Fusion of Opportunistic Networks with Machine Learning: Present and Future

Sonal Beniwal[1], Puneet Garg[2], Rakesh Rajpal[3], Neetu Sharma[4], Harish Kumar Mittal[5]

[1]Associate Professor, BPSMV, Khanpur Kalan, Gohana, Sonipat, Haryana, India
[2]Associate Professor, KIET Group of Institutions, Delhi NCR, Ghaziabad, India
[3]Professor, SAITM, Gurugram Delhi NCR, Haryana, India
[4]Professor, Galgotias University Greater Noida, Uttar Pradesh, India
[5]Professor, BMIET, Sonepat, Delhi NCR, India
sonalkharb@gmail.com[1], puneetgarg.er@gmail.com[2], rajpal.rakesh@gmail.com[3], neetush75@gmail.com[4], mittalberi@gmail.com[5]

Abstract: Opportunistic Networks (OppNets) are mobile ad hoc networks characterized by intermittent connectivity and the lack of a guaranteed end-to-end path between source and destination. Nodes in an OppNet employ a store-carry-forward strategy – messages are stored and carried by mobile nodes until a communication opportunity arises, at which point they are forwarded. This paradigm enables data delivery in challenging environments (disaster areas, remote regions, etc.) where conventional infrastructure is absent, but it also introduces high delays and uncertainty. Machine Learning (ML) has emerged as a powerful tool to improve OppNet performance by exploiting patterns in node mobility, contact frequency, and context. This paper provides an extensive survey of the state-of-the-art in merging ML with OppNets and discusses future developments. In this paper, we analyze how ML algorithms have enhanced message delivery rates, reduced delays, and improved decision-making in OppNets (often outperforming traditional protocols by significant margins), as illustrated by recent results in the literature. Key challenges at this fusion include data sparsity, computational constraints on mobile devices, privacy/security concerns, and the need for realistic testing.

Keywords: Machine Learning, Opportunistic Networks, OppNet

1. Introduction

Opportunistic Networks (OppNets) are a class of delay-tolerant wireless networks in which mobile nodes communicate *opportunistically* – i.e. only when they happen to come into contact – rather than over stable end-to-end links. In an OppNet, a contemporaneous multi-hop path rarely exists between a given source and destination. Instead, data are transferred through a series of sporadic contacts, as nodes *store* messages, *carry* them while moving, and *forward* them upon encountering new nodes [23][27].

This store-carry-forward mechanism, illustrated conceptually in Figure 1, allows OppNets to deliver messages despite frequent disconnections and topology changes. Common examples of OppNets include mobile social networks formed by pedestrians or vehicles, Delay-Tolerant Networks for wildlife tracking or rural communications, and sensor data collection networks using mobile sinks. In all such cases, devices must tolerate delays and uncertainty; for instance, a sensor reading might be relayed through multiple opportunistic hops over hours or days to reach a data center [20][21].

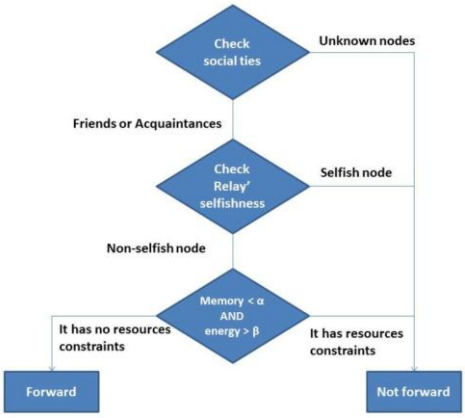

Figure 1: Example decision flow in an ML-enhanced OppNet routing strategy. A node forwards a message only if certain learned conditions are met (e.g. the next hop is a socially connected, non-selfish node with sufficient resources); otherwise it waits for a better opportunity. Such intelligent forwarding decisions, often obtained via machine learning models, can greatly improve delivery outcomes in intermittently connected networks.

Despite their fragmented connectivity, OppNets are significant because they enable communication in scenarios where conventional networks fail. By leveraging node mobility and peer-to-peer contact opportunitiesRouting in such environments is non-trivial – classical protocols (designed for connected networks) perform poorly, so specialized delay-tolerant routing protocols have been developed (e.g. Epidemic routing, PRoPHET, Spray-and-Wait). These protocols use probabilistic or copy-based strategies to maximize delivery, but they may flood the network with replicas or rely on simplistic heuristics [10-16] .

2. Background

2.1 Opportunistic Networks.

In an OppNet, nodes communicate via direct wireless contacts in a peer-to-peer, intermittent fashion. Because nodes are only connected sporadically, an end-to-end path between any two nodes may never exist at once. Instead, communication relies on nodes physically moving and encountering one another. When two nodes come within wireless range (e.g. via Bluetooth or Wi-Fi Direct), they exchange messages and possibly carry each other's messages forward. This *episodic forwarding* continues until messages eventually reach their destinations or expire. The network is thus a delay/disruption-tolerant network (DTN) – delays can range from minutes to days, and nodes must store data in between contacts [18][19].

Classic examples of OppNet applications include:
(a) *Wildlife And Environmental Monitoring***:** sensor nodes on animals or drifting instruments exchange data when close, forming a sparse network

(b) *Vehicular Oppnets:* vehicles (cars, buses) opportunistically swap data – for instance, for traffic updates or urban sensing – when they pass each other or roadside units

(c) *Pocket Switched Networks:* humans' mobile devices (phones, tablets) share content directly when people meet (leveraging social interactions), enabling services like file sharing or emergency messaging off the grid. In all these cases, OppNets enable communication "when possible" rather than "always on."

OppNets are grounded in the *store-carry-forward* principle. Each node acts as a carrier of messages: it stores incoming messages in a buffer, carries them while moving, and forwards them upon meeting another node that is deemed a good next hop. This paradigm makes OppNets highly tolerant to disruption – even if the network partitions, messages are not lost but rather held until the partition heals via node mobility. However, it also means that performance depends heavily on node mobility patterns and contact opportunities. If the network is very sparse or mobility is limited, delays will be extremely high or delivery may fail. To improve delivery

many OppNet routing protocols employ replication: a message may be duplicated to multiple carriers to increase the likelihood it reaches the destination. For example, *Epidemic routing* floods copies to every contact (ensuring delivery if a path exists eventually, but with high resource cost) [22][24][25].

The challenging conditions of OppNets can be summarized by a few key characteristics:

a) *Frequent Disconnections* (nodes spend significant time out of range of others),
b) *Variable Link Quality* (contacts may be short or have low bandwidth),
c) *Low Duty Cycle* (nodes like phones may turn radios off or sleep to save power, limiting contact opportunities),
d) *Limited Resources* (battery power, buffer memory, and communication bandwidth are scarce)
e) *Unpredictable Mobility* (node movement can be random or influenced by human behavior, making contacts stochastic). These factors complicate tasks such as routing, which must decide when and to whom to forward messages without a global view of the network. Traditional routing protocols assume stable connectivity and thus fail in OppNets, where they might never find a contemporaneous route. Designing OppNet protocols requires incorporating uncertainty and tolerance to long delays.

2.2 Machine Learning.

ML is a broad field of algorithms that allow computers (or network nodes, in this context) to learn patterns and make intelligent decisions based on data. Rather than following fixed if-else rules, an ML-based system improves its performance as it "experiences" more data. The main branches of ML include: *supervised learning*, where models are trained on example input-output pairs (e.g. past contact patterns → next contact prediction) ;*unsupervisedlearning*, where models discover structure in unlabeled data (e.g. clustering nodes by mobility behavior) ; and *reinforcement learning (RL)*, where an agent learns policies by interacting with an environment to maximize a reward (e.g. a node learns a forwarding policy that maximizes delivery probability) In OppNets, ML can similarly be used to predict future connectivity, classify node characteristics (e.g. detect "social" hubs or selfish nodes), and make adaptive routing decisions [26][27][28].

3. Literature Review

Research on merging Machine Learning with Opportunistic Networking has accelerated in the past decade. Early surveys noted that while OppNet research was rich in mobility modeling and protocol design, the use of ML in this field was nascent. Recently, however, numerous studies have applied learning-based techniques to OppNet problems.
We organize the discussion by topical sub-areas detailed below:

3.1 Supervised Learning for Routing

Several OppNet routing protocols leverage supervised ML algorithms (classification and regression) to decide when and to whom to forward messages. For example, KNNR (K-Nearest Neighbor Routing) by Abdulkader *et al.* [1] trains a K-Nearest Neighbor classifier to choose the next hop. In their approach, each node uses features like its encounter frequency with others and queue length to classify whether a given neighbor is a "good forwarder" or not. The KNN model was trained offline on simulated network data, and then embedded in the routing protocol. Similarly, SVM-BDT (Support Vector Machine with BrownBoost) is a supervised approach in MANETs that was later adapted to OppNets – it trains an SVM classifier (boosted by BrownBoost ensemble) to predict link quality and uses that to forward messages, achieving energy-efficient routing. Supervised ML is effective when past network behavior is predictive of future events, which is often the case in settings with repetitive mobility (e.g. daily routines) [3][5][6].

3.2 Reinforcement Learning (RL) for Routing

RL has gained traction as a way for nodes to *learn by doing* in OppNets. A prominent example is **QLearning-based routing**. In some recent research work, Researchers applied Q-learning to an

OppNet routing problem, modeling each node as an agent that learns a Q-value for "forward this message to encountered node X" vs "wait" [3].

3.3 Deep Reinforcement Learning (deep RL)

Kaviani *et al.* [7] introduced *DeepMPR*, a multi-agent deep RL routing scheme for wireless opportunistic networks that uses deep Q-networks to train relay selection policies in complex network scenarios . They report significant gains in delivery rate and latency over classical OLSR and even over single-agent RL, thanks to multi-agent coordination. It is worth noting that RL approaches often aim to maximize a cumulative performance metric (e.g. end-to-end delivery probability or minimize delay), and they naturally handle the sequential decision nature of OppNet routing (deciding a sequence of forwards).

3.4 Neural Networks and Deep Learning

Beyond their use as function approximators in RL, neural networks have been applied directly for prediction tasks in OppNets. A key problem is link prediction – i.e., forecasting which nodes will meet (or how strong their connection will be) in the future. Cai *et al.* [2] employed a recurrent neural network (RNN) to predict future contacts for each node pair in an OppNet . Each node fed the sequence of past inter-contact times with another node into an RNN, which then predicted the probability of contact in the next time window. This was used to route messages to those likely to meet the destination.

Similarly, Liao *et al.* [8] used a deep belief network (DBN) for link prediction in OppNets, combining a random-walk based feature generator with a DBN classifier. These deep models outperformed simpler predictors like Markov chains or regression, especially in networks with complex, nonlinear mobility patterns. Qafzezi *et al.* [10] proposed a fuzzy routing protocol for vehicular OppNets where a fuzzy inference system (with rules learned from data) decides whether to forward, incorporating node speed, direction, etc. . Overall, deep learning in OppNets is still emerging, partly limited by the small sizes of OppNet datasets available for training. Yet, as more data (e.g. from city-wide experiments) become available, we expect increased use of graph neural networks, LSTMs, and other advanced models to capture the temporal-spatial patterns of node mobility.

3.5 Social and Context-aware Learning

A distinctive feature of many OppNets (especially those formed by humans or vehicles) is that node interactions are driven by **social patterns** or schedules. Souza et. al. [12] have exploited this by using ML to infer social structures and use them in forwarding. *The Friendship and Selfishness Forwarding (FSF)* protocol is a prime example that incorporates social tie strength and node selfishness into routing decisions using ML techniques. FSF builds a *friendship graph* of nodes (where edge weights indicate frequency/duration of past encounters) and uses a combination of supervised learning and rule-based logic to decide forwarding: a node will only forward a message to encountered node *C* if *C* is a friend or close acquaintance of the destination *B*, and if *C* is not overly selfish (willing to carry others' data) and has sufficient resources .These conditions were tuned based on data; essentially FSF learned thresholds for what constitutes a "friend" (e.g. >5 contacts per week) and what battery level is considered non-critical, etc.

3.6 Mobility Prediction and Scheduling

Another line of work uses ML to predict node mobility or contact schedules, which can then be used to improve data delivery or plan forwarding. Traditional mobility models (random walk, community-based models) are often too generic. As an alternative, researchers have trained models on real mobility datasets. For example, neural network models have been trained to predict a node's next location or the time until its next contact. Zhang et. al. [17] applied a deep learning approach to predict human mobility in OppNets and showed improved accuracy over Markov models in forecasting encounters. These predictions can feed into routing metrics – e.g., a node can skip forwarding to someone if the model predicts a better contact soon (this concept

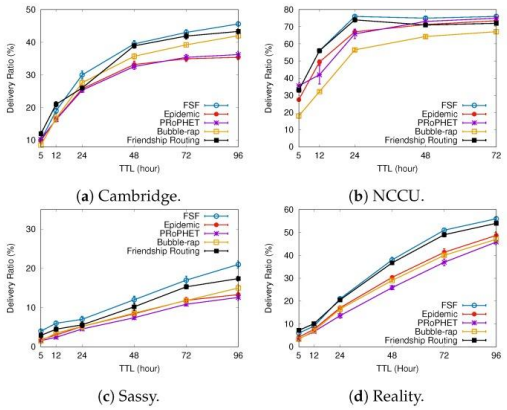

Figure 2: Delivery ratio vs. message TTL for various routing algorithms in an example study (FSF protocol) . As the allowed TTL increases (x-axis, in hours), all protocols deliver more messages, but the ML-guided FSF consistently achieves the highest delivery ratio in four real-world trace scenarios (Cambridge, NCCU, Sassy, Reality). This illustrates the measurable gains possible by incorporating social-tie learning and adaptive decisions in OppNet routing.

was demonstrated in RL routing as well, where the learned Q-values implicitly encode future contact prospects). In vehicular OppNets, where mobility might follow bus routes or traffic patterns, researchers have used techniques like support vector regression to predict vehicle meeting times and destinations. Such predictions enable more effective data muling and offloading.

3.7 Message Scheduling

When a node has multiple messages and limited contact time, which should it send first? ML can learn to prioritize messages that have higher urgency or lower remaining TTL, or those for which the current encounter is especially promising (e.g. meeting a node that rarely comes by). By formulating scheduling as a classification (send vs. wait) or regression (predict delivery likelihood if sent now vs later), nodes can dynamically schedule transmissions. One study formulated this as a multi-armed bandit problem and used reinforcement learning to decide which message to transmit in each contact to maximize overall delivery. Such intelligent scheduling is increasingly important in congested OppNets to avoid buffer overflow and wasted contact opportunities.

3.8 Real-World Deployments and Case Studies

A crucial aspect of OppNet research is how solutions perform on *real data* and in field deployments. Several works have evaluated ML-driven OppNet techniques on real mobility datasets or through testbed experiments. For instance, Rashidibajgan*et al.*[11] used three real trace datasets (GPS logs of taxis in San Francisco and Rome, and buses in Münster) to evaluate an ML model that predicts the optimal network parameters (like number of message copies, buffer size needed) for maximizing delivery in OppNets. They trained a predictive model using features extracted from these traces (e.g. node density, mobility speed) and found it could recommend parameter settings that improve delivery by 5–18% over default settings in protocols. This kind of study bridges simulation and reality, showing that ML can handle the variability of real-world movement. Another case is the Opportunistic Data Offloading scenario:

Han *et al.* [5] deployed an OppNet system where smartphones offload video traffic opportunistically to nearby devices (eventually to reach an access point), and they used a learning-based scheme to decide which content to offload based on context (time of day, density of users). This reduced cellular usage significantly. In the domain of public transportation OppNets, some pilot projects equipped buses with devices that learn and share routes for data collection (e.g. the DTN code project); ML was used to cluster buses by route similarity and schedule message ferrying accordingly. These case studies underscore that ML techniques can be successfully applied outside of simulation, provided that they are trained on relevant data and account for practical constraints (like limited training time on resource-constrained hardware). the studies reviewed, a common theme is that ML-enabled OppNets outperform traditional

approaches under a variety of metrics. Improvements in delivery probability on the order of 5–15% are frequently reported , along with reductions in latency or overhead.

4. Challenges

While the fusion of ML and OppNets is promising, it also brings forth a set of significant challenges. These challenges stem both from the nature of OppNet environments and the requirements of machine learning techniques.

We discuss the major issues identified in current research:

4.1 Limited and Noisy Training Data

ML algorithms typically require substantial data to train reliable models, but OppNets often yield sparse or noisy datasets. Contacts between certain node pairs may be infrequent, providing limited examples to learn from Mobility traces, when available, represent specific scenarios and may not generalize. Potential solutions include online learning (continuously updating models as new data come in) and transfer learning (leveraging data from one network or time period to inform another). However, online learning in OppNets must cope with concept drift and ensure stability despite constantly arriving data in an uncertain stream.

4.2 Computational and Energy Constraints

OppNet nodes are often resource-constrained devices (phones, sensors, embedded computers). Advanced ML algorithms, especially deep learning, can be computationally intensive and may not be feasible to run on-device in real timeTechniques like model compression and approximate inference have been suggested to make ML suitable for OppNets. Reinforcement learning algorithms must also be carefully designed; naive RL might require extensive exploration (i.e. trying suboptimal forwarding actions) which can degrade network performance during learning. Some works mitigate this by using simulation-based training or safe reinforcement learning that limits bad decisions.

4.3 Need for Ground Truth and Labeled Data

In supervised learning approaches, obtaining labeled training data is a hurdle. For instance, to train a classifier to predict “will delivering via node X succeed?”, one needs examples of successful vs. failed deliveries – essentially requiring many experiment runs or simulations to gather outcomes . In real deployments, it’s hard to get ground truth about optimal routes or global network state, since no centralized observer exists. This overlaps with the emerging idea of Federated Learning in OppNets, where each node trains on its local data and only exchanges model updates (which could be more efficient and privacy-preserving). Federated Learning itself introduces challenges of synchronization and model consistency in an OppNet setting, where nodes meet sporadically.

4.4 Privacy and Security Concerns

Using ML often means collecting detailed data about node contacts, mobility, and possibly user context (such as social relationships, locations visited). This raises privacy concerns:; users might be unwilling to share raw data if it reveals sensitive information about their habits or social circles. Another aspect is that learned models themselves could leak information (through model inversion attacks) – so techniques like differential privacy might be needed when sharing model parameters among nodes.

4.5 Generalization and Adaptability

OppNet environments can differ vastly from one another – a model trained on one scenario (say, an urban downtown with many pedestrians) might not work in another (say, a rural area with few nodes). Ensuring that ML models generalize and adapt is challenging. If a model is overfit to a particular network’s characteristics, its performance may degrade when nodes or mobility patterns change. There is also the issue of evaluation because OppNets are so context-dependent,

comparing algorithms fairly is hard. Different works use different mobility traces or simulation setups. A challenge for the community is to establish common benchmarks and perhaps a repository of diverse OppNet scenarios for testing learning-based protocols.

4.6 Overhead and Complexity of ML Integration

Incorporating ML can add significant overhead in terms of computation, storage, and even additional communication (for exchanging ML model information or feedback). In some cases, this overhead could offset the gains. Network operators might be wary of deploying routing protocols that behave opaquely due to learned components. There is a need for explainable AI techniques in OppNets, so that the rationale for forwarding decisions can be understood (especially important in scenarios like military or emergency networks where trust in the network's operation is critical).

4.7 Lack of Incentive and Deployment Hurdles

A more socio-technical challenge (not purely technical) is getting real-world adoption of ML-based OppNet solutions. Users might need incentive to participate (e.g., to keep a service running on their phone that learns and forwards others' data) . Without a clear benefit (or compensation), users could opt out, reducing the network's effectiveness. Some recent Research papers highlight the absence of viable business models for OppNets as an impediment to wide deployment. While this goes beyond ML, it intersects: complex ML algorithms might require more user cooperation (sharing data, running background processes), which requires building user trust and providing reward (e.g., access to network services). Thus, a challenge is designing incentive mechanisms. Initial steps in this direction use ideas from game theory and economics – e.g., giving users credits for carrying traffic, or using ML to dynamically adjust rewards for forwarding based on contribution (there's recent work employing ML to optimize these incentive allocations).

5. Future Directions

The integration of machine learning into opportunistic networking is still evolving. Building on the challenges discussed, we outline several future research directions and technological developments that could drive the field forward:

5.1 Advanced Deep Learning and AI Techniques:

As more OppNet data becomes available (through larger experiments or continuous community data collection like the CRAWDAD repository), there is an opportunity to apply advanced deep learning models to OppNet problems. There is also room for hybrid AI, combining logic and learning – e.g., using deep learning to estimate probabilities that feed into a traditional routing algorithm's framework (so the core algorithm remains explainable, but it gains a "smarter" metric).

5.2 Federated and Distributed Learning:

Given the decentralized nature of OppNets, future solutions are likely to employ Federated Learning (FL) or other distributed learning paradigms to train models across nodes without centralized data aggregation. In Federated Learning, each node would train the model on its local contact data and share only model updates (gradients) during encounters with peers or periodically to some fusion point. This approach keeps personal data local (improving privacy) and naturally fits OppNet operation – model updates can spread through the network similar to how messages do. Recent advancements in FL address issues like limited communication (by compressing updates) and device heterogeneity, which align with OppNet constraints.

5.3 Peer-to-Peer Federated Learningin OppNets

Nodes could average their models when they meet (since encounters are random, this becomes a decentralized averaging scheme). Ensuring convergence of such an algorithm in an OppNet (which is essentially a time-varying graph) is a research question. Some initial work in mobile networks indicates gossip-based model averaging can work, but more theoretical and practical

exploration is needed. Federated Learning could also incorporate incentive mechanisms (giving nodes reward for participating in training, not just routing), combining two challenge areas into a unified solution.

5.4 Edge Computing and 5G/6G Integration

As 5G and forthcoming 6G networks roll out, with them comes the concept of Multi-access Edge Computing (MEC) – servers at the edge of the network (e.g., Base Stations or access points) that can offload computation from devices. Future OppNets might integrate with edge infrastructure intermittently. A future direction is designing protocols that detect when an OppNet node has a high-bandwidth low-latency link to an edge server and opportunistically utilize it to improve the ML aspects of OppNet routing. For example, a node might upload a chunk of contact logs to an edge server when possible; the server could run a more complex analysis (like social network analysis or community detection algorithm) and send back distilled insights (e.g., "you are in community 3, use routing policy X"), which the node can then use offline during pure opportunistic contacts. This kind of opportunistic offloading of ML tasks ties into broader research on distributed AI in networks.

5.5 Privacy-Preserving Learning and Security Enhancements

To tackle privacy concerns, future OppNet ML designs will likely incorporate techniques like differential privacy, secure multi-party computation, and secure aggregation. Some initial work in this vein uses reputation systems enhanced by ML to identify anomalous behavior (e.g., a node that always claims it will forward but never does) . The future might see adversarial machine learning considerations: ensuring that routing decisions are not easily manipulable by crafted inputs. For example, an attacker could try to spoof contact events to confuse a learning algorithm – defenses like anomaly detection or validating contacts via multi-factor signals (Bluetooth + physical sensors) could mitigate this.

5.6 Cross-Layer and Contextual Learning

Future ML-OppNet solutions may expand beyond the network layer (routing) into other layers and aspects of the network stack. For instance, MAC layer learning could help in scheduling contact durations – if a node has learned that it often meets two other nodes simultaneously, a learning-based MAC could prioritize which node to exchange with first based on message importance. Transport layer could benefit too: an ML model could decide when to start transmitting a bundle of data given predicted contact duration to avoid mid-transfer cuts. Beyond networking context, integrating contextual data (from sensors on the device or external sources) is a promising direction. These ideas intersect with delay-tolerant networking in smart cities and IoT, where context and networking interplay.

5.7 Evaluation in Diverse Scenarios & Standardization

As research matures, it will be important to validate ML-driven OppNets in a wide range of scenarios: dense vs. sparse networks, high-mobility vs. pedestrian, with varying data loads, etc. One future direction is establishing benchmark scenario suites (possibly leveraging networks of autonomous vehicles, drone swarms, or large-scale simulations) to evaluate learning algorithms. Standardizing interfaces for ML components in OppNet protocols could also help with broader adoption – for example, an API such that any routing protocol can query a prediction service (which could be implemented with different ML models) for contact probabilities. This would allow plug-and-play experimentation and eventually standardization once a particular approach proves generally superior. Moreover, as part of future directions, it's worth noting the potential for real-world trials.

6. Conclusion

In conclusion, the fusion of OppNets and ML stands out as a promising path to greatly improve communications in environments where conventional networks are unreliable or absent. By enabling nodes to learn from experience, we move away from one-size-fits-all protocols toward smarter networks that can adapt to their context – a long-standing goal in networking research.

The work summarized in this paper lays the groundwork for that vision, and we hope it will spur further innovation at this exciting intersection of mobile networking and machine intelligence.

References

[1] Abdulkader, B., Launay, P., &Guidec, F. (2015). *Solving Consensus in Opportunistic Networks*. **International Journal of Distributed Sensor Networks, 11**(10), 1-12..

[2] Cai, X., Shu, J., & Al-Kali, M. (2018). Link prediction approach for opportunistic networks based on recurrent neural network. **IEEE Access, 7**, 2017–2025

[3] Chaudhary, S., & Johari, R. (2020). ORUML: Optimized routing in wireless networks using machine learning. **International Journal of Communication Systems, 33**(7), e4394

[4] Dhurandher, S. K., Singh, J., Obaidat, M. S., et al. (2020). *Reinforcement learning-based routing protocol for opportunistic networks*. In **Proc. IEEE ICC 2020**, 1–6.

[5] Han, B., Hui, P., Kumar, V. S. A., et al. (2011). Mobile data offloading through opportunistic communications and social participation. **IEEE Transactions on Mobile Computing, 11**(5), 821–834.

[6] Jiang, J., Han, G., & Lin, C. (2023). A survey on opportunistic routing protocols in the internet of underwater things. **Computer Networks, 219**, 109658.

[7] Kaviani, S., Ryu, B., Ahmed, E., et al. (2023). DeepMPR: Enhancing opportunistic routing in wireless networks via multi-agent deep reinforcement learning. In **Proc. IEEE MILCOM 2023**, 51–56.

[8] Liao, Z., Liu, L., & Chen, Y. (2020). A novel link prediction method for opportunistic networks based on random walk and a deep belief network. **IEEE Access, 8**, 16236–16247.

[9] Mota, V. F., Cunha, F. D., Macedo, D. F., et al. (2014). Protocols, mobility models and tools in opportunistic networks: A survey. **Computer Communications, 48**, 5–19.

[10] Qafzezi, E., Bylykbashi, K., Higashi, S., et al. (2025). An intelligent fuzzy-based routing protocol for vehicular opportunistic networks. **Wireless Personal Communications, 122**(1), 489–507.

[11] Rashidibajgan, S., &Hupperich, T. (2022). Improving the performance of opportunistic networks in real-world applications using machine learning techniques. **Journal of Sensor and Actuator Networks, 11**(4), 61.

[12] Souza, C., Mota, E., Soares, D., et al. (2019). FSF: Applying machine learning techniques to data forwarding in socially selfish opportunistic networks. **Sensors, 19**(10), 2374.

[13] Spyropoulos, T., Psounis, K., & Raghavendra, C. (2005). Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In **Proc. ACM SIGCOMM Workshop on Delay-Tolerant Networking**, 252–259.

[14] Wang, M., Cui, Y., Wang, X., et al. (2018). Machine learning for networking: Workflow, advances and opportunities. **IEEE Network, 32**(2), 92–99.

[15] Wu, J., Guo, Y., Zhou, H., et al. (2020). Vehicular delay tolerant network routing algorithm based on Bayesian network. **IEEE Access, 8**, 18727–18740.

[16] Wu, Y., Zhao, Y., Riguidel, M., & Yi, P. (2015). Security and trust management in opportunistic networks: A survey. **Security and Communication Networks, 8**(9), 1812–1827.

[17] Zhang, Z., Zhou, S., & Xu, Z. (2019). Delay-minimization routing for heterogeneous VANETs with machine learning based mobility prediction. **IEEE Transactions on Vehicular Technology, 68**(4), 3967–3979.

[18] Zheng, Y., Zhao, L., & Zhou, W. (2023). Understanding mobility behavior in aging people via opportunistic networks: A self-sustainable solution. **Wireless Networks,** (Online first).

[19] Zhou, C., Eugenio, N., & Louis, S. (2019). Friendship and selfishness forwarding: Applying machine learning techniques to opportunistic networks data forwarding. **arXiv preprint arXiv:1705.08808**

[20] P. Garg, A. Dixit and P. Sethi, "Ml-fresh: novel routing protocol in opportunistic networks using machine learning," Computer Systems Science and Engineering, vol. 40, no.2, pp. 703–717, 2022. DOI:10.32604/csse.2022.019557

[21] Garg, P., Dixit, A., Sethi, P., & Pinheiro, P. R. (2020). Impact of node density on the qos parameters of routing protocols in opportunistic networks for smart spaces. Mobile Information Systems, 2020. https://doi.org/10.1155/2020/8868842

[22] N. Puri, P. Saggar, A. Kaur and P. Garg, "Application of ensemble Machine Learning models for phishing detection on web networks," 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), 2022, pp. 296-303, doi: 10.1109/CCiCT56684.2022.00062.

[23] Garg, P., Dixit, A., Sethi, P., & Pruthi, J. (2023). Strengthening Smart City with Opportunistic Networks: An Insight. https://doi.org/10.1109/icacctech61146.2023.00117

[24] Singh, M., Garg, P., Srivastava, S., & Saggu, A. K. (2024, April). Revolutionizing Arrhythmia Classification: Unleashing the Power of Machine Learning and Data Amplification for Precision Healthcare. In 2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT) (pp. 516-522). IEEE. DOI: 10.1109/CCICT62777.2024.00086

[25] Kumar, R., Das, R., Garg, P., & Pandita, N. (2024, April). Duplicate Node Detection Method for Wireless Sensors. In 2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT) (pp. 512-515). IEEE. DOI: 10.1109/CCICT62777.2024.00085

[26] Garg, P., Dixit, A., & Sethi, P. (2021, May). Link Prediction Techniques for Opportunistic Networks using Machine Learning. In Proceedings of the International Conference on Innovative Computing & Communication (ICICC). http://dx.doi.org/10.2139/ssrn.3842849

[27] Garg, P., Dixit, A., & Sethi, P. (2021, April). Opportunistic networks: Protocols, applications & simulation trends. In Proceedings of the International Conference on Innovative Computing & Communication (ICICC). http://dx.doi.org/10.2139/ssrn.3834099

[28] Garg, P., Dixit, A., & Sethi, P. (2021). Performance comparison of fresh and spray & wait protocol through one simulator. IT in Industry, 9(2). https://doi.org/10.17762/itii.v9i2.369