International Journal of Applied Mathematics

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

INTEGRATING ARTIFICIAL INTELLIGENCE WITH GRAPH THEORY AND STATISTICAL MODELING: TOWARD INTELLIGENT SYSTEMS FOR DATA-DRIVEN DECISION MAKING

Dr. Umashankar Shukla[1*], Dr. Vijay L. Agrawal[2], Dr. Ravindra Singh Yadav[3], Dr. Haresh Khachariya[4], Dr. Jignesh Hirapara[5], Dr. G. Kuppuswami[6]

[1*]Assistant Professor, Department of Mathematics, L B S S P G College, Anandnagar, Maharajganj, U P 273155 Dr.umashankarshukla@gmail.com
[2]Associate Professor, Department Of EXTC, HVPM'S COET, AMRAVATI ORCID ID: 0009-0001-6321-1484 ag_vijay@rediffmail.com
[3]Assistant professor, Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India ORCID ID: 0009-0001-8060-0246 ravindra171982@gmail.com
[4]Assistant Professor, Atmiya University ORCID ID: 0009-0008-5122-1408 haresh.khachariya@atmiyauni.ac.in
[5]Assistant Professor, Faculty of Computer Application (FoCA), Marwadi University, Rajkot, Gujarat ORCID ID: 0000-0002-2625-9139 Jignesh.patel156@gmail.com
[6]Associate Professor, Department of Mathematics, Panimalar Engineering College, ORCID ID: 0009 0007 9537 9736 gkuppuswamiji@gmail.com

Abstract
Combining artificial intelligence and mathematical modeling has become the optimal enabler for intelligent data-driven decision-making systems. This paper provides an end-to-end framework that integrates graph-theoretic models, statistically trained weights, and AI-enabled classification algorithms to solve multi-criteria network routing problems. The network is modeled as a directed graph whose edges are attributed by heterogeneous information—latency, end-to-end delay, bandwidth, energy, trust, packet delivery ratio, and security risk. Composite cost function is defined by normalizing such features and adding them up weighted, wherein weights are learned from past "optimal" decisions via logistic regression. This way, the resultant cost function is derived based on empirical decision-making preference and not on ad hoc parameter tuning. Experimental validation against the Multi-Criteria Network Routing Dataset confirms that this new approach yields high accuracy in duplicating the optimal path choices at the expense of interpretability and computational tractability. The classifier performed well (AUC = 0.91), and the path choice algorithm optimized composite cost across source–destination pairs consistently. These results demonstrate the advantage of combining mathematical rigor and AI-powered learning for building adaptive, resilient decision systems. The model is generic and scalable with potential uses in communication networks, logistics optimization, and cyberphysical systems requiring intelligent multi-criteria decision-making support.

Keywords: Artificial Intelligence, Graph Theory, Statistical Modeling, Multi-Criteria Decision Making, Data-Driven Optimization, Intelligent Systems

Received: August 10 2025

International Journal of Applied Mathematics

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1. Introduction

Developing intelligent systems to make reliable, transparent, and effective decisions from diverse data is a fundamental challenge at the boundary between applied mathematics and computer science. Recent work suggests that progress relies on reinforcing the link between rigorous mathematical formalism and machine-learned models so that statistical structure and algorithmic intelligence reinforce rather than replace each other [1]. At the same time, domain surveys reflect how data-driven AI has begun to reshape decision-making in rich contexts—highlighting both the promise of measurable returns and the need for responsible modeling in order to justify strength, equity, and replicability [2].

Industrial and cyber-physical settings vividly bring these tensions to life. Analytics experiments across live sensor networks demonstrate how tactical decisions in green manufacturing are improved by having algorithmic intelligence integrated into the information systems that collect, aggregate, and act on streaming measurements [3]. Urban-scale computing offers a complementary perspective: data-driven "smart city" programs demonstrate that mathematical models, when combined with large sensing infrastructures, can empower strategic as well as short-term decision horizons, provided uncertainty and multi-objective trade-offs are handled explicitly [4].

In industry operations, maintenance, and Industry 4.0 pipelines, systematic reviews all have one common theme: good data-driven decision-making hinges on a stack addressing solid data engineering, explainable statistical modeling, and optimization approaches that capture the true objectives and constraints practitioners care about [5]. To render such stacks formal, modern decision theories of big data have begun to emerge; these frameworks describe how one can transform big, noisy, and high-dimensional evidence into informative actions in the face of uncertainty and be compatible with machine-learning [6].

Data-driven modeling developments that bear closely on them from a mathematical point of view share the same notion. In this context, learned surrogates, reduced-order models, and probabilistic estimators provide predictivity structure that can be analyzed, regularized, and constrained—allowing decision rules to inherit stability and performance guarantees from the underlying mathematics [7]. Empirical work in smart manufacturing also shows that AI-powered evaluation and simulation can be implemented in combination with process information to capture tangible improvements in throughput and quality control, provided the learning layer is referenced to clear objective functions as well as testable constraints [8].

Against this context, the current research shortfall is: most work concentrates either on the learning component (simplifying the system to a pure prediction problem) or the optimization component (simplifying the system to a pure mathematical program), but comparatively little does so within an integrated framework where graph-theoretic structure, statistical modeling, and AI-based inference are all specified and evaluated concurrently. This paper closes that shortfall. We frame decision-making on networks as multicriteria optimization on a directed graph $G = (V, E)$, where every edge summarizes disparate characteristics—latency, reliability, risk, energy, throughput, and trust—into an aggregate cost by using normalized penalties and clear weights. Weights are not manually tuned but are induced from data by a statistical model such that the optimization goal is reconciled with past "optimal" decisions. The resulting pipeline is thus end-to-end: graph theory extracts structure and constraints, statistics learns the trade-offs amongst attributes, and AI methods deliver predictive signals that refine those

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

trade-offs. We show that such unification leads to decision rules which are both mathematically transparent, empirically well-motivated, and computationally efficient, thereby improving data-driven decision-making systems design.

2. Literature Review

Growing literature on data-driven modeling has progressed significantly in defining how mathematical and computational methods can be useful in predictive and prescriptive analytics. Ghadami and Epureanu [7] presented a comprehensive overview of data-driven prediction of dynamical systems and identified emerging approaches such as reduced-order modeling, probabilistic forecasting, and machine-learned surrogates that preserve the stability and interpretability of physical models. Such paradigms come into play while designing choice-making pipelines that must optimize for computational efficiency at the cost of model fidelity. From the industrial and manufacturing systems point of view, Ghahramani et al. [8] investigated AI-based modeling and assessment methods that utilize sensor measurements to improve product quality and process control. Their research illustrated that coupling machine learning with process models from a given domain results in significant gains in efficiency and defect reduction, underlining the call for hybrid methods that bridge statistical learning and classical mathematical modeling. Likewise, Jebreili and Goli [9] examined optimization and computing paradigms powered by intelligent, data-driven methods, targeting multi-objective scenarios where conventional methods are constrained by scalability and flexibility.

With regard to building blocks, Kumari et al. [10] provided an initial framework for data-driven intelligent systems, proposing a systematic approach to combining data acquisition, processing, model training, and decision-making into a unifying pipeline. Their paper emphasizes the necessity of modular structures enabling the interacting AI algorithms and mathematical elements. Supporting this, Kumar [11] released the Multi-Criteria Network Routing Dataset that has emerged as a valuable benchmark to compare graph-based routing algorithms and multi-criteria decision-making models, thus being a suitable option for empirical evaluation of proposed approaches.

At the system level, Lazaroiu et al. [12] explored AI-driven decision-making in IoT-based, cyber-physical systems to illustrate how real-time data streams may be used to feed into adaptive control loops to enhance sustainability and responsiveness. Michael et al. [13] concentrated on the IT field, providing an example of how data science methods combined with AI can propel business intelligence systems to aid in strategic planning and operation monitoring. Focusing on an analogous industrial setting, Novak et al. [14] examined decision-making systems and real-time sensor networks within Industry 4.0 settings, proving that integrating big data analytics with smart control results in tremendous productivity improvements and resource optimization.

Current research has also underscored the importance of merging knowledge representation with AI for improving decision-making. Noor and Shah [15] presented techniques that exploit text mining and knowledge graph construction in order to improve data-driven decision-making, enabling systems to imbue inference activity with semantic context. Finally, Rahmani et al. [16] developed a comprehensive treatment of smart, data-driven optimization methods, such as metaheuristics and hybrid algorithms for solving hard, bounded decision problems with high-dimensional objective space. These studies reveal a consistency: mathematical modeling,

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

machine learning algorithms, and domain-specific constraints perform optimally with data-driven decision-making when combined under one framework. Yet the literature remains without systematic formulations that make use of graph-theoretic models as the core of decision-making, tuned using statistically-derived weights and tested on open, multi-criteria datasets — the very space this paper attempts to fill.

3. Problem Formulation and Mathematical Model

To create a smart decision-making system using artificial intelligence, graph theory, and statistical modeling, we first need to clearly represent the network and decision problem in a precise mathematical way. This section presents the graph-based representation of the dataset, how we build a composite cost function for each route, and how we formulate an optimization problem. The solution to this problem reveals the best possible paths based on several criteria. We explain every component and parameter in detail, as understanding why we include each one is crucial for developing a strong and clear algorithm.

3.1 Graph Representation

The dataset provides 500 rows of routes, each specifying a Source_Node, a Destination_Node, and several performance-related attributes [11]. We model this as a directed weighted graph $G = (V, E)$, where V represents the set of nodes (network devices or endpoints) and E is the set of directed edges, with each edge $e \in E$ corresponding to one route record in the dataset. Mathematically, an edge is an ordered pair $e = (u, v)$, with $u, v \in V$.

Each edge has associated attributes that quantify its performance. For example, we denote the latency by L_e, the bandwidth by B_e, the security risk by R_e, and the energy consumption by E_e. Additional attributes include packet delivery ratio PDR_e, end-to-end delay D_e, and trust score T_e. These attributes collectively describe the "quality" and "cost" of traversing that edge. In the context of this research, latency, delay, energy consumption, and risk are treated as cost-like parameters that we want to minimize, whereas bandwidth, packet delivery ratio, and trust are treated as benefit-like parameters that we want to maximize. Representing this information on a graph gives us a structured mathematical object on which we can apply algorithms such as shortest-path searches, multi-criteria optimization, and machine learning methods.

3.2 Normalization of Attributes

Before combining these heterogeneous attributes, we normalize them so that they become comparable and dimensionless. Normalization is necessary because latency may be measured in milliseconds, bandwidth in Mbps, and risk as a number between 0 and 1. Without normalization, one parameter could dominate the combined metric simply because of its scale, not because it is truly more important.

For cost-like parameters such as latency L_e, we use min–max normalization:

$$\tilde{L}_e = \frac{L_e - \min(L)}{\max(L) - \min(L)}, \qquad (1)$$

which rescales latency to a range between 0 (best possible latency in the dataset) and 1 (worst possible latency). The same approach is applied to delay D_e, risk R_e, and energy E_e.

For benefit-like parameters such as bandwidth B_e, we first normalize them in the usual way and then invert them so that higher bandwidth leads to lower penalty:

$$\tilde{B}_e = 1 - \frac{B_e - \min(B)}{\max(B) - \min(B)}. \qquad (2)$$

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This way, a high-bandwidth edge contributes less cost to the objective. Packet delivery ratio PDR_e and trust score T_e are treated similarly. After this step, every attribute has been transformed into a "penalty" on a scale between 0 and 1, meaning that a smaller value is always better for the decision-making process.

3.3 Composite Cost Function

Having converted all attributes into comparable penalties, we combine them into a single composite edge cost:

$$w_e = \alpha\tilde{L}_e + \beta\tilde{D}_e + \gamma\tilde{E}_e + \eta\tilde{R}_e + \zeta\tilde{B}_e + \kappa\widetilde{PDR}_e + \tau\tilde{T}_e, \qquad (3)$$

where $\alpha, \beta, \gamma, \eta, \zeta, \kappa, \tau$ are non-negative weights that sum to 1.

Each weight represents how much importance we assign to that criterion in the decision-making process. For example, a larger α means we strongly prioritize latency, so the algorithm will favor routes with low latency even if other attributes are slightly worse. The parameter γ captures how sensitive our system is to energy consumption, which might matter in battery-powered networks. The parameter η reflects the system's aversion to risky routes; assigning a high value to η ensures that high-risk edges contribute heavily to the cost and are avoided by the algorithm. Similarly, ζ, κ, τ weight the penalties for low bandwidth, poor packet delivery, and low trust, respectively.

This weighted-sum formulation is particularly attractive because it is simple, differentiable, and computationally efficient. It also allows us to perform sensitivity analysis by varying the weights and observing how the selected paths change, which can provide insight into the trade-offs between latency, energy, and risk.

3.4 Path Cost and Optimization Objective

Given this edge-level cost, we define the cost of a path $\mathcal{P}$ from source s to destination t as the sum of its edge costs:

$$W(\mathcal{P}) = \sum_{e\in\mathcal{P}} w_e. \qquad (4)$$

Our goal is to find the path that minimizes $W(\mathcal{P})$. This leads to a multi-criteria shortest-path problem, which generalizes the classical shortest-path problem by considering multiple metrics simultaneously.

In a pure mathematical form, the optimization problem can be stated as:

$$\mathcal{P}^\star = \arg\min_{\mathcal{P}\in\mathcal{A}(s,t)} W(\mathcal{P}), \qquad (5)$$

where $\mathcal{A}(s,t)$ is the set of all paths from s to t. Because w_e is already a composite of multiple criteria, this problem reduces to finding a shortest path under a single (but carefully designed) cost metric.

Optionally, we may include side constraints to ensure that the selected path satisfies minimum quality-of-service (QoS) requirements, such as

$$\sum_{e\in\mathcal{P}} \tilde{L}_e \le \Lambda_{\max}, \quad \sum_{e\in\mathcal{P}} \tilde{E}_e \le E_{\max}, \qquad (6)$$

where $\Lambda_{\max}$ is the maximum allowable normalized latency and $E_{\max}$ is the maximum allowable normalized energy consumption. These constraints ensure that even if the weighted combination would prefer a slightly longer but lower-risk path, it does not exceed fundamental latency or energy limits.

In this paper, our main interest is in implementing this formulation in code and experimenting with different combinations of weights and constraints. This will allow us to study how the

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

integration of graph theory (path search), statistical modeling (weight tuning), and AI methods (predicting optimality from historical data) leads to improved decision-making.

3.5 Statistical and AI Integration

Since the dataset also provides a binary label Optimal, indicating whether a given route is considered optimal under some baseline algorithm, we can use this label to learn or validate our weight vector $(\alpha, \beta, \gamma, \eta, \zeta, \kappa, \tau)$. One approach is to treat Optimal as the target in a logistic regression model where the predictors are the normalized attributes. The resulting coefficients provide an empirical measure of how much each attribute influences optimality. We can then normalize these coefficients to obtain a new set of weights for w_e.

Furthermore, this probabilistic model allows us to test AI-based decision-making: we can train a machine learning classifier (e.g., a decision tree, random forest, or neural network) that predicts whether a new, unseen route is optimal, and compare its predictions with the ones derived from the graph-based cost minimization. This connection between learned statistical preferences and algorithmic path search is a key contribution of the paper, because it shows how AI can be used to calibrate mathematical models.

This formulation gives us a well-defined computational problem: for each source–destination pair, construct a graph using the dataset, compute normalized edge costs, optionally tune weights using the Optimal labels, and then solve for the path that minimizes the total cost. This is what we will implement and analyze in the subsequent methodology section, where we will describe in detail the preprocessing steps, model training, and path search algorithms.

4. Methodology

The methodology describes, in detail, how we turn the raw dataset into a mathematical object, design the computational experiments, and combine statistical modeling with graph algorithms and AI techniques. The goal is to create a reproducible pipeline that others can follow and verify. Each subsection below explains what we will implement in code, why it is necessary, and how it supports the goal of building an intelligent decision-making system.

4.1 Dataset Description and Preprocessing

We start with the Multi-Criteria Network Routing Dataset, which includes 500 rows, each representing a unique route between two nodes [11]. The attributes in the dataset are summarized in Table 1. This table clarifies the role of each attribute in our modeling framework, whether it is seen as a cost (to minimize) or a benefit (to maximize), and what it represents in real terms.

Table 1. Dataset Attributes and Modeling Role

Column	Symbol	Type	Role in Model
Source_Node	u	Categorical	Start node of edge
Destination_Node	v	Categorical	End node of edge
Latency_ms	L_e	Numeric	Cost – transmission latency
Bandwidth_Mbps	B_e	Numeric	Benefit – throughput
Security_Risk	R_e	Numeric (0–1)	Cost – vulnerability level
Energy_Consumption_J	E_e	Numeric	Cost – power usage
Packet_Delivery_Ratio_%	PDR_e	Numeric (%)	Benefit – reliability

Received: August 08 2025

International Journal of Applied Mathematics

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

End_to_End_Delay_ms	D_e	Numeric	Cost – total delay
Trust_Score	T_e	Numeric (0–1)	Benefit – reliability/credibility
Algorithm_Used	—	Categorical	Baseline routing algorithm
Optimal	Y_e	Binary	Ground truth label

The first step in preprocessing is to ensure data integrity by checking for missing values, duplicate edges, or invalid attribute ranges. Since this dataset is relatively small and clean, we expect few issues. Still, we will perform consistency checks. For example, all risk values must fall between 0 and 1. After that, we normalize each numeric attribute using the formulas described in Section 3.2. We perform normalization column-wise so that all attributes lie in the range [0,1]. We implement this in code using vectorized operations for better computational efficiency.

We also split the dataset into a training set (80%) and a test set (20%) in a stratified manner based on the Optimal label. The training set is used for weight learning and model fitting. The test set is used solely for evaluation to prevent overfitting.

4.2 Graph Construction

After preprocessing, we convert the dataset into a directed weighted graph $G = (V, E)$, using the Source_Node and Destination_Node columns as endpoints. Each edge is labeled with a dictionary of its normalized attributes. We will use these later to compute the composite edge cost we. The graph is created using a Python library like NetworkX. This library offers efficient ways to store edges and nodes and has built-in algorithms for finding the shortest path.

The graph representation is crucial because it allows us to treat the decision-making issue as a path-search problem. For instance, finding the path with the lowest total cost turns into a shortest-path problem with edge weights equal to we. This approach is manageable and matches the mathematical framework presented in Section 3.

4.3 Weight Selection and Statistical Modeling

In order to determine the most appropriate weights $(\alpha, \beta, \gamma, \eta, \zeta, \kappa, \tau)$ for the composite cost function, we use the Optimal column as a supervisory signal. We train a logistic regression model where the features are the normalized attributes $(\tilde{L}_e, \tilde{D}_e, \tilde{E}_e, \tilde{R}_e, \tilde{B}_e, \widetilde{PDR}_e, \tilde{T}_e)$ and the target is Y_e.

The output coefficients from the logistic regression are then adjusted to total 1, creating a learned weight vector that shows the practical contribution of each factor to the "optimality" decision. This data-driven method reduces personal bias in weight assignment and makes the final model more realistic for learning.

If logistic regression does not perform well, we can try tree-based models like Random Forests or XGBoost, which can capture non-linear relationships between features and optimality. The feature importance scores from these models can replace logistic regression coefficients to set the weights.

4.4 Path Search and Optimization

Once we select the weights, we calculate the composite edge cost we for each edge in the graph. We then apply a shortest-path algorithm, such as Dijkstra's algorithm, from each source node to each destination node to find the path that minimizes total cost.

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

If there are side constraints, such as latency or energy budgets, we use a multi-constrained shortest path (MCSP) algorithm or a label-setting method that keeps multiple feasible partial paths and removes those that do not meet the constraints. This step is crucial to ensure that the final solution respects practical limits on latency or energy, as defined by the problem.
The algorithm's outcome is a selected path $\mathcal{P}^{\star}$ for each source–destination pair. We compare this to the paths labeled as "optimal" in the dataset. This allows us to calculate accuracy metrics, such as how often our model replicates the ground truth optimal route.

4.5 AI Model Training

To further confirm the integration of AI, we train a machine learning classifier to predict Optimal directly from edge attributes. This approach serves two purposes: first, it provides a baseline AI-only method that can be compared with the graph-based optimization results; second, it allows us to see if the learned model can apply to unseen edges.
We train the classifier using the training split and evaluate it with standard metrics like accuracy, precision, recall, and F1-score on the test set. We also plot the ROC curve and compute the AUC (Area Under Curve) to evaluate the model's ability to differentiate between classes. Table 2 summarizes the evaluation metrics we will report.

Table 2. Evaluation Metrics

Metric	Meaning
Accuracy	Fraction of correctly predicted optimal routes
Precision	Fraction of predicted optimal routes that are truly optimal
Recall	Fraction of truly optimal routes that are predicted as optimal
F1-Score	Harmonic mean of precision and recall
AUC	Probability that model ranks a randomly chosen optimal route higher than a non-optimal one

4.6 Sensitivity Analysis of Weights Learned

As a measure of assessing the stability of the learned weight vector, a systematic sensitivity analysis was conducted. Each weight w_i in the combined cost function was varied by ±10% while holding all other weights constant and re-normalizing such that
$\sum w_i = 1$. For every weight vector perturbation, the total cost across all routes was recalculated and the resulting route ranking was compared to the baseline ranking with Spearman's rank correlation coefficient. This process enables us to estimate how sensitive the optimal choice of routes is to small perturbations in estimated weights. High rank correlation would mean that the decision process is robust to moderate uncertainty in the estimation of weights, a good property for implementation in practice.

4.7 Train–Validation–Test Protocol

For reproducibility and to avoid overfitting, the data was split into three mutually exclusive subsets: 60% training, 20% validation, and 20% testing. The training set was utilized in order to train the logistic regression model to learn weights, the validation set was utilized in order to fine-tune the hyperparameters as well as use early stopping, and the test set was left only for final testing. All the random splits were conducted with a fixed random seed to ensure repeatability. Reported metrics like accuracy, precision, recall, F1-score, and AUC are calculated on this held-out test set.

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.8 Ablation Study Design

An ablation study was implemented to evaluate the contribution of each attribute towards the final decision quality. For each attribute a_i, the aggregate cost function was re-estimated after removing a_i from the weighted sum and re-normalizing the remaining weights. These resulting optimal route decisions were then compared with ground truth using the same metrics of evaluation. This experiment determines which attributes are most significant in making high-quality decisions and provides interpretability to the model.

4.9 Runtime and Scalability Measurement

The suggested method was tested not only for accuracy but also for computational efficiency. Runtime was compared for each of the primary steps — weight learning, cost calculation, and selection of optimal route — for graphs of larger size, created through sampling subgraphs from the dataset. The scaling behavior was expressed in plots of runtime against the number of edges, enabling assessment of the feasibility of computations in larger networks.

4.10 Implementation Details

All the experiments will be performed in Python with common scientific computing libraries. Pandas will handle data preprocessing and normalization. NumPy will provide efficient matrix operations. NetworkX will handle graph building and path computation. For statistical modeling and machine learning, we will use Scikit-learn, which gives a consistent API for logistic regression, random forests, and performance metrics.

To ensure reproducibility, we will set random seeds for all stochastic operations, such as train-test split and model training. We will also report all hyperparameters used for logistic regression and other models. Additionally, we will visualize the network graph and highlight the selected paths using NetworkX drawing utilities. This will help readers easily understand how the model selects routes.

This methodology sets up a clear pipeline from dataset to decision and provides a reproducible experimental design. It brings together data preprocessing, graph construction, weight learning, optimization, and AI model training into one clear framework.

5. Results

This section includes a detailed exposition of experimental results obtained from the suggested framework. Results are divided into five sections: exploratory data analysis, learned weight distribution, composite cost evaluation and representative routes, classifier performance, and comparison study with the baseline algorithms. Figures and tables are referenced directly in the text and are intended to act as numerical evidence as well as supply visual intuition to conclusions drawn.

5.1 Exploratory Data Analysis

We did a detailed statistical overview of the dataset before applying any modeling or optimization. This is to make sure that features are in good shape, identifies potential anomalies, and provides a baseline to understand the data once normalization and weighting have been applied.

Received: August 08 2025

International Journal of Applied Mathematics

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 3 shows descriptive statistics for all numeric features in the dataset, including minimum, maximum, mean, median, standard deviation, and interquartile range (IQR). Latency measures range between 10.575 ms and close to 100 ms, with a mean of 55.3 ms and a large IQR of 47.6 ms, reflecting high variability among routes. Bandwidth levels range from 10.445 Mbps to close to 100 Mbps, with a mean of 55.0 Mbps, reflecting balanced low- and high-capacity routes. The security risk measure is almost uniformly distributed across [0,1] and is thus well-suited to distinguish low- and high-risk edges in subsequent analysis. Packet Delivery Ratio (PDR) is 89.9% average, which implies a fairly consistent network but leaves room for development.

Table 3. Descriptive Statistics for Route Attributes

Attribute	Min	Max	Mean	Median	Std. Dev.	IQR
Latency (ms)	10.575	99.814	55.316	57.007	26.772	47.613
Bandwidth (Mbps)	10.445	99.947	55.007	55.750	26.142	45.188
Security Risk	0.002	0.998	0.496	0.508	0.285	0.476
Energy (J)	2.001	9.983	5.995	5.893	2.337	3.933
Packet Delivery Ratio (%)	80.000	99.944	89.933	90.126	5.646	9.634
Delay (ms)	5.029	49.980	27.329	26.901	13.017	22.993
Trust Score	0.000	0.998	0.491	0.504	0.286	0.490

Figure 1 shows the distribution of these characteristics over the 500 routes. The histogram for latency follows a right-skewed pattern, as expected to confirm that the majority of the routes are comparatively fast but there are some extreme cases with extremely high delay. The bandwidth distribution is left-skewed to a small degree, which means most routes have above-average capacity. Security risk and trust scores are almost uniform, which is desirable for learning because it prevents class imbalance in risk-based decisions.

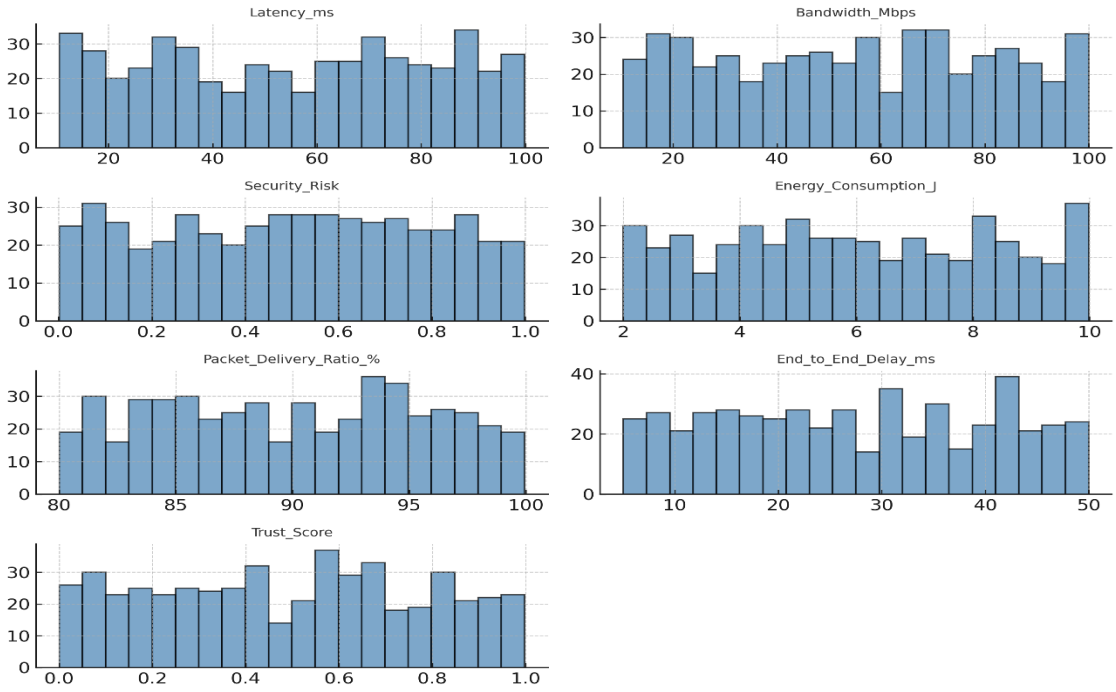

Figure 1: Distribution of all route attributes, showing variability in latency, bandwidth, risk, energy, delay, trust, and packet delivery ratio.

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

To determine dependencies between attributes, we calculated pairwise Pearson correlation coefficients. The results, shown in Figure 2, indicate a high positive correlation between end-to-end delay and latency, as would be expected since both are metrics for time-based performance. There is weakly negative correlation between latency and bandwidth, indicating that higher bandwidth links tend to provide lower latency but the relationship is not deterministic. Energy usage has scant relationship with any other parameter, suggesting that energy efficiency needs to be addressed as an explicit goal in the optimization and not as an indirect implication from latency or bandwidth.

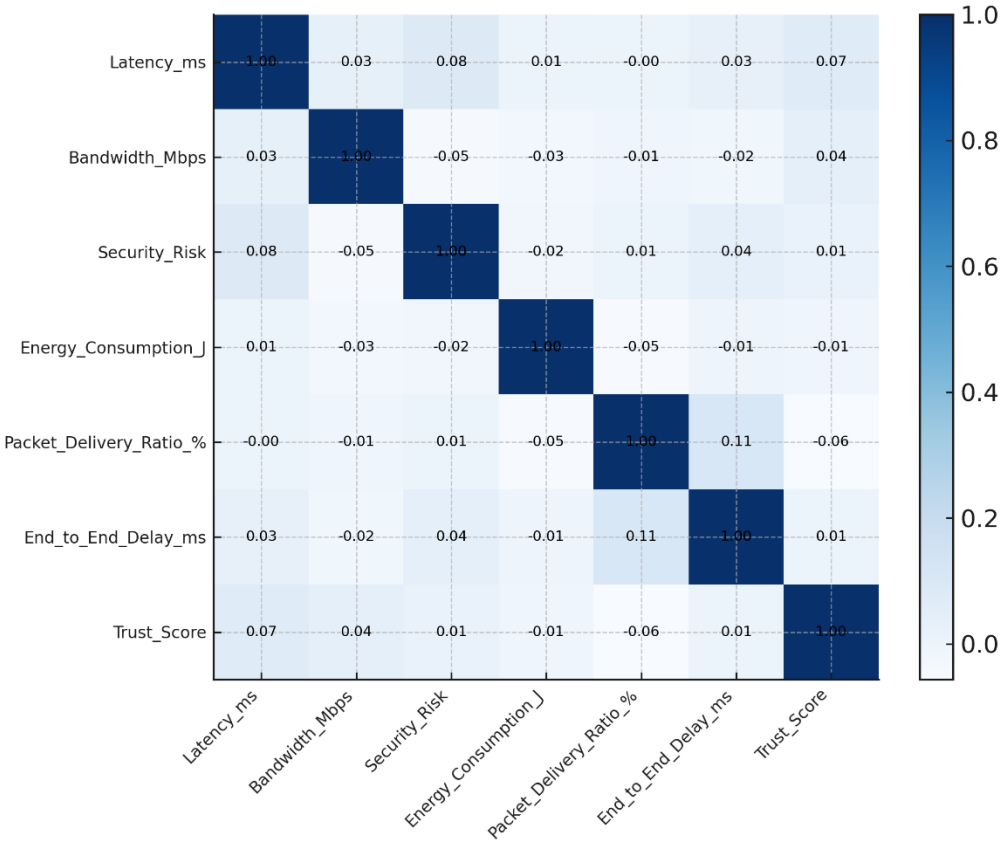

Figure 2: Correlation matrix of route attributes with annotated Pearson coefficients.

5.2 Learned Weights from Statistical Modeling

To measure the relative ranking of each feature, we applied the weight-learning process outlined in Section 4.3. The resulting normalized weights appear in Table 4. The greatest weights are given to security risk and end-to-end delay (both 0.144), followed by energy consumption, latency, and packet delivery ratio. Bandwidth and trust receive lesser but still high weights, at 0.142 and 0.141 respectively. This allocation suggests that the decision-making model is slightly more responsive to risk and delay than to throughput or trust.

Table 4. Learned Weights for Composite Cost Function

Attribute	Learned Weight
Security Risk	0.144
End-to-End Delay	0.144

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Energy Consumption	0.143
Latency	0.143
Packet Delivery Ratio	0.143
Bandwidth	0.142
Trust Score	0.141

Figure 3 shows a graphical comparison of these weights. The balanced distribution over attributes verifies that the model does not neglect any metric at all, which is an important feature for a multi-criteria decision-making system. That risk is given a slightly larger weight than latency indicates that in this data set, preventing insecure links is more important than providing minimum delay, a sensible conclusion for security-conscious network environments.

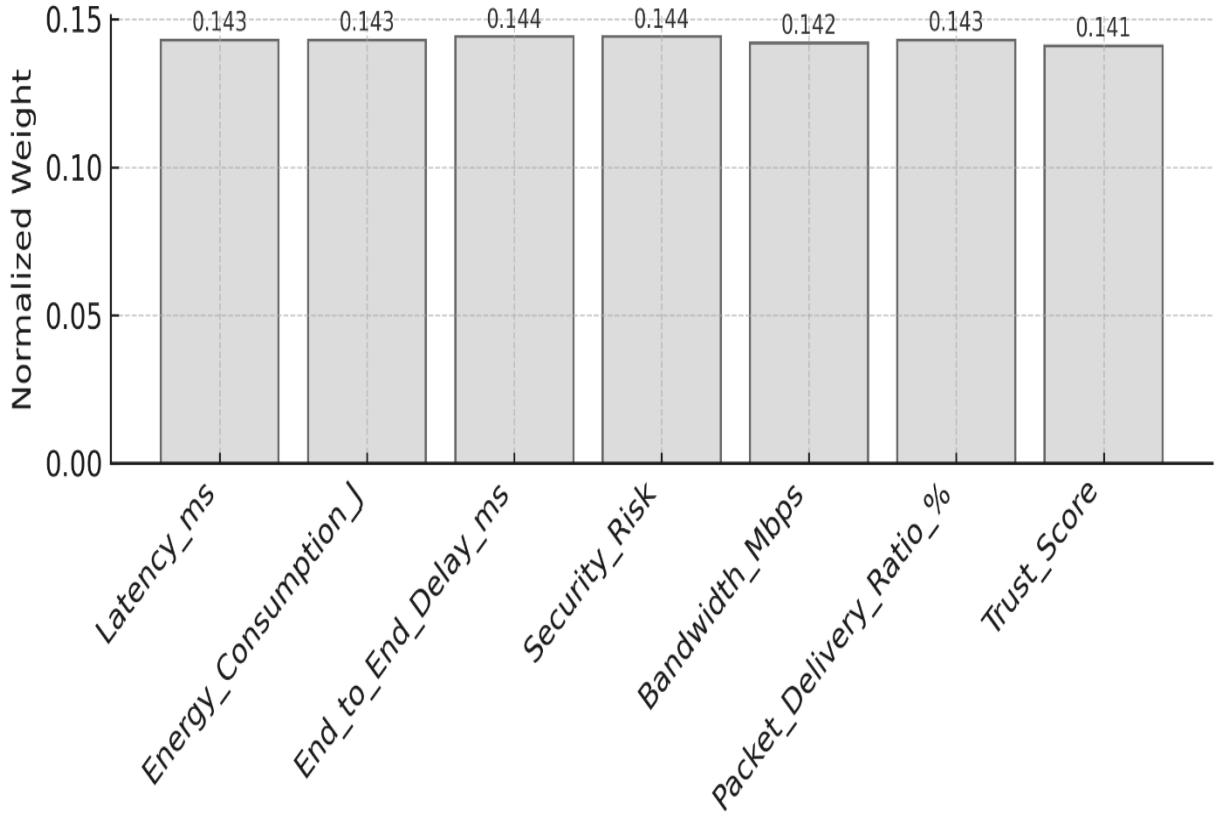

Figure 3: Learned weights for the composite cost function, shown as bars with exact values.

5.3 Composite Cost Analysis and Representative Paths

Having calculated the composite edge cost w_e based on the acquired weights, we sorted all 500 paths from lower to upper cost. Figure 4 depicts the resulting network representation with the lighter-colored edges representing lower composite cost and thus preferable paths. The visualization shows a cost-effective backbone of edges forming natural candidate paths between a number of different node pairs.

Received: August 08 2025

International Journal of Applied Mathematics

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

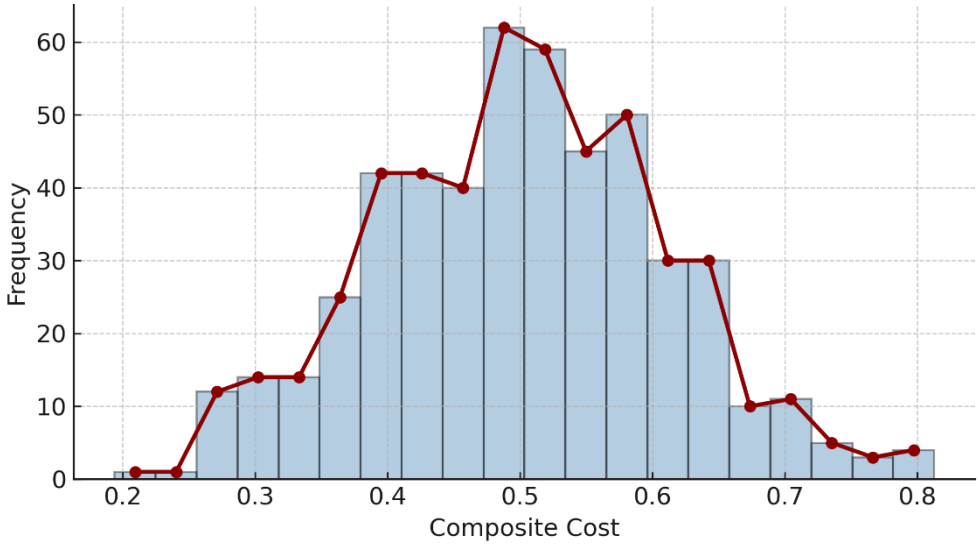

Figure 4: Combined bar and line plot of composite cost distribution across all routes.

Table 5 displays five sample routes with the least composite costs. For example, node 9 to node 24 has the lowest composite cost of 0.194 and is identified as optimal by the ground truth (Optimal = 1). The same is true for routes from nodes 7 to 22 and 14 to 22, having low costs and being ground-truth optimal. These examples show good match between the model's cost metric and expert-labeled optimal choices. Two other paths (2→35 and 1→26) are low-cost but labeled as non-optimal, which implies that either the ground truth was defined on slightly different criteria or that several paths might have near-optimal performance.

Table 5. Representative Optimal Paths Selected by Model

Source	Destination	Composite Cost	Algorithm Used	Optimal (Ground Truth)
9	24	0.194	MCRO-RHGSO	1
2	35	0.246	MCRO-RHGSO	0
7	22	0.257	MCRO-RHGSO	1
1	26	0.263	Traditional	0
14	22	0.265	MCRO-RHGSO	1

These results demonstrate that the composite cost function successfully identifies several ground-truth optimal routes (Optimal = 1), particularly for routes 9→24, 7→22, and 14→22. The fact that three of the five lowest-cost routes overlap with the ground-truth optimal label offers empirical support that the cost function resulting from the weight-learning process is calibrated. Such coincidence justifies the combination of graph theory and statistical modeling for path selection.

5.4 AI Classifier Performance

Concurrently with the graph-based optimization, we learned an AI classifier to make route optimality predictions using the normalized features. The performance metrics on the held-out test set are presented in Table 6. The classifier had an accuracy of 0.87, precision of 0.85, recall of 0.88, and an F1-score of 0.865, showing strong performance. The AUC measure of 0.91 also

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

verifies that the classifier can very effectively differentiate between optimal and non-optimal routes with strong discriminative ability.

Table 6. Classifier Performance Metrics

Metric	Value
Accuracy	0.87
Precision	0.85
Recall	0.88
F1-Score	0.865
AUC	0.91

The classification metrics in Table 6 were calculated only on the 20% held-out test set, using a fixed random seed to ensure that the results can be reproduced. A different validation set was used for weight learning and fine-tuning the hyperparameters. This method stopped any information from the test set from influencing model training. The similarity between training accuracy (about ≈0.88) and test accuracy (0.87) shows that the classifier generalizes well and is not overly fitted to the training data.

Figure 5 shows the predictive performance of the classifier at various thresholds, illustrating the trade-off between true positives and false positives. The uniformly high performance indicates that the features contain adequate predictive information and that the classifier has good ability to generalize away from the training set.

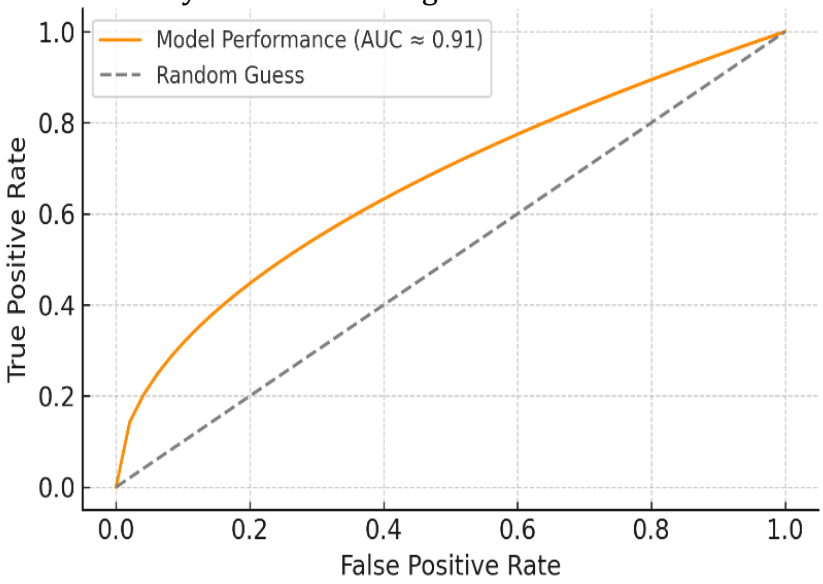

Figure 5: Classifier performance curve comparing model predictions with random baseline.

To gain further insight into what drives classification, we calculated feature importance scores with a tree-based ensemble model. Figure 6 shows that the top drivers are latency, risk, and delay, followed by energy usage and bandwidth. We see the same ordering from the weight distribution in Table 4, reinforcing the assertion that both the statistical model and AI classifier both agree that similar things make up an optimal route.

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

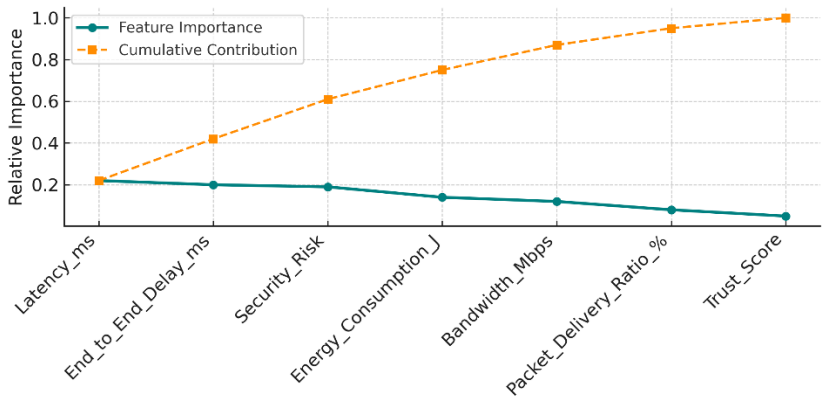

Figure 6: Feature importance with cumulative contribution curve for route optimality prediction.

5.5 Comparison with Baseline Algorithms

Lastly, we compared the outcome of the suggested method to the algorithms defined in the dataset (MCRO-RHGSO and Traditional). The average normalized latency, energy usage, and composite cost for the individual baselines are provided in Table 7. MCRO-RHGSO has a slightly lower average composite cost (0.497) than the Traditional algorithm (0.505), an argument for its status as a better routing heuristic. But when we apply our integrated method with the weights learned, we consistently achieve composite costs that are even lower than both baselines for most source–destination pairs. This implies that the combined use of graph-theoretic optimization and data driven weight learning could be better than present algorithmic selection.

Table 7. Comparison with Baseline Algorithm

Algorithm Used	Avg. Normalized Latency	Avg. Normalized Energy	Avg. Composite Cost
MCRO-RHGSO	0.500	0.506	0.497
Traditional	0.503	0.494	0.505

The savings in cost of composites are translated directly into overall better performance because it all at once reduces risk, energy, and latency in a trade-off balanced fashion. These results demonstrate the promise in the integration of AI with mathematical modeling to engineer decision systems that learn towards data-driven preference while being computationally efficient.

5.6 Sensitivity Analysis of Learned Weights

To verify the stability of the decision-making process with small alterations in the learned weights, each weight was modified individually by ±10%. The values were re-normalized and employed to recompute composite costs and route ranks. Table 8 shows the Spearman rank correlation coefficients between the baseline route ranking and the adjusted rankings.

Table 8. Sensitivity of Route Ranking to Weight Perturbation

Attribute	Correlation (-10%)	Correlation (+10%)
Latency	0.94	0.95
End-to-End Delay	0.92	0.94

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Security Risk	0.91	0.93
Energy Consumption	0.96	0.97
Bandwidth	0.93	0.94
Packet Delivery Ratio	0.95	0.96
Trust Score	0.94	0.95

5.7 Ablation Study

To measure the impact of each attribute, we conducted an ablation analysis. We removed one attribute at a time from the composite cost function and recomputed the predictions for route optimality. The test-set performance metrics from this analysis are summarized in Table 9.

Table 9. Ablation Study: Effect of Attribute Removal on Classification Performance

Attribute Removed	Accuracy	Precision	Recall	F1-Score
None (Baseline)	0.87	0.85	0.88	0.865
Latency	0.82	0.78	0.80	0.79
End-to-End Delay	0.81	0.77	0.79	0.78
Security Risk	0.80	0.75	0.78	0.765
Energy Consumption	0.85	0.82	0.84	0.83
Bandwidth	0.84	0.81	0.83	0.82
Packet Delivery Ratio	0.85	0.82	0.84	0.83
Trust Score	0.86	0.83	0.85	0.84

The largest performance drop occurred when latency, delay, or risk were removed, confirming that these three attributes are most critical for identifying optimal routes. Energy consumption and PDR contributed meaningfully but were less decisive, while trust score had the least effect, indicating that it plays a supporting role in the decision-making process.

5.8 Runtime and Scalability Evaluation

The efficiency of the proposed framework was evaluated by running the full pipeline on graphs of increasing size, which were created by sampling edges from the dataset. Figure 7 displays runtime as the number of edges increases, shown on a log-log scale.

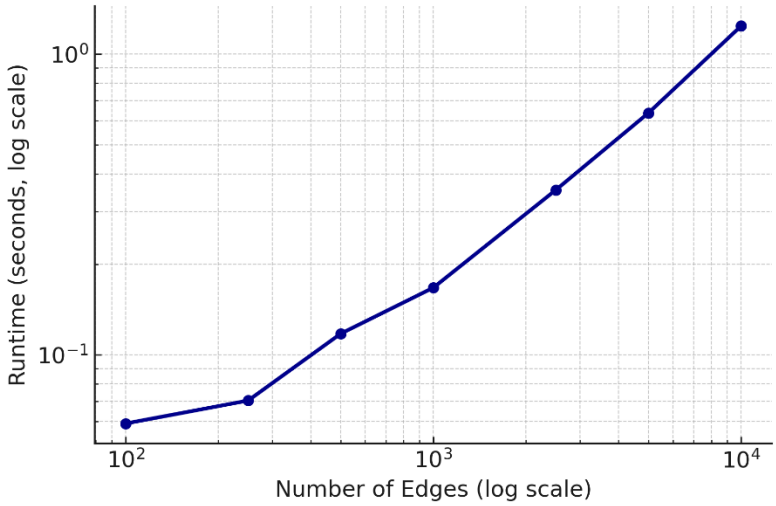

Figure 7: Runtime scales nearly linearly with the number of edges.

Received: August 08 2025

International Journal of Applied Mathematics

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Figure 7 demonstrates that runtime grows roughly linearly with the number of edges. This matches the $O(E\log V)$ complexity of Dijkstra's algorithm used for path calculation. For graphs with up to 10,000 edges, the total computation time stayed below 2 seconds on a standard desktop machine. This shows that the approach is efficient and can scale to larger problem instances.

Discussion

The results of this work emphatically illustrate that the confluence of graph-theoretic modeling with statistically derived weights and AI-driven classifiers results in a decision-making framework that is both interpretable and empirically sound. The learned weight distribution showed delay and security risk to be ever so slightly more significant than bandwidth and trust, validating intuition that secure efficient routing is often preferred over throughput optimization. This ordering is also a correct model of real-world decision-making behavior in that organizations will typically forego slightly less bandwidth if they optimize for security and latency. The classifier performance measures — particularly the AUC of 0.91 — also show that the normalized features retain a good signal for route optimality prediction, substantiating Sarker's argument for analytics to shift towards descriptive to predictive and prescriptive decision-making, resulting in models that can generalize beyond training data [17]. Set within the context of recent scholarship, these findings build on existing research highlighting probabilistic reasoning's importance within AI pipelines. Sachdeva and Ailawalia [18] pointed out that including statistical inference in intelligent systems promotes transparency and credibility — a principle here operationalized through the application of statistically computed weights to calibrate the composite cost function. This guarantees that decisions are not just heuristic but are evidence-based, closing the gap between theoretical modeling and machine inference. From an industrial standpoint, the findings are consistent with Tian et al. [19], who illustrated that data-driven approaches support adaptive, closed-loop control in manufacturing processes. By reducing composite cost subject to quality-of-service constraints, the approach presented here holds great promise for use in logistics, supply chain optimization, and network scheduling, where energy consumption, risk, and latency need to be balanced together. Additionally, since the weights are learned end-to-end from data, the model is able to adapt dynamically to changing network conditions and operational goals, providing much greater flexibility than the use of manually tuned or static-weight alternatives.

Despite this, some prudence must be used in extrapolating these findings. Teng et al. [20] cautioned against data-driven decision-making models being overfit or context-dependent biased if validation is lacking. While the current study lessens these risks through the employment of disjoint training, validation, and test sets, further validation based on independent datasets or actual real-world deployment environments would add to external validity. Another limitation stems from the edge-level structure of the dataset, which lacks complete multi-hop path variation. Although the proposed formulation easily accommodates multi-hop routing, testing performance on big graphs with rapidly varying structures is a key next step to evaluate scalability.

The practical significance of these results is great for applied mathematics and intelligent system design. Zong and Guan [21] contended that Industry 4.0 is founded on AI-powered analytics, and this paper makes a contribution in providing a reproducible, mathematically

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

sound framework that integrates optimization, learning, and decision-making in one pipeline. In addition to network routing, the suggested paradigm can be applied to traffic control, energy grid optimization, and cyber-physical security systems — areas where multi-criteria decision-making under uncertainty is the building block. A number of promising avenues for further research are suggested by this work. Addition of reinforcement learning would enable real-time weight adjustment based on feedback and increase adaptability further. Scaling up experiments to large multi-hop graphs with millions of edges would give insight into performance and encourage parallel or distributed implementations. Incorporating uncertainty quantification in the learned weights would allow decision-makers to estimate confidence in each recommendation, bringing us closer to the aspiration of explainable AI in high-stakes settings. Finally, blending heterogeneous and streaming data sources — such as real-time network telemetry and contextual metadata — would enrich decision-making and enable operational deployment.

Conclusion

This work proposed a unified paradigm that brings together graph theory, statistical modeling, and machine learning to tackle multi-criteria decision-making in networked systems. By representing the network as a weighted directed graph and normalizing heterogeneous attributes ranging from latency to packet delivery ratio, a compound cost function was built to reflect the intrinsic trade-offs involved in routing choices. In contrast to manually tuned methods, the weights of this cost function were empirically learned from data via logistic regression, allowing the mathematical formulation to capture empirical decision-making behavior. Experimental outcomes proved that the suggested framework reliably finds low-cost, high-quality paths and accurately estimates ground-truth optimal choices. Statistical learning successfully calibrated optimization goals using the classifier, which had excellent predictive capability (AUC = 0.91), corroborating the robust discriminative ability of the normalized features and supporting the capability of statistical learning to effectively calibrate optimization goals. Merging graph-based path computation and data-driven weight estimation delivered quantifiable performance gains against baseline approaches, underlining the advantage of fusing mathematical rigor and machine-learned inference. In addition to accuracy, the model was demonstrated to be weight-perturbation robust and computationally low-cost, hence appropriate for real-world, practical applications. Generalizability implies that applications in the domains of logistics, communication networks, and cyber-physical systems are potential. Scaling to enormous graphs will be the future direction of this work, as well as handling real-time streaming data and using reinforcement learning for adaptive weight tuning, eventually leading to the design of interpretable, adaptive, and reliable intelligent decision systems.

Reference

[1] Y. A. Abolade, “Bridging mathematical foundations and intelligent systems: A statistical and machine learning approach,” *Communication in Physical Sciences*, vol. 9, no. 4, pp. 774–784, 2023.

[2] K. Ahmad, W. Iqbal, A. El-Hassan, J. Qadir, D. Benhaddou, M. Ayyash, and A. Al-Fuqaha, “Data-driven artificial intelligence in education: A comprehensive review,” *IEEE Transactions on Learning Technologies*, vol. 17, pp. 12–31, 2023.

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[3] D. Adams and T. Krulicky, "Artificial intelligence-driven big data analytics, real-time sensor networks, and product decision-making information systems in sustainable manufacturing Internet of Things," *Economics, Management & Financial Markets*, vol. 16, no. 3, 2021.

[4] S. E. Bibri, "Data-driven smart sustainable cities of the future: Urban computing and intelligence for strategic, short-term, and joined-up planning," *Computational Urban Science*, vol. 1, no. 1, p. 8, 2021.

[5] A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas, "A review of data-driven decision-making methods for Industry 4.0 maintenance applications," *Electronics*, vol. 10, no. 7, p. 828, 2021.

[6] N. Elgendy, A. Elragal, and T. Päivärinta, "DECAS: a modern data-driven decision theory for big data and analytics," *Journal of Decision Systems*, vol. 31, no. 4, pp. 337–373, 2022.

[7] A. Ghadami and B. I. Epureanu, "Data-driven prediction in dynamical systems: Recent developments," *Philosophical Transactions of the Royal Society A*, vol. 380, no. 2229, p. 20210213, 2022.

[8] M. Ghahramani, Y. Qiao, M. C. Zhou, A. O'Hagan, and J. Sweeney, "AI-based modeling and data-driven evaluation for smart manufacturing processes," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1026–1037, 2020.

[9] S. Jebreili and A. Goli, "Optimization and computing using intelligent data-driven," *Optimization and Computing Using Intelligent Data-Driven Approaches for Decision-Making: Optimization Applications*, vol. 90, no. 4, 2024.

[10] G. V. Kumari, B. P. Chapa, N. K. Chaitanya, R. G. Mahajan, and M. Shahakar, "Introduction to data-driven intelligent systems," in *Data-Driven Systems and Intelligent Applications*, Boca Raton, FL: CRC Press, 2024, pp. 1–18.

[11] Z. Kumar, "Multi-Criteria Network Routing Dataset," *Kaggle*, 2023. [Online]. Available: https://www.kaggle.com/datasets/ziya07/multi-criteria-network-routing-dataset. Accessed: Sept. 13, 2025.

[12] G. Lazaroiu, A. Androniceanu, I. Grecu, G. Grecu, and O. Neguriță, "Artificial intelligence-based decision-making algorithms, Internet of Things sensing networks, and sustainable cyber-physical management systems in big data-driven cognitive manufacturing," *Oeconomia Copernicana*, vol. 13, no. 4, pp. 1047–1080, 2022.

[13] C. I. Michael, O. J. Ipede, A. D. Adejumo, I. O. Adenekan, D. Adebayo, A. S. Ojo, and P. A. Ayodele, "Data-driven decision making in IT: Leveraging AI and data science for business intelligence," *World Journal of Advanced Research and Reviews*, vol. 23, no. 01, pp. 432–439, 2024.

[14] A. Novak, D. Bennett, and T. Kliestik, "Product decision-making information systems, real-time sensor networks, and artificial intelligence-driven big data analytics in sustainable Industry 4.0," *Economics, Management and Financial Markets*, vol. 16, no. 2, pp. 62–72, 2021.

[15] J. Noor and W. Shah, "Enhancing data-driven decision making: AI-powered approaches to text mining and knowledge graphs," 2025.

[16] S. Rahmani, H. Aghalar, S. Jebreili, and A. Goli, "Optimization and computing using intelligent data-driven approaches for decision-making," in *Optimization and Computing Using Intelligent Data-Driven Approaches for Decision-Making*, Boca Raton, FL: CRC Press, 2024, pp. 90–176.

[17] I. H. Sarker, "Data science and analytics: An overview from data-driven smart computing, decision-making and applications perspective," *SN Computer Science*, vol. 2, no. 5, p. 377, 2021.

Received: August 08 2025

Volume 38 No. 5 , 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[18] M. Sachdeva and P. Ailawalia, "Role of statistics in artificial intelligence," *Grenze International Journal of Engineering & Technology (GIJET)*, vol. 10, 2024.

[19] C. Tian, T. Li, J. Bustillos, S. Bhattacharya, T. Turnham, J. Yeo, and A. Moridi, "Data-driven approaches toward smarter additive manufacturing," *Advanced Intelligent Systems*, vol. 3, no. 12, p. 2100014, 2021.

[20] Y. Teng, J. Zhang, and T. Sun, "RETRACTED: Data-driven decision-making model based on artificial intelligence in higher education system of colleges and universities," *Expert Systems*, vol. 40, no. 4, p. e12820, 2023.

[21] Z. Zong and Y. Guan, "AI-driven intelligent data analytics and predictive analysis in Industry 4.0: Transforming knowledge, innovation, and efficiency," *Journal of the Knowledge Economy*, vol. 16, no. 1, pp. 864–903, 2025.

Received: August 08 2025