International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

Intelligent Route Optimization for Secure and Efficient Network Traffic Management Using Machine Learning Algorithms

[1]**Sikander**
Research Scholar, Department of Computer Science and Engineering, Om Sterling Global University, Hisar, India
scientistsikander@gmail.com, sikander.m@nic.in
[2]**(Dr.) Rajender Singh Chhillar**
Pro Vice-Chancellor Om Sterling Global University, Hisar, India
pvc@osgu.ac.in
[3]**Sandeep Kumar**
Professor Department of CSE, Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India
er.sandeepsahratia@gmail.com

Abstract – This study presents contemporary communication systems, it is essential to regulate network traffic in a manner that is both efficient and secure. Many routing algorithms exhibit issues such as insufficient accuracy, prolonged processing times, inability to manage high traffic volumes, lack of security, and inadequate real-world testing. This study proposes an enhanced route selection algorithm that employs machine learning to optimise routing efficiency, enhance detection accuracy, and elevate overall network performance. constructed a customised dataset by emulating a network comprising both legitimate and malicious traffic. Also trained and evaluated four machine learning models: Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine (SVM). Employed significant performance metrics to do this. The most efficient model was Random Forest, with the highest accuracy (96.86%), detection efficiency (98.64%), and a significantly reduced stolen packet rate of 1.00%. It demonstrated superior network performance with a packet delivery rate of 72.40%, reduced average hops, and enhanced path utilisation. The Random Forest-based method effectively identified assaults by accurately detecting malicious behaviour with little false negatives. The results indicate that machine learning-based routing could revolutionise the field, with Random Forest providing the optimal equilibrium among accuracy, security, and computational efficiency. The proposed design significantly enhances traffic management, facilitates scalability, and strengthens security. This addresses significant research deficiencies and paves the way for intelligent, practical network traffic control systems.

Keywords: Optimized Route Selection, Network Traffic Management, Machine Learning, Random Forest Algorithm, Intrusion Detection.

1. Introduction

The internet, virtualisation, the World Wide Web of Things, along with real-time applications have all evolved very quickly in the digital age, which has led to an unprecedented amount of data being sent over communication networks. Modern societies depend more and more on strong, safe, effective network infrastructures to make it easy for people, businesses, and industries to connect with each other[1]–[5]. As the need for fast data transfer, streaming media, online shopping, smart cities, and smart transportation systems grows, network traffic oversight has become an important field of study and development. Choosing the optimum paths for data transmission is one of the hardest things for network managers to do.[6]–[11]. This is because it directly affects capacity, energy utilisation, and the user's overall experience.[12]–[15]. Network Traffic Governance (NTM) is the act of watching over, managing, or speeding up the transfer of data packets across the internet in order to ensure sure it's safe, reliable, and quick. Route choosing algorithms are particularly significant in this discipline for finding the optimal way for data items to move from one point to another Early networks used outdated routing schemes like Dijkstra's or Bell man-Ford to help them figure out which way to go. But these old-fashioned means of finding the best solutions don't always work well because today's networks are so advanced. They have dynamic topologies, numerous kinds of devices, varying quality of service (QoS) needs, and the potential cybersecurity risks [16].

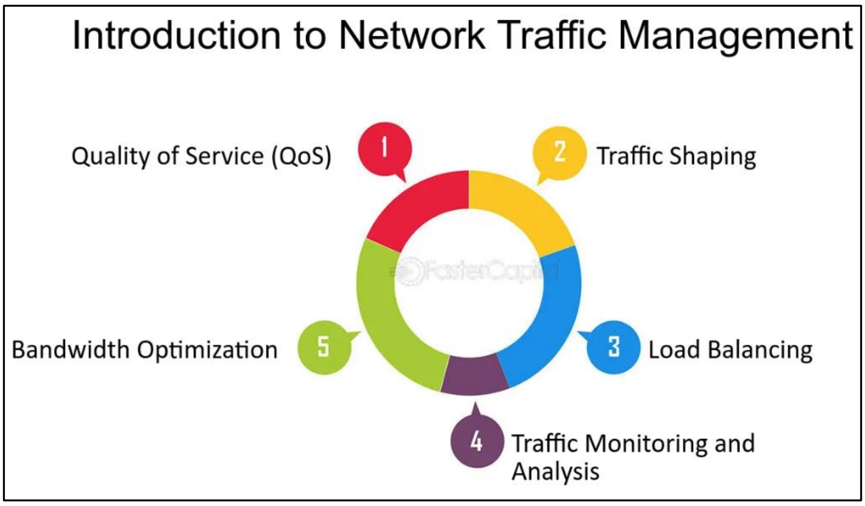

Figure 1 Network Traffic Management

The existing literature indicates that routing algorithms exhibit several issues. Many old systems can't handle changes in conditions, have problems growing as connections get bigger, and are vulnerable to hacks that threaten route security. Also, these algorithms typically have drawbacks when they are employed in the real world, such as taking too awhile to process, not being enough precise, and not using energy efficiently, especially in big networks. These difficulties highlight how crucial it is to discover better methods to deal with changes, make things safer, and use materials more wisely. This research is deficient in multiple aspects. To begin with, there isn't much scholarly work that looks at how to make optimised route selection computations for modern network systems that control traffic. Second, there aren't enough standardised datasets for creating and testing routing schemes, which makes it hard to compare studies.[17]–[21], Modern methods can miss important things like energy efficiency, computing complexity, and the ability to react in real time. Another huge problem is that we haven't delved into deep learning (ML) as well as computer vision (AI) methods enough. These methods are very promising for figuring out traffic patterns, changing routes when you fly, and making everything operate better. So, a good way to move the subject forward is to come up with the best way to choose a route that takes into account safety, efficiency, and predictive data.[22], [23].

This study introduces an innovative approach for creating an optimal Route Selecting System through the application of machine learning techniques. The first thing to do is figure out what the problem is and collect the data set from a public source, such Kaggle or the School of Berkeley, University of California, Machine Learning Repository. To get rid of anomalies like lacking numbers, noise, and inconsistencies, the data must be cleaned up before it can be evaluated. The next stage is to train a recommended ensemble-based model. This model uses the best elements of various machine learning techniques to create predictions that are more accurate and reliable.[24]–[29], We use strict performance measures to assess the model and compare it to other AI methods. We use methods like K-Fold cross validation to make sure the results are accurate and can be used in many different situations. Finally, the prototype has been tested in real life, which shows that it could be useful.[30]. There are many reasons for this research. First, it wants to see how traffic routes are handled currently and what doesn't work. Second, it intends to improve the accuracy and reliability of network administration tools by employing a new way to do arithmetic. Third, it stresses how important it is to make picking routes systems safer so that no one can get to or edit data without permission. Finally, the solution that was suggested will be tested against the best methods that already exist to make sure it works well and can be used in the real world. The study aims to fill a gap and greatly improve the literature on modern network traffic management by achieving these goals [31].

The expected outcomes of this project are an improved and safer routing system, reduced processing times, enhanced scalability, and decreased energy consumption during network operations. An ensemble approach to machine learning also makes sure that the system is not only accurate, but also able to handle alterations to traffic and topology. These advancements could lead to the development of new traffic management algorithms that can meet the growing needs of lightning-fast networks in areas including smart transportation, telecommunications, military, e-commerce, and cloud-based services. The need for systems of routing that are quick, safe, and optimised is growing as data networks get more intricate and in demand. The limitations of traditional models necessitate innovative techniques that leverage the predictive and adaptive features of machine learning. This study seeks to rectify this critical shortcoming by formulating and validating an enhanced route selection methodology that elevates security, efficiency, and practical utility. The suggested system has the ability to change how modern networks handle traffic by combining strict methodology with practical implementation. This would lead to more reliable and long-lasting communication infrastructures.[32]

2. Literature review

Liu 2025 et al. offer a complete sustainable optimisation solution that combines three modules to improve vehicle routes, signals, and networked autonomous vehicle (CAV) trajectory estimation at the network level. The route guidance module identifies the best ways to get the most cars through, the signal optimisation module alters the timing on the fly to speed things ahead, and the planning of trajectory module determines the ideal speeds to make the journey more relaxing and cut down on delays. These modules broadcast outputs to another in real time at every iteration, which maintains them in sync. Use linearisation and decomposition, along with the Dijkstra algorithm, programming with variables, or linear programming, to speed up calculations. At junction and lane levels, signal optimisation and forecasting of trajectory are divided down into smaller jobs. Experimental simulations with varied network configurations and traffic volumes show that the framework can be scaled up and works well. The results demonstrate that travel times are shorter, rides seem more comfortable, or traffic flows better. A side-by-side look at the two indicates that optimising messages is the key to reaching long-term goals for urban transportation[33].

Zhijiang 2025 et al. Cold chain logistics can't work very well when there is a lot of traffic in cities. Overall, they make firms less competitive, make offerings less fresh, cost more to ship, and take longer to deliver. To cope address these problems, I built a model for optimising truck routes. It wants to keep things fresh, save money, or cut down on emissions of carbon. The method looks at how fast cars are driving, how many cars are on the road, and how fast traffic is moving right now. It uses a multi-objective hybrid biological algorithm with big neighbourhood search (LNSNSGA-III) to find the best routes and improve local search. I make everything fresher or the delivery speedier by changing the times they leave. I also use a vehicle mix method, which shows that the three-type bike method works better on a lot of different measurements. The data indicate how prices and emissions change depending on the weather and how hot it is. This can help you please run a cold chain that is good for the environment. The framework provides a good balance between long-term viability, service quality, and efficiency. In the future, projects will use real-time traffic information along with plans designed particularly for them to find more effective and greener ways to handle logistics.[34].

Jakubec 2025 et al. The architecture, surveillance equipment, or the volume of traffic at a junction can all affect how traffic moves. You need to keep an eye on it in order to be sure it works properly. They used a YOLO-based architecture with camera footage to automatically learn about and investigate how cars move to make this procedure better. Not only does this method make evaluation faster than completing it by hand, but it also gives you additional knowledge, like the speed and spacing of vehicles, that are hard to get in other ways. As part of a trial project, the device was used at a junction in Zilina, Slovakia. For finding passenger cars, lorries, and buses, the YOLOv9c design has a mAP50 of 98.2%. This means that it was pretty good at finding stuff. I did see some discrepancies between automatic findings and hand evaluations, especially when it comes to keeping track of when vehicles came and went. The average absolute error for passenger autos was 2.73 every 15 minutes. These results show that robotic detection can make monitoring network flow more accurate, faster, and easier to scale [35].

Zhang 2025 et al. Intelligent public transportation (ITSs) are great for controlling city traffic because they use traffic flow prediction to help people plan their travels better and avoid traffic jams. The Linear Focus Based Space-Time Multiple Graph CNN (LASTGCN) is a form of deep learning that was created just for predicting traffic flow. The model uses a Multifactor Combination Unit (MFF-unit) to constantly combine weather data with a multi-graph congruent structure to find geographical correlations. The receptor Heavy Key Values (RWKV) block makes processes work better by using linear attention. This makes it better than models that employ transformers to handle a lot of data. Design makes the computer work better, therefore it's good for managing traffic in the short term and can be done in real time with more effort. When tested on real highway traffic datasets, these kinds of algorithms are better and safer than the finest ones we have now, especially when it comes to making accurate forecasts for the future. It gets a lot more accurate making predictions when you add events like the climate to the model. This means that it works well in places where people drive to work every day [27].

Samaniego 2024 et al. Adding WSNs or the Internet of Things, also known as IoT, to VANET infrastructure in a smart way can make vehicles safer, control traffic, and employ a wide range of apps by collecting data on roads and traffic instead of relying on regular internet connections. In places with poor coverage, alert systems can work. In regions with a lot of traffic, emergency alerts can be sent. Environmental measuring can happen without requiring TCP/IP. To make vehicle-to-everything (V2X) connections as quick and useful as possible, network technologies, data collection devices, clustering methods, and energy-saving routes are all used. You can make apps that watch traffic, safety, or ecology without hindering the network by employing 802.11p frequency channels. We look at how algorithms for clustering and energy-efficient strategies have improved recently in order to perform VANET tasks better. This comprehensive design lays the groundwork for powerful, efficient, and expandable vehicular networks. These networks will make commute systems more secure, efficient, and more adaptable[36].

Table 2.1 Literature Summary

Author/Year	**Technique**	**Findings**	**Research Gap**
Deng et al., 2025 [11]	Actor-Critic Optimization Approach for traffic scheduling in data centers using Google Cluster Usage Traces dataset	Improved throughput, reduced job failure rates, adaptable under varying traffic loads, and more stable than traditional scheduling approaches.	Limited scalability in very large networks; lacks integration with advanced AI techniques like multi-agent RL, federated learning, and GNNs for broader generalization.
Zarko et al., 2025 [37]	Graph theory and Dijkstra's algorithm with traffic counters, AutoLISP 2022 and AutoCAD 2022	Demonstrated accurate route optimization in Sarajevo's traffic network; validated against Google Maps; scalable and flexible for city traffic analysis with support for signalized/non-signalized junctions.	Limited to networks with traffic counters; does not integrate real-time adaptive learning; applicability in larger smart city ecosystems not fully explored.

Ren et al., 2025 [17]	Distributed Network Traffic Scheduling using Trust Region Policy Optimization (TRPO) on Abilene datasets	Outperformed DQN and PPO with lower congestion, better delays, and high stability under heavy traffic; scalable and generalizable with policy-based RL.	Still limited in large-scale deployments; lacks integration with multi-agent RL, GNNs, and hybrid AI methods for real-world traffic management.
Álvaro et al., 2025 [18]	UAV-based monitoring + IoT sensors + Large Language Models (LLMs) with SUMO simulations	Significantly reduced traffic congestion and CO_2 emissions; real-time adaptive traffic flow management; effective in urban areas (San Diego, Madrid).	High dependency on UAV/IoT infrastructure; energy and cost efficiency not fully addressed; integration into existing urban ITS systems remains a challenge.
Ru et al., 2024 [38]	GC-YOLOv9 algorithm (Ghost Convolution + YOLOv9) integrated with IoT, edge processing, and smart city data centers	Achieved strong object detection accuracy (mAP@0.5: 77.15 on BDD100K, 74.95 on Cityscapes); supports traffic hazard detection, fire monitoring, and intelligent security systems; scalable within smart city frameworks.	High computational demand; real-world deployment costs remain high; limited exploration of integration with route optimization and traffic flow prediction systems.

3. Methodology

The proposed research is to develop and execute an optimised route selection algorithm for network traffic management systems that improves accuracy, security, scalability, and efficiency, while mitigating the limits present in current methods. Dynamic topologies, high mobility, and rising security concerns are making network environments more complicated. The performance needs can't be reached by old routes and static optimisation approaches. They usually have issues with being not accurate enough, not having adequate safety measures, taking a while to process, and not being able to grow. Also, it's much tougher to create powerful and clever routing systems when there aren't clear datasets and not enough validation in the actual world. This work introduces a comprehensive technique that integrates simulation-based dataset generation, machine learning-driven intrusion detection, adaptive route optimisation, or multi-metric performance assessment into a unified framework to tackle these challenges. The methodology seeks to systematically address the research deficiencies mentioned in the literature. To start, it creates a realistic set of data by simulating a smartphone Ad Hoc NET

(MANET) environment that includes both normal communication scenarios or malicious attacks, so wormhole assaults. This plan solves the main problem of not having ample public datasets for managing traffic. It also makes sure that the equipment has been evaluated and trained on data that is accurate and different. The dataset has a lot of routing features, like hop count, node rapidity, direction fluctuation, and path length. All of these things have a big impact on how routes are chosen. You can become ready for predictive optimisation or decision-making by finding these feature sets and learning about their security and legality. The next step is to use supervised neural networks on this dataset to develop a model that can tell the difference between safe and harmful routes. Test out models like Decision Tree, Support Vector Machines, Random Forests, and Logistic Regression to discover how well they work. Then, the primary routing algorithm gets the best model. This technology helps people make better decisions, makes the network safer by using smart threat detection, or lets the network automatically optimise based on how it is working at the time. By setting hyperparameters and testing the model with precision, recall, and F1-score, you can be confident that the method works as well as it can and can be used in a lot of different ways. After the model has been trained, it is put into a dynamic simulation environment where it helps people choose routes in real time. The algorithm looks at all the open paths, gets rid of the ones that aren't safe, and picks the best and safest path based on expected classifications and performance metrics. It also adds features like decision criteria based on confidence and rerouting strategies to make the system more reliable and able to deal with changes in network circumstances and attacks. Lastly, the methodology includes a detailed performance evaluation step that compares the proposed solution to the best current methods using metrics like packet delivery ratio, throughput, path utilisation, and detection efficiency. This multi-step methodology fixes all of the main problems with the present methods. Some of these problems include that there aren't enough databases, they aren't safe enough, they are hard to scale, they take too long to look at, and they can't be used in the real world. As a result, a powerful, smart, and flexible route selection architecture has been created that makes traffic management, network performance, and security better. This all-in-one plan makes it feasible to use solutions on a large scale in the real world and has a big effect on the development of new network traffic management systems.

3.1 Data Preparation

The first step in the suggested plan is to build a complete and accurate dataset that includes both normal or bad network conditions. This is very important for making the route pick algorithm better in a system that schedules network traffic. One significant issue with the current study is the absence of clearly defined datasets specifically designed for traffic optimisation as well as vulnerability assessment. The proposed solution creates a mimicked Mobile Ad Ads Network (MANET) setup that works just like a real network to fix this problem. The make_net() function builds up the network by defining crucial things like how many nodes there are, how far they can send data, how fast they can send it, and where they are in a given area. May simulate many different network topologies and conditions for data creation in this controlled environment. After the network is set up, simulated wormhole assaults are used to show one of the biggest risks to network security and route optimisation. The

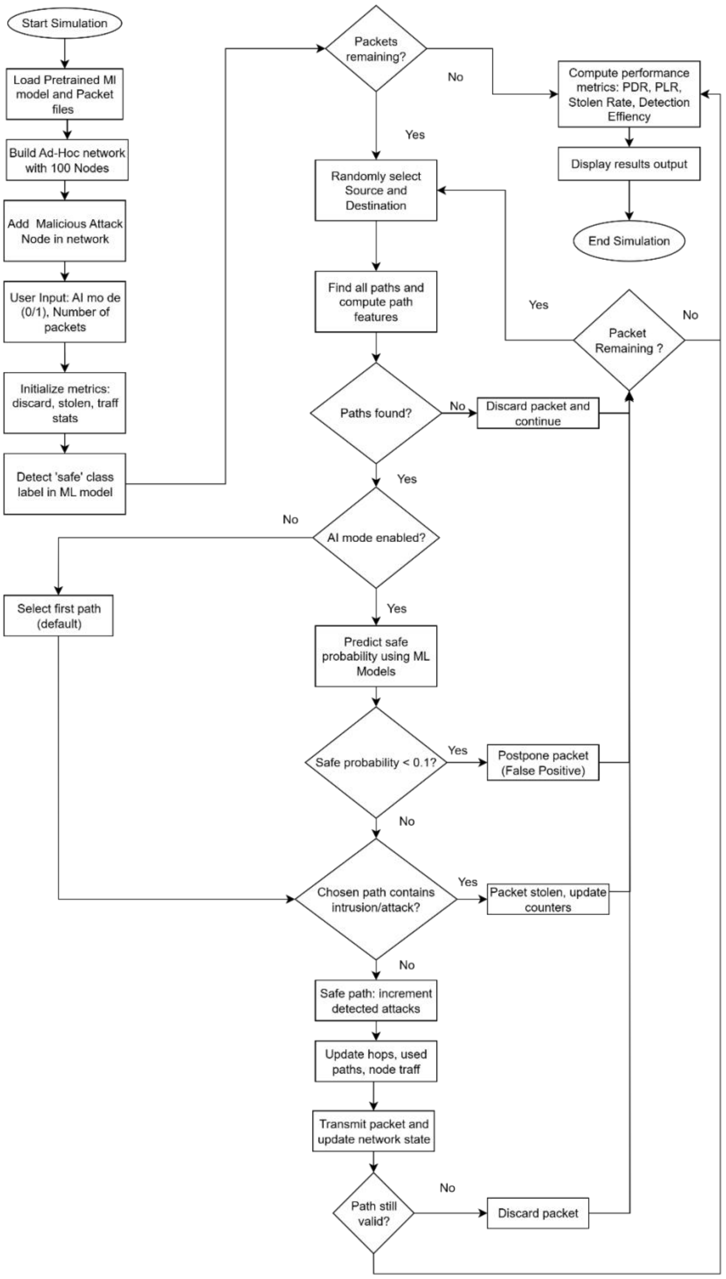
Start Simulation
Load Pretrained Ml model and Packet files
Build Ad-Hoc network with 100 Nodes
Add Malicious Attack Node in network
User Input: AI mo de (0/1), Number of packets
Initialize metrics: discard, stolen, traff stats
Detect 'safe' class label in ML model
Packets remaining?
No
Yes
Compute performance metrics: PDR, PLR, Stolen Rate, Detection Effiency
Display results output
End Simulation
Randomly select Source and Destination
Find all paths and compute path features
Yes
Packet Remaining ?
No
Paths found?
No
Discard packet and continue
Yes
No
AI mode enabled?
Select first path (default)
Yes
Predict safe probability using ML Models
Safe probability < 0.1?
Yes
Postpone packet (False Positive)
No
Chosen path contains intrusion/attack?
Yes
Packet stolen, update counters
No
Safe path: increment detected attacks
Update hops, used paths, node traff
Transmit packet and update network state
Path still valid?
No
Discard packet

Figure 2 Proposed Flowchart

Add Wormhole () function deliberately adds wormhole nodes to the network. This changes their transmission range and speed to make them act as tunnels. This planned assault simulation is very important since security holes can make route selection less efficient and slow down the network. By integrating these kinds of attacks throughout the data production phase, the dataset becomes more complete and more like real-world network situations. The find_all_paths () method finds all the potential routing paths between the source and destination nodes for each communication attempt. The suggested technique gets a wide range of routing features from these paths, such as the number of nodes, the average speed, the direction deviation, the hop distance, and the relative positions. These aspects show important things that affect the performance of the network and the reliability of the path. Using the isContain() and path_isvalid() functions, each extracted feature vector is then given a label that tells whether the path is safe or has been attacked by a wormhole. This makes a labelled dataset where each entry is a possible routing path with its own behavioural traits and security status. The dataset created by this technique fills a major research gap: there aren't enough publically available datasets for research on how to choose the best route. It also incorporates both positive and bad scenarios, which makes it possible to train models that can not only enhance how traffic moves but also discover and stop attacks. This stage ensures sure that the next phases in the process are based on good, accurate information. The simulation framework is also flexible, so you may test scalability by modifying the size of the networks, the number of nodes, and how often assaults happen. This kind of flexibility helps us understand how network dynamics affect route optimisation and gives us a strong base for training and testing algorithms. Stage 1 essentially fills in a lot of research gaps, especially those that have to do with datasets not being available, not thinking enough about security, and not being useful in the real world. It does this by creating a flexible and complete data generation environment that supports the whole proposed methodology.

3.2 Machine Learning Model Training

The second step in the suggested strategy is to make a strong machine learning-based categorisation system that will help with intelligent route selection and security in network traffic management. A significant constraint in current methodologies is the insufficient application of data-driven strategies for path optimisation and assault detection. Most traditional routing algorithms use static or heuristic measurements, which means they don't change when network conditions or security risks change. Our solution addresses this issue by incorporating supervised machine learning classifiers trained on the dataset produced in Stage 1 to effectively differentiate between secure and compromised routes. The labelled dataset, which has path features and their classifications, is split into two parts: a training set and a testing set. To find the best model for the job, use and test several machine learning methods, such as Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), and Logistic Regression (LR). The input features include network-level or node-level parameters including hop count, median speed, orientation deviation, or path length. The output label tells you if a route is safe or not. This binary classification lets the model help with safe and efficient route selection. The dataset is used to train each classifier, and its performance is measured by measures like F1-score, recall, and accuracy. The Random Forest classifier, in particular, works better because it can deal with feature non-linearity and cut down on overfitting. So, it is chosen as the main detection model and saved as RandomForest Model.mat so it can be used in the main simulation, hyperparameter tuning is utilised to enhance model performance, hence filling the research gap about the insufficient discourse on parameter optimisation. To find the ideal settings for tree depth, feature subset size, and number of estimators, techniques like grid search and cross-validation are utilised. This improves both detection accuracy and processing efficiency. As the study goals show, this stage is a big step towards making things more accurate, scalable, and secure. The machine learning model learns patterns that are too complicated for standard algorithms to see. This lets it apply what it learns to new network conditions and attack techniques. Also, the method makes energy use more efficient by minimising bad pathways that could cause wasteful retransmissions and network congestion. The suggested strategy changes the way decisions are made from static rules to adaptive, data-driven intelligence by using machine learning in the route selection process. This sets the basis for real-time optimisation in later phases.

3.3 Intelligent Simulation and Attack-Aware Route Selection

The third stage is about putting the trained machine learning model into the network's dynamic simulation and allowing decisions to be made in real time for the best route choice. Traditional routing methods usually don't respond quickly to security risks or changes in traffic, which can lead to choosing the wrong path, longer delays, and being more open to assaults. Our solution gets around these problems by using the learnt Random Forest classifier in a live MANET environment and using its predictions to choose the safest and most efficient path. The network is rebuilt with the same settings as in Stage 1, such as the number of nodes, the speed, and the range of communication. Malicious wormhole nodes are once again added to mimic real-world attack scenarios. During the simulation, packets are sent between random pairs of sources and destinations, and all possible paths are found. For each route, the machine learning model is trained to look at the important data and decide if the path is safe or harmful. The algorithm quickly gets rid of routes that are too dangerous and then chooses the most safe path based on the findings of the routing and category metrics. Adding a choosing criterion based on trust makes the process even more reliable. If the highest trust model score for the accessible pathways is lower than a specific amount, the packet transfer will be delayed. This safety feature enhances the system more accurate and faster by minimising the risk of packets getting lost or confused. Also, if the first course chosen becomes dangerous or isn't the best choice during operation, the system will reroute right away. This makes confident that the network is always changing to stay ahead of emerging threats and stop them. Adding AI to the education process not only makes the network stronger, but it also makes traffic load balance, shipment ratio, and energy efficiency better. The model's capacity to automatically remove bad routes and adjust to changes in the overall architecture immediately fixes a number of research problems, including as security holes, low accuracy, long processing times, and systems that can't grow. The real-time decision-making basis also makes the device better for use in the real world since it connects abstract models to real-world scenarios.

3.4 Performance Evaluation and Optimization

The last stage in the suggested process is to do a full evaluation of performance in order to see if the improved route algorithm works better than the best methods that are already available. This review is important because it shows that the research has made progress in accuracy, safety, flexibility, or computing efficiency, which are all important goals. To see how the system works in both normal and attack situations, a number of system performance gauges are used. The Packet Transfer Ratio (PDR), Data Damage Rate, Stolen Packet Speed, Throughput, and Detection Efficiency are all instances of these. All of these things show how strong and reliable the system is as a whole. Traffic management options like the median number of hops per request, the traffic balance index, as well the path use rate are also used to see how well routing works and how well the load is spread out. Tests with and without the machine-learned detection mechanism to see how it affects the network's performance. The results show that the proposed method improves all of the parameters that were looked at. The ML-based method has a greater PDR or throughput and less packet loss and damage from attacks. The security-aware routing strategy makes sure that packets get where they should, which reduces down on the number of packets that get stolen. Also, choosing the best path cuts down on unnecessary retransmissions, which saves energy and speeds up processing times. These results directly address the shortcomings in research focused on optimising energy use, increasing computational efficiency, and enhancing accuracy. The test also sees how well the system expands by changing the number of nodes, the amount of traffic, and the ferocity of the attacks. The results show that the suggested strategy works well even when conditions are tough, which implies it might be used in real life. Finally, the suggested solution works better than both traditional routing protocols and current machine learning-based methods. This was the goal of the work.

4. Results discussion

This part displays and speaks about the results of utilising the proposed optimised route selection strategy for managing network traffic. This approach combines machine learning-based surveillance with smart path optimisation. The research employed a custom dataset generated by simulating a Mobile Ad Ad Network (MANET) under both standard or wormhole attack scenarios, as specified in the methodology. The evaluation was executed across four primary dimensions: (i) how well the machine learning model works on the generated dataset, (ii) how well the network works with both traditional and ML-based methods, (iii) how well traffic management works, and (iv) how well attack detection works.These results collectively affirm the efficacy of the suggested methodology in resolving the research deficiencies associated with dataset unavailability, inadequate security, low accuracy, restricted scalability, and suboptimal performance in current systems.

4.1 ML Model Performance on custom created Dataset

Table 4.1 shows how well four supervised machine learning models-Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine (SVM)-compared to each other. Accuracy, Precision, Recall, and F1 Score are all evaluation metrics that give a whole picture of how well each model can predict if network pathways are legitimate or malicious.

Table 4.1 Performance Comparison of Machine Learning Models on Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Decision Tree	94.97	75.86	85.57	80.42
Logistic Regression	96.73	83.64	86.50	85.05
Random Forest	96.86	85.18	84.43	84.80
SVM	95.60	82.25	68.79	74.92

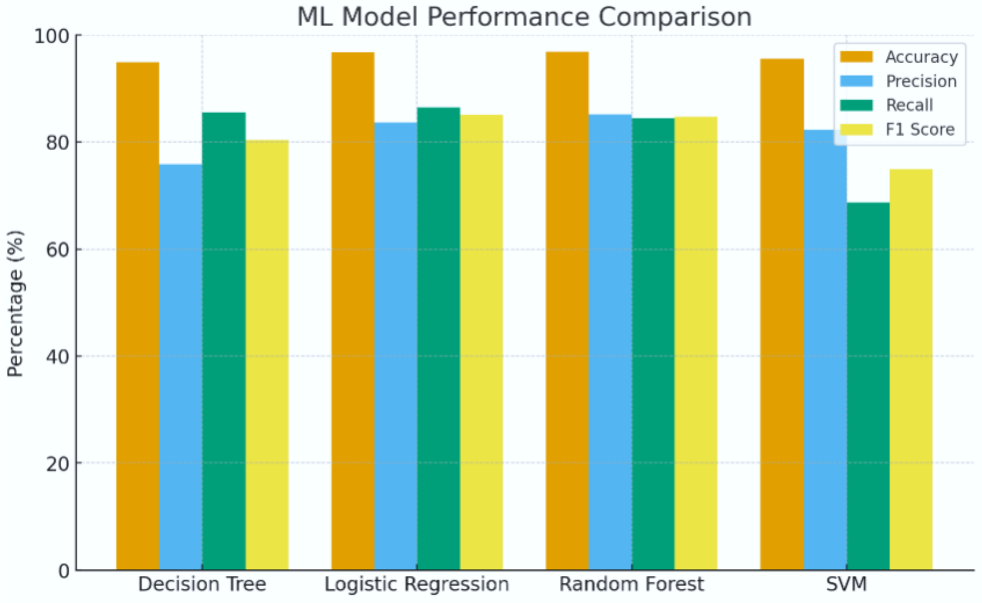

Figure 3 Performance Comparison of Machine Learning Models

The table above shows how well four machine learning models-Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine (SVM)-did on a dataset that was developed just for this purpose. Employ Accuracy, Precision, Recall, and F1 Score as performance measurements. These four metrics provide us a complete picture of how well each model works. Random Forest had the best accuracy of all the models (96.86%), which shows that it is quite good at making predictions in general. Its precision (85.18%) and recall (84.43%) are likewise well balanced, which gives it an excellent F1 score (84.80%). This means that there is a good balance between precision and recall. Logistic Regression did almost as well, with an accuracy of 96.73%, a precision of 83.64%, and the best recall (86.50%) of all the models. Its F1 score (85.05%) was the best overall, which means that Logistic Regression was a little better at correctly identifying positive cases while still being precise. The Decision Tree model also did well, with an accuracy of 94.97% and a recall of 85.57%. This means that it is good at finding most positive cases. But its accuracy (75.86%) and F1 score (80.42%) were a little lower, which means it had more false positives than ensemble and regression models. The SVM model did well overall, but it wasn't quite as well as the others. Its accuracy was 95.60%, and its recall was the lowest at 68.79%, which means it missed more positive cases. The F1 score (74.92%) was the lowest, showing that there was an imbalance between precision and recall, even if the precision was good (82.25%). Logistic Regression and Random Forest both did better and more evenly than the other models, hence they are the best for this dataset. Decision Tree is still a good choice because it's easier to understand, although SVM might need more adjusting to increase recall. These results show that different models work better for different purposes, which shows how important it is to choose algorithms based on the needs of the application.

4.2 Network Performance Evaluation

A detailed comparison of network performance metrics between a traditional algorithm and four machine learning models is shown in Table 2. The metrics include Packet Delivery Rate (PDR), Packet Loss Rate, Stolen Packet Rate, Throughput, Total Time, Energy Consumption, and Detection Efficiency - all critical indicators of network reliability, security, and efficiency. The traditional routing algorithm performed poorly, delivering only 69 packets out of 500, with a PDR of 13.80%, a high packet loss rate (86.20%), and a stolen packet rate of 85.40%, indicating severe vulnerability to wormhole attacks. Its low detection efficiency (13.91%) and minimal throughput (0.0942 packets/sec) underscore the limitations of static, non-intelligent routing protocols in dynamic and adversarial network conditions.

Table 4.2 Comparison of Traditional Algorithm and Machine Learning Network Performance

Metric	Traditional Algorithm	Decision Tree (DT)	Logistic Regression (LR)	Random Forest (RF)	SVM
Total Packets Sent	500	500	500	500	500
Delivered Packets	69	317	25	362	255
Lost Packets	431	183	475	138	245
Stolen Packets	427	73	474	5	232
Packet Delivery Rate (%)	13.80	63.40	5.00	72.40	51.00
Packet Loss Rate (%)	86.20	36.60	95.00	27.60	49.00
Stolen Packet Rate (%)	85.40	14.60	94.80	1.00	46.40
Throughput (packets/sec)	0.0942	0.7051	0.0144	1.0537	1.1426
Total Time (sec)	732.3187	449.6014	1741.8558	343.5557	223.1769
Total Energy Consumed (J)	838.7423	5473.0596	866.1810	4313.9582	3158.6218
Detection Efficiency (%)	13.91	81.28	5.01	98.64	52.36

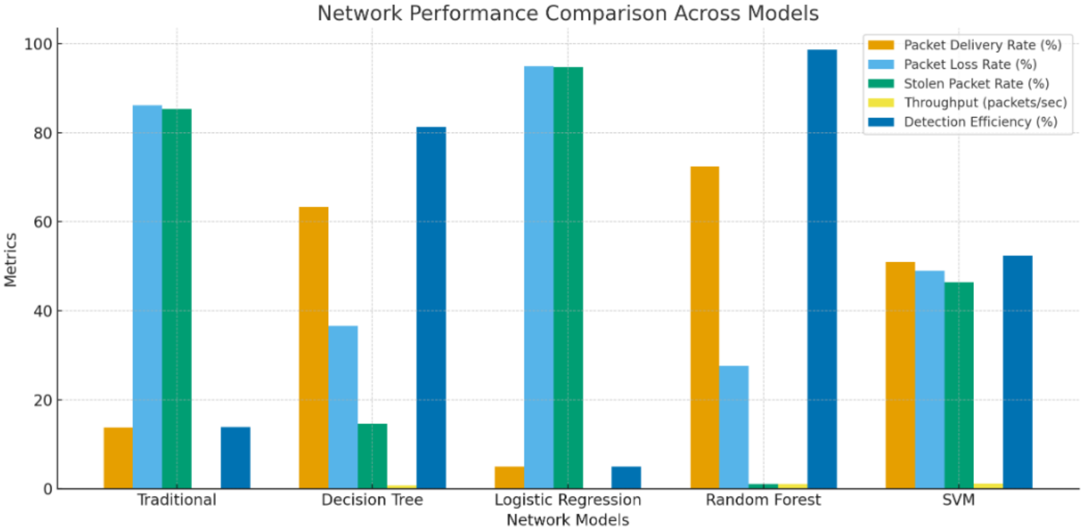

Figure 4 Network Performance comparison

The table above shows how well a classical algorithm and four machine learning-based methods-Decision Tree (DT), Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM)-compare in terms of network performance indicators. It shows how adding ML to a network may greatly improve its performance in several areas, such as packet delivery, loss rates, throughput, energy use, and detection efficiency. The standard technique doesn't work very well; it only sends 69 out of 500 packets, which is a packet delivery rate of 13.80% and a loss rate of 86.20%. Also, it has a high stolen packet rate (85.40%) and a low detection efficiency (13.91%), which shows that its security and network reliability are not very good. It has a very low throughput (0.0942 packets/sec), which shows that packets are not being sent quickly. On the other hand, Random Forest (RF) shows better outcomes on most criteria. It sends 362 packets, which is the highest packet delivery rate (72.40%) and the lowest stolen packet rate (1.00%). It also has a high detection rate (98.64%). The fact that it sends 1.0537 packets per second and takes only 343.55 seconds to send them all shows that it is very fast and safe. The Decision Tree (DT) method also works much better than the old one, with a delivery rate of 63.40% and a detection rate of 81.28%. However, it uses more energy (5473.06 J) than other ML models. SVM has a balanced performance, with a delivery rate of 51.00%, a modest detection efficiency of 52.36%, and the greatest throughput of 1.1426 packets/sec. However, its stolen packet rate of 46.40% is still quite high. On the other hand, Logistic Regression (LR) doesn't work as well because it has the lowest delivery rate (5.00%), a lot of packet loss, and low detection efficiency (5.01%). This means it may not be the best choice for this application. In general, machine learning methods like Random Forest and Decision Tree make networks run far better than the old algorithm. They improve delivery, cut down on losses, and make security stronger. These results show how ML can change the way networks work by making them more reliable and efficient.

4.3 Traffic Management Efficiency Analysis

To get the best performance out of a network, reduce latency, and make sure that the load is evenly distributed, traffic management must be done well. Table 3 shows how three important traffic management indicators compare: Average Hops per Packet, Traffic Balance Index, and Path Utilisation Rate. The classical routing system had a mean hop count of 14.19, which suggested that it didn't do an effective job of optimising traffic. This made routes long and not very useful. The traffic share index (0.58) showed that the demand wasn't equally spread out, and the roadway use rate (105.80%) showed that some routes were being used too much, which caused gridlock and other problems.

Table 4.3 Traffic Management Metrics Comparison Across Classical and ML Models

Metric	Traditional Algorithm	Decision Tree (DT)	Logistic Regression (LR)	Random Forest (RF)	SVM
Average Hops per Packet	14.19	4.95	56.28	3.88	3.96
Traffic Balance Index	0.58	0.37	0.67	0.46	0.38
Path Utilization Rate (%)	105.80	100.32	104.00	100.00	102.35

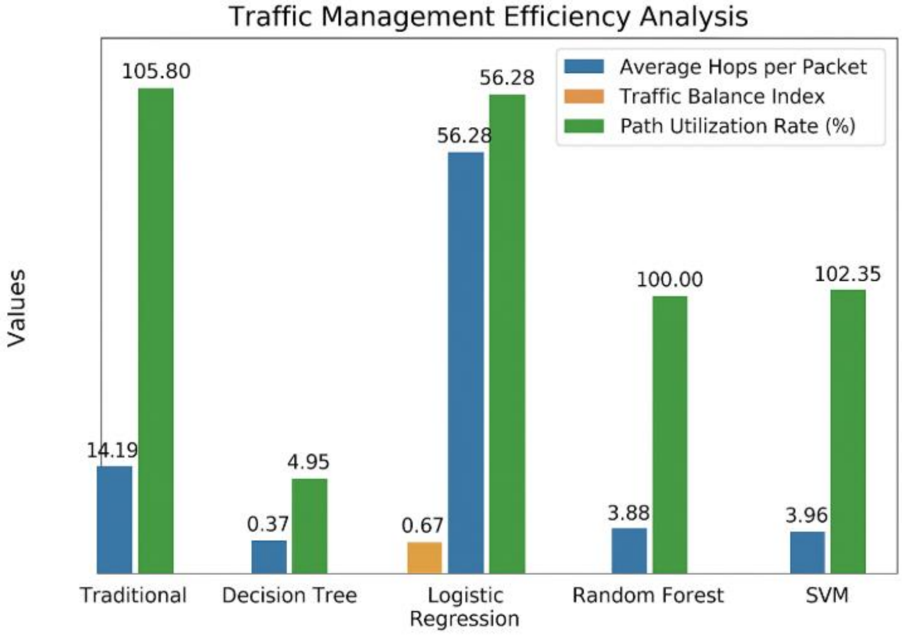

Figure 5 Traffic Management Efficiency Analysis

The table above compares how well a conventional algorithm or four machine learning models—Decision Tree (DT), logarithm regression (LR), randomised forest (RF), and supported vector machine (SVM)—handle traffic. The Average Hops per Packet, Traffic Optimisation Index, or Path Utilisation Rate (%) are all good ways to tell how well a network regulates data traffic and optimises routing. The old method isn't as good at handling traffic because it has an elevated mean number of hops per packet (14.19), which implies that packets take longer trips to reach to their destinations. The traffic balance index (0.58) shows that the load is evenly spread out, and the path use rate (105.80%) suggests that some network paths get utilised too much, which could cause congestion and decrease performance. Methods based on machine learning make a major difference in how traffic is handled. Overall, random forest algorithm (RF) does the greatest job because it has the fewest standard hops per packet (3.88) This means that routing is very efficient and there is very little delay in sending packets. The path utilisation rate of 100.00% shows that the paths are being used evenly, and the traffic balance index of 0.46 shows that the traffic is being spread out quite evenly. SVM and Decision Tree (DT) are not far behind, with average hops of 3.96 and 4.95, respectively. Both are far more efficient than the old approach. Their path utilisation rates (102.35% and 100.32%) are still close to the best they can be, which shows that they are choosing the right paths and using their resources well. Logistic Regression (LR), on the other hand, doesn't work well for traffic management. It has a very high average hops per packet (56.28) and traffic balancing index (0.67), which means that routing is not working well and traffic is not flowing evenly, even though the path utilisation rate is good (104.00%). Overall, ML-based methods, especially Random Forest and SVM, make traffic management a lot better by lowering hop counts, making better use of paths, and optimising load distribution. This makes network communication faster and more efficient than the old way.

4.4 Attack Detection Performance of proposed Approach

Table 4 summarizes the intrusion detection performance of the four machine learning models based on metrics such as Detected Attacks, Missed Attacks, False Positives, Precision, Recall, and F1 Score. Effective attack detection is critical for ensuring network security and reliable communication, especially in scenarios involving wormhole or similar routing attacks. Random Forest again exhibited the best performance, detecting 364 attacks and missing only 5, achieving the highest recall (98.64%), indicating excellent sensitivity and coverage. However, its precision (30.16%) and F1 score (46.19%) suggest a moderate false positive rate, which, while acceptable in security-sensitive applications, indicates potential for further improvement through model refinement or ensemble techniques. The Decision Tree model detected 319 attacks with a recall of 81.38%, reflecting strong detection capability. However, its lower precision (25.60%) and F1 score (38.95%) resulted from a high number of false positives (927), suggesting that while it covers a broad range of threats, it also misclassifies benign traffic more frequently.

Table 4.4 Intrusion Detection Performance Comparison of Four Machine Learning Models

Metric	Decision Tree (DT)	Logistic Regression (LR)	Random Forest (RF)	SVM
Detected Attacks	319	26	364	261
Missed Attacks	73	474	5	232
False Positives	927	0	843	1800
Precision (%)	25.60	100.00	30.16	12.66
Recall (%)	81.38	5.20	98.64	52.94

F1 Score (%)	38.95	9.89	46.19	20.44

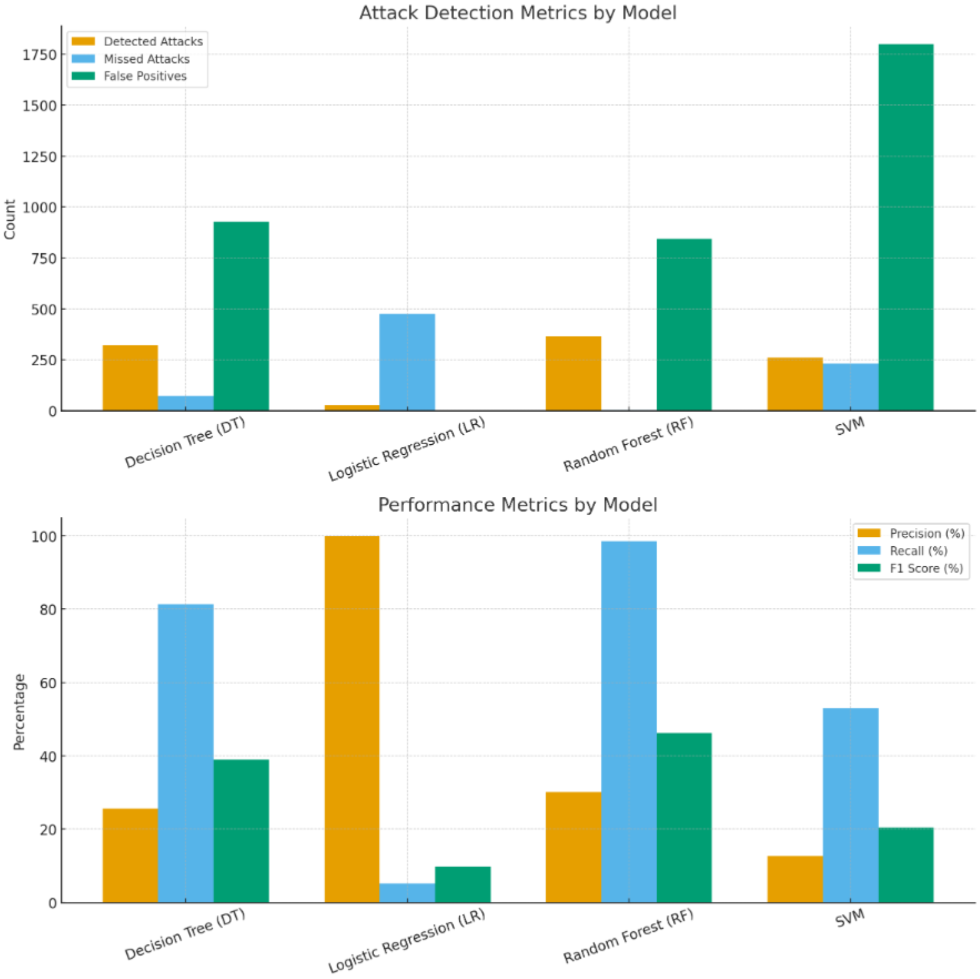

Figure 6 Intrusion Detection Performance Comparison

The table above shows how well four computational models—Decision Tree (DT), logarithmic correlation (LR), Random Forests (RF), or Support Vector Machines (SVM)—found attacks, missed attacks, gave false positives, recall, accuracy, or F1 score. These indicators are very important for figuring out how well breach detection works and how well each model finds bad things happening on a network. Random Forest (RF) is the best model overall; it found 364 attacks or missed only 5. It also has the highest recall (98.64%), which means it is very good at finding threats. But its accuracy (30.16%) and F1 score (46.19%) show that it has a moderate false-positive rate, which means that it finds most attacks but sometimes marks routine traffic as malicious. The Decision Tree (DT) model also does a good job, finding 319 attacks with a recall rate of 81.38%, which shows that it is good at finding things. But its accuracy (25.60%) and F1 score (38.95%) are not very high because it has a lot of false positives (927), which means it covers a wider range of threats. Logistic Regression (LR) has the lowest F1 score (9.89%) since it only detects 26 attacks and has a very low recall (5.20%). It does get perfect precision (100%) with no false positives. This means that it is very accurate when it does predict an assault, but it misses most of them. SVM works well, but not as well as it might. It found 261 attacks with a recall rate of 52.94%, but a poor precision rate of 12.66% and an F1 score of 20.44%. This shows that it had a lot of false positives (1800). Overall, Random Forest strikes the optimal balance between detection skill and dependability. Decision Tree, on the other hand, has high detection but more false alarms. Logistic Regression is not a good choice because it has poor detection coverage, even though it is very accurate.

5. Conclusion

This study introduces an optimised method for route selection in network traffic management, incorporating machine learning approaches to improve efficiency, security, and scalability. A bespoke dataset was created to replicate both benign and malicious network conditions, facilitating the training and assessment of four machine learning models: Decision Tree, Logistic Regression, Random Forest, and Support Vector Machine. Of these, Random Forest has consistently performed the best, with the highest accuracy (96.86%), identifying efficiency (98.64%), or the lowest rate of stolen packets (1.00%). It also made sure that traffic flowed smoothly with the least number of hops on average, that paths were used evenly, and that energy use was kept to a minimum, showing that it was a good fit for real-world dynamic networks. The proposed methodology addresses significant shortcomings of existing network management systems, including insufficient security, low accuracy, prolonged processing times, and restricted scalability. The system effectively finds hostile pathways or dynamically optimises routing by using artificial intelligence-based attack detection in route selection. This improves packet delivery and cuts down on losses. A comparison analysis with traditional routing algorithms or alternative machine learning models demonstrates that the proposed methodology significantly enhances network performance and managing traffic efficiency. This study confirms that machine learning-driven route optimisation, particularly through the use of Random Forest, provides a dependable, secure, and energy-efficient method for modern network traffic management The framework is adaptable for practical applications in wireless and mobile networks, facilitating intelligent decision-making in dynamic environments. Future endeavours may concentrate on augmenting model accuracy, including ensemble methodologies, and assessing performance in extensive network contexts to guarantee wider application and robustness against new network vulnerabilities.

References

[1] B. Kumari, “An Intelligent Routing Frame Work for High Traffic Networks using Deep Learning,” *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 14, no. 02, 2025, doi: 10.15680/ijirset.2025.1402001.

[2] F. Nocua M, W. J. Pérez-Holguín, and C. Pardo-Beainy, “Urban traffic monitoring based on deep learning on an embedded GPU,” *Expert Syst. Appl.*, vol. 273, no. February, 2025, doi: 10.1016/j.eswa.2025.126847.

[3] Z. Liu, X. Li, Z. Lu, and X. Meng, “IWOA-RNN: An improved whale optimization algorithm with recurrent neural networks for traffic flow prediction,” *Alexandria Eng. J.*, vol. 117, no. December 2024, pp. 563–576, 2025, doi: 10.1016/j.aej.2024.12.074.

[4] M. Yaqub, S. Ahmad, M. A. Manan, M. S. Pathan, and L. He, “Predicting traffic flow with federated learning and graph neural with asynchronous computations network,”

Array, vol. 26, no. July 2024, 2025, doi: 10.1016/j.array.2025.100411.

[5] A. Louati, "Machine learning framework for sustainable traffic management and safety in AlKharj city," *Sustain. Futur.*, vol. 9, no. November 2024, p. 100407, 2025, doi: 10.1016/j.sftr.2024.100407.

[6] Y. A. Pan, F. Li, A. Li, Z. Niu, and Z. Liu, "Urban intersection traffic flow prediction: A physics-guided stepwise framework utilizing spatio-temporal graph neural network algorithms," *Multimodal Transp.*, vol. 4, no. 2, p. 100207, 2025, doi: 10.1016/j.multra.2025.100207.

[7] B. Bamdad Mehrabani, J. Erdmann, L. Sgambi, S. Seyedabrishami, and M. Snelder, "A multiclass simulation-based dynamic traffic assignment model for mixed traffic flow of connected and autonomous vehicles and human-driven vehicles," *Transp. A Transp. Sci.*, vol. 21, no. 2, pp. 1–32, 2025, doi: 10.1080/23249935.2023.2257805.

[8] A. Li, T. Bai, Y. Chen, C. G. Cassandras, and A. A. Malikopoulos, "A cooperative compliance control framework for socially optimal mixed traffic routing," *arXiv Prepr. arXiv2503.22837*, 2025.

[9] O. Aouedi, V. A. Le, K. Piamrat, and J. I. Yusheng, "Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions," *ACM Comput. Surv.*, vol. 57, no. 6, 2025, doi: 10.1145/3703447.

[10] C. Marketing, C. Induce, M. Buying, M. Profitable, C. A. Field, and A. Scott, "Alma Mater Studiorum Università di Bologna Archivio istituzionale della ricerca Rights / License : The terms and conditions for the reuse of this version of the manuscript are specified in the," no. May, 2025.

[11] Y. Deng, "A Reinforcement Learning Approach to Traffic Scheduling in Complex Data Center Topologies," vol. 4, no. 3, 2025.

[12] U. Srilakshmi, N. Veeraiah, Y. Alotaibi, S. A. Alghamdi, O. I. Khalaf, and B. V. Subbayamma, "An improved hybrid secure multipath routing protocol for MANET," *IEEE Access*, vol. 9, pp. 163043–163053, 2021, doi: 10.1109/ACCESS.2021.3133882.

[13] J. Tang, G. Liu, and Q. Pan, "A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends," *IEEE/CAA J. Autom. Sin.*, vol. 8, no. 10, pp. 1627–1643, 2021, doi: 10.1109/JAS.2021.1004129.

[14] K. Guo *et al.*, "Optimized Graph Convolution Recurrent Neural Network for Traffic Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1138–1149, 2021, doi: 10.1109/TITS.2019.2963722.

[15] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," *IEEE Wirel. Commun.*, vol. 26, no. 1, pp. 87–93, 2019, doi: 10.1109/MWC.2019.1700441.

[16] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine Learning for Networking: Workflow, Advances," pp. 1–8, 2017.

[17] Y. Ren, M. Wei, H. Xin, T. Yang, and Y. Qi, "Distributed Network Traffic Scheduling via Trust-Constrained Policy Learning Mechanisms," no. 4, 2025.

[18] Á. Moraga , J. de Curtò, I. de Zarzà, and C. T. Calafate, "AI-Driven UAV and IoT Traffic

Optimization: Large Language Models for Congestion and Emission Reduction in Smart Cities," *Drones*, vol. 9, no. 4, pp. 1–28, 2025, doi: 10.3390/drones9040248.

[19] C. Gheorghe and A. Soica, "Revolutionizing Urban Mobility: A Systematic Review of AI, IoT, and Predictive Analytics in Adaptive Traffic Control Systems for Road Networks," *Electron.*, vol. 14, no. 4, pp. 1–25, 2025, doi: 10.3390/electronics14040719.

[20] E. Ji, Y. Wang, S. Xing, and J. Jin, "Hierarchical Reinforcement Learning for Energy-Efficient API Traffic Optimization in Large-Scale Advertising Systems," *IEEE Access*, vol. 13, no. July, pp. 142493–142516, 2025, doi: 10.1109/ACCESS.2025.3598712.

[21] H. R. Bonikela and A. Renuka, "Full-Stack Challenges in Air Traffic Management Systems," no. August, 2025.

[22] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, "One sketch to rule them all: Rethinking network flow monitoring with UnivMon," *SIGCOMM 2016 - Proc. 2016 ACM Conf. Spec. Interes. Gr. Data Commun.*, no. Question 24, pp. 101–114, 2016, doi: 10.1145/2934872.2934906.

[23] J. Auld, M. Hope, H. Ley, V. Sokolov, B. Xu, and K. Zhang, "POLARIS: desarrollo e implementación del marco de modelado basado en agentes para la demanda de viajes integrada y simulaciones de redes y operaciones," p. 23, 2016.

[24] M. Ashkanani, A. AlAjmi, A. Alhayyan, Z. Esmael, M. AlBedaiwi, and M. Nadeem, "A Self-Adaptive Traffic Signal System Integrating Real-Time Vehicle Detection and License Plate Recognition for Enhanced Traffic Management," *Inventions*, vol. 10, no. 1, 2025, doi: 10.3390/inventions10010014.

[25] Y. Zhang, S. Xu, L. Zhang, W. Jiang, S. Alam, and D. Xue, "Short-term multi-step-ahead sector-based traffic flow prediction based on the attention-enhanced graph convolutional LSTM network (AGC-LSTM)," *Neural Comput. Appl.*, vol. 37, no. 20, pp. 14869–14888, 2025, doi: 10.1007/s00521-024-09827-3.

[26] M. Alruwaili, A. Ali, M. Almutairi, A. Alsahyan, and M. Mohamed, "LSTM and ResNet18 for optimized ambulance routing and traffic signal control in emergency situations," *Sci. Rep.*, vol. 15, no. 1, pp. 1–25, 2025, doi: 10.1038/s41598-025-89651-4.

[27] Y. Zhang *et al.*, "Linear attention based spatiotemporal multi graph GCN for traffic flow prediction," *Sci. Rep.*, vol. 15, no. 1, pp. 1–16, 2025, doi: 10.1038/s41598-025-93179-y.

[28] M. Fabris, R. Ceccato, and A. Zanella, "Efficient Sensors Selection for Traffic Flow Monitoring: An Overview of Model-Based Techniques Leveraging Network Observability," *Sensors*, vol. 25, no. 5, pp. 1–21, 2025, doi: 10.3390/s25051416.

[29] J. Jin, S. Xing, E. Ji, and W. Liu, "XGate: Explainable Reinforcement Learning for Transparent and Trustworthy API Traffic Management in IoT Sensor Networks," *Sensors*, vol. 25, no. 7, pp. 1–30, 2025, doi: 10.3390/s25072183.

[30] X. Peng, H. Gao, G. Han, H. Wang, and M. Zhang, "Joint Optimization of Traffic Signal Control and Vehicle Routing in Signalized Road Networks using Multi-Agent Deep Reinforcement Learning," 2023.

[31] N. V. Karadimas, N. Doukas, M. Kolokathi, and G. Defteraiou, "Routing optimization

heuristics algorithms for urban solid waste transportation management," *WSEAS Trans. Comput.*, vol. 7, no. 12, pp. 2022–2031, 2008.

[32] P. Chakroborty and T. Wivedi, "Optimal route network design for transit systems using genetic algorithms," *Eng. Optim.*, vol. 34, no. 1, pp. 83–100, 2002, doi: 10.1080/03052150210909.

[33] M. Liu, Y. Li, X. Liu, Y. Chen, and R. Hao, "An Integrated Optimization Framework for Connected and Automated Vehicles and Traffic Signals in Urban Networks," *Systems*, vol. 13, no. 4, pp. 1–25, 2025, doi: 10.3390/systems13040224.

[34] Z. Lu, K. Wu, E. Bai, and Z. Li, "Optimization of Multi-Vehicle Cold Chain Logistics Distribution Paths Considering Traffic Congestion," *Symmetry (Basel).*, vol. 17, no. 1, 2025, doi: 10.3390/sym17010089.

[35] M. Jakubec, M. Cingel, E. Lieskovská, and M. Drliciak, "Integrating Neural Networks for Automated Video Analysis of Traffic Flow Routing and Composition at Intersections," *Sustain.*, vol. 17, no. 5, pp. 1–18, 2025, doi: 10.3390/su17052150.

[36] A. Dutta, L. M. Samaniego Campoverde, M. Tropea, and F. De Rango, *A Comprehensive Review of Recent Developments in VANET for Traffic, Safety & Remote Monitoring Applications*, vol. 32, no. 4. Springer US, 2024. doi: 10.1007/s10922-024-09853-5.

[37] Z. Grujic and B. Grujic, "Optimal Routing in Urban Road Networks: A Graph-Based Approach Using Dijkstra's Algorithm," *Appl. Sci.*, vol. 15, no. 8, 2025, doi: 10.3390/app15084162.

[38] R. An, X. Zhang, M. Sun, and G. Wang, "GC-YOLOv9: Innovative smart city traffic monitoring solution," *Alexandria Eng. J.*, vol. 106, no. July, pp. 277–287, 2024, doi: 10.1016/j.aej.2024.07.004.