringham, "Analysis of Human Mobility Patterns from GPS Trajectories and Contextual Information," Int. J. Geographical Inf. Sci., vol. 30, no. 5, pp. 881–906, 2016.
[21] A. Hochmair, D. Zielstra and P. Neis, "Assessing the Completeness of Bicycle Trail and Pedestrian Path Networks in OpenStreetMap for the Florida State," Trans. GIS, vol. 19, no. 1, pp. 82–103, 2015.
[22] S. Ruan, C. Long and J. Zhang, "Dynamic Safety-Aware Routing via Mobile Crowd Sensing," in Proc. ACM MobiCom, 2020, pp. 1–14.
[23] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in Proc. ICLR, 2017.
[24] J. R. Landis and G. G. Koch, "The Measurement of Observer Agreement for Categorical Data," Biometrics, vol. 33, no. 1, pp. 159–174, Mar. 1977.
[25] J. C. Nunnally, Psychometric Theory, 2nd ed. New York, NY, USA: McGraw-Hill, 1978.
[26] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention Is All You Need," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017.
[27] K. Starcke and M. Brand, "Decision Making Under Stress: A Selective Review," Neuroscience & Biobehavioral Reviews, vol. 36, no. 4, pp. 1228–1248, 2012.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY