JMBI (Journal of Marketing and Business Intelligence)
JMBI, Volume 04, Number 01, May 2026
EISSN: 2987-0879; PISSN: 2987-0860, pp. 42-52
https://journal.yhmm.or.id/index.php/JMBI

A LIGHTWEIGHT AI-BASED FRAMEWORK FOR MSME NETWORK PERFORMANCE OPTIMIZATION

Rahman Md Maksudur[1], Muhamad Hariz Bin Muhamad Adnan[1*], Satria Abadi[1]
[1]Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, Perak
Faculty of Computing & Meta-Technology Sultan Idris Education University, Campus Sultan Abdul Jalil Shah 35900 Tanjong Malim, Perak, Malaysia
E-mail: chymaksud175@gmail.com, mhariz@meta.upsi.edu.my, satriaabadi@meta.upsi.edu.my

Article history:
Received: March 23, 2026
Revised: April 9, 2026
Accepted: April 27, 2026

Corresponding authors
chymaksud175@gmail.com

Keywords:
MSME;
AI optimization;
NS-3 simulation;
congestion classification;
Random Forest

Abstract
Small businesses are now required to be efficient in all aspects of their operations, including hiring staff members, spending less money, and cutting back on extra cost. This also applies to network infrastructure. The majority of MSMEs utilize a network that is neither configured nor dynamic and is rarely monitored after installation since they cannot afford to hire a dedicated IT specialist. In most situations, these arrangements work fairly well. As a result, push them during times of high usage, peak traffic, sudden spikes in demand for cloud-based apps, and all open gaps: Uploads become slower, calls begin to decease, and pages begin to index. In this paper, we describe an AI-driven lightweight framework that uses a controlled simulation using NS-3 to address this problem. In order to model and learn the network using supervised learning models, this method employs two network simulations: one for normal MSME operation and one for generating the necessary congestion. A Random Forest (RF) classifier that can differentiate between normal and congested traffic and only launch a set of targeted and limited actions when there is congestion is trained using the flow level KPIs that are extracted from the simulations. Initial stress test shows good results: end-to-end latency decreased by approximately 37% and there was no change to throughput. The main drawback of this study is that everything has been tested and simulated, the next step is to replicate in real life. The goal of this study is to apply artificial intelligence (AI) to optimize MSME network infrastructure.

I. INTRODUCTION

MSMEs are the foundation of majority of the developing nations' economies. They contribute significantly to GDP, employment generation, and are

frequently at the center of innovation at the local level (OECD, 2017; World Bank, 2025). Moderate use of digital connectivity was typical for these companies ten years ago, but in recent years, it has emerged as a critical component. People's reliance on internet connections has drastically changed during the past ten years. Cloud-based storage, video conferencing, enterprise resource planning (ERP) software, and e-commerce platforms, which were once exclusive to large corporations, are now widely used in home-based enterprises and small offices (Cisco, 2022).

Even though MSMEs don't manage the network and have little staff and funding, network performance is a crucial business for them. The majority of small businesses find it impossible to hire a dedicated network expert, and even those that are able to do so typically configure network before letting it handle itself. As a result, an MSME network that is statically established will allocate the same bandwidth, queue, and route settings for months or years after setup day, regardless of changes in network traffic (Evanita & Fahmi, 2023; Umoga et al., 2024). Under typical network conditions, it is acceptable. They are extremely vulnerable to congestion during periods of high usage, which can result application to slow down, voice and video suffer, user getting irritated, and adversity in business operations.

An AI network management system might be a viable choice. A simple and straightforward concept is to always run the most crucial performance metrics, understand what normal performance look like, recognize when a deviation would indicate congestion, and take appropriate action to address congestion without waiting for an administrator to become aware of it and take corrective action (Ali & Awad, 2025). The research community is taking this approach, and AI-specific network management is already at the center of discussions about 6G architecture (You et al., 2020). Supervised classification techniques have shown promise in distinguishing between different network states in particular environment (Alqudah & Yaseen, 2020).

When the challenge is directed towards MSMEs, it's the availability of appropriate training data. However, there are very few instances of congestion in real MSME network with excellent network performance, and the supervised learning models must learn the model. In addition, training data is usually collected at non-peak or low load times and requires a substantial amount of data. This suggests that a classifier that is directly trained with actual MSME traffic would either overestimate or underestimate the amount of congestion it is meant to identify. MSMEs frequently have this data imbalance issue, although it may be more noticeable if there are a few brief periods of strong demand.

This study is unique because it provides a framework which is not restricted. In order to avoid waiting for the network to naturally experience congestion events, the method simulates a baseline of a typical network and adds a purposeful stress simulation to the baseline to create balanced training set. When it is in the high congestion condition, this data is forwarded to a Random Forest classifier, which is used to initiate a set of optimized lightweight actions. A functional network should not be active in and of itself, it can only be activated when and where it is needed.

RELATED WORK

These days, NS-3 is one of the most popular tools for closely examining network performance. It is an event-driven simulation model that can replicate protocol-level characteristics. The FlowMonitor module includes a collection of measures that are crucial for network optimization research, such latency,

throughput, packet loss, and jitter (Lavacca et al., 2020). It can be used in SDN frameworks, in a traditional IP system, and for a variety of necessary and non-physical test-setup studies.

Over the past ten years, machine learning in networking has grown significantly. It has been applied to a number of tasks, such as congestion forecasting, anomaly detection, traffic classification and quality of service prediction (Dhanya et al., 2019; Song et al., 2023). The latter is typically applied to the multidimensional and structured telemetry data using a Random Forest (RF) classifier. This approach, first introduced by Breiman (2001) and subsequently expanded by Lu et al. (2024), is a desirable choice for real world applications with heterogeneous settings where measurement accuracy cannot be guaranteed because to its robustness against noisy inputs and overfitting. In addition to increasing its applicability, the model's interpretability is crucial in an operational context when attempting to understand its forecast, especially when that prediction is one of the model's goals (Lu et al., 2025).

Most of the research on AI-based network optimization has been conducted in large-scale settings, including carrier-grade infrastructures, SDN environments, and enterprise data centers (Mestres et al., 2017). There are major distinctions between these settings and those found in MSMEs, though, as the former may lack centralized management plans, have a significant of historical traffic data, or have specialized operating teams. Applying these solutions in small business setting is not particularly easy, and research has shown that managing MSMEs involves a number of principles, one of which is light control, low overhead and selective intervention rather than constant intervention (Beshley et al., 2023; Yuwono et al., 2024).

Low usage enterprise networks are another kind of data imbalance that receives less attention. If the majority of the data comes from regular operations, most machine learning pipelines adopt a biased view of reality and consistently underweight or ignore congestion states that are most operationally significant for identifying. In order to get around this, this study attempts to directly address the issue by purposefully oversampling the congestion regime in the stress-based simulation. This way, the classifier can learn from a training set that is likely to contain useful decision boundaries between normal and degraded network conditions.

PROPOSED FRAMEWORK

Framework Overview

he suggested framework consists of the following four steps: The network is first simulated by NS-3 (in both baseline and stress-induced scenarios), flow-level KPIs are extracted from the NS-3 simulation output, a Random Forest model trained on extracted KPIs is used to classify the network congestion state, and light-weight optimization actions are then selected based on the classification outcome. Figure 1 shows the overall pipeline.

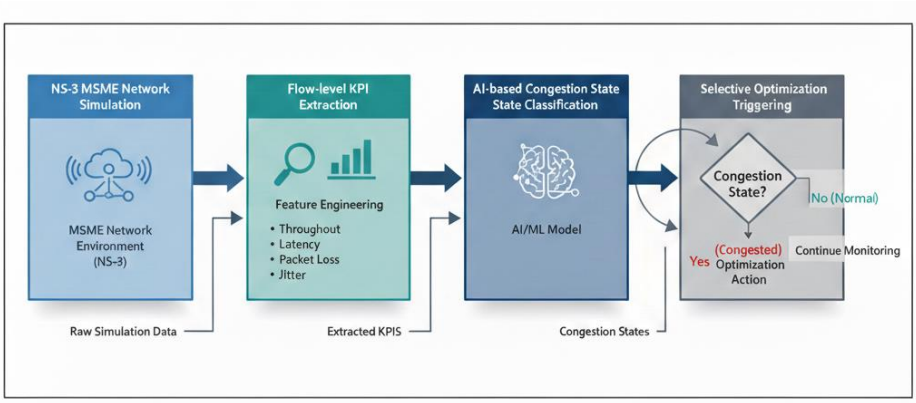

Figure 1: Proposed AI-based network optimization framework

Baseline MSME Simulation

The topology of a small business will be the foundation and a few of the client PCs will be connected to the same router that will route TCP traffic to a central server. The design is based on typical MSME network configurations that have been documented in the literature (Howard et al., 2022; Bhatti et al., 2022) and typical network conditions under normal conditions, which include no traffic "bottlenecks," controlled network queues, and no persistent network congestion. The scenario's drawback is that it only addresses half of the machine-learning problem. A classifier is less effective as a learner of the network congestion pattern when it occurs if it is not trained using any network stress since it has only been exposed to a little quantity of network stress. Baseline simulation provides the "LOW" half of the training data, while an additional procedure yields the "HIGH" half. The topology of our investigation is depicted in Figure 2.

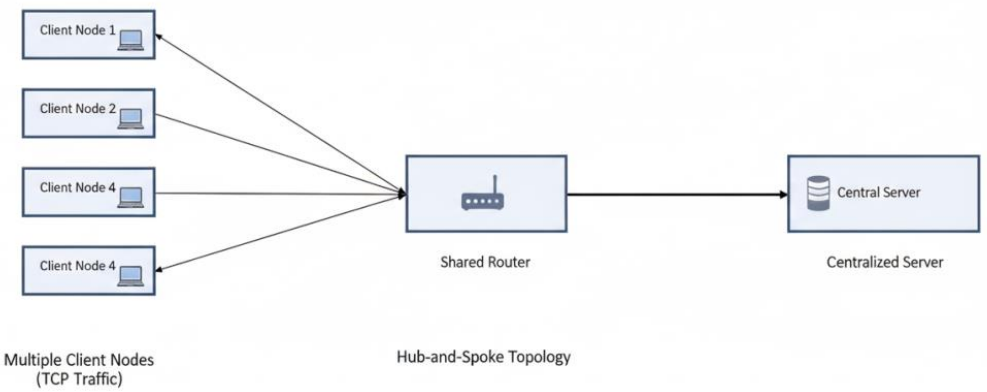

Figure 2: Baseline simulation network topology

Stress-Based MSME Simulation

The traffic samples that the classifier needs to learn are obtained by running a series of stress simulations with a comparable topology but increasing traffic volumes and a regulated amount of congestion on the router-to-server channel. Since the MSME will have developed during a busy time of day, this will then produce additional queues. The conditions replicated without waiting for them to occur in the actual world. Another solution uses two simulations to directly address the training data imbalance. In order to learn the two regimes required to learn the appropriate decision boundaries, some of the samples are included in the final training set. Figure 3 displays the topology structure of the

stress simulation.

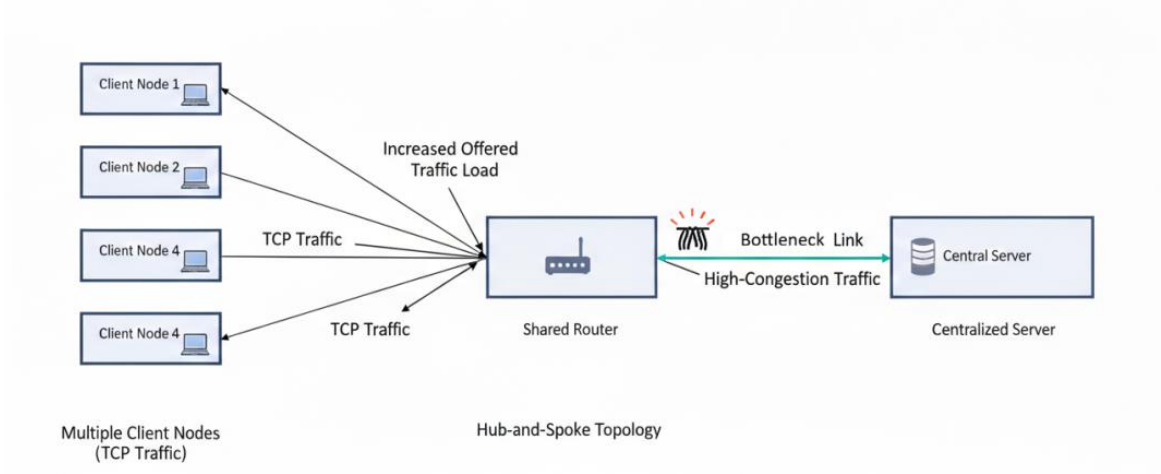

Figure 3: Stress-based simulation network topology

Dataset Construction and Feature Engineering

For both simulation phases, XML output is produced by the NS-3 FlowMonitor module. After being read, these files are transformed into tabular format, with each line denoting a single network flow. The remaining KPIs average end-to-end latency, throughput, packet loss ratio, jitter and utilization ratio are computed at the flow level. The ultimate result is a training set, which is a collection of all the records marked as LOW and HIGH for both kinds of simulation. A few performance metrics are utilized to differentiate between the LOW and HIGH congestion.

AI Model Training and Optimization Trigger

An offline Random Forest (RF) is trained using all of the collected data. The combined date is used to train an offline Random Forest (RF) classifier. The following factors led to the selection of RF: First, it functions effectively with tabular data; second, it can tolerate noise in relation to the input features and Thirdly, in a real-world business setting where business decisions must be made based on the model, the framework may be utilized to analyze the generated models in order to gain better knowledge of the features that contribute to the classification outcome (Huang et al., 2025). The classifiers parameters included 200 decision trees, a minimum sample split of 5, and a maximum depth of 10. Hyperparameters are carefully adjusted for a reasonable balance between computation and accuracy, and balanced class weights are employed to resolve any potential feature imbalance among distinct classes.

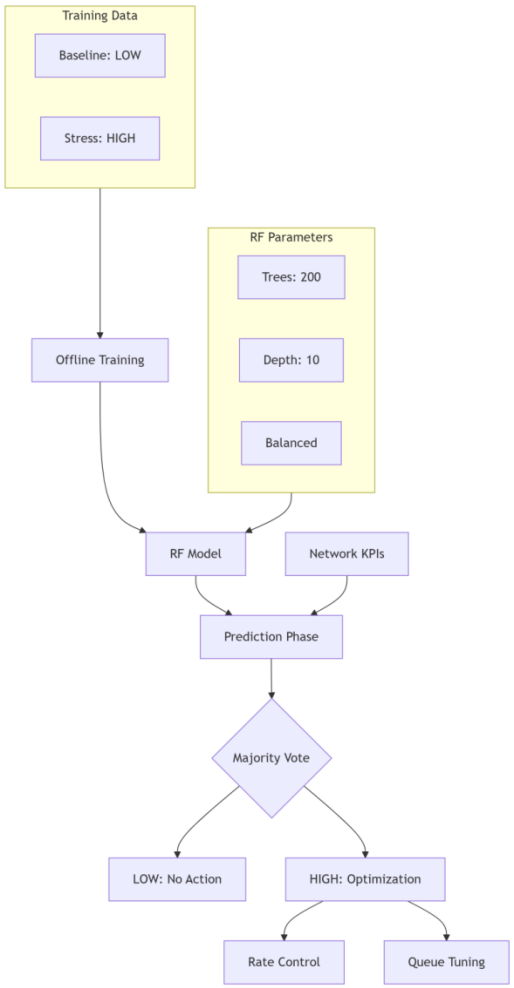

Figure 4: Random Forest (RF) classifier with training and prediction phases

The classifier predicts congestion after receiving a set of KPIs for the current flows. In the event that the prediction is HIGH: Client-side rate limiting; two optimizations will optimize the queue size at the rate limiting link. The goal of both intentional low-level reductions is to lessen the immediate congestion pressure without lowering throughput or adding more management overhead. There is no intervention if the prediction is LOW.

II. METHODOLOGY

This is accomplished through seven iterations of the baseline simulation, stress simulation, extraction and feature engineering, balanced dataset generation, random forest classifier training, AI-optimized simulation, and paired performance assessment. The entire process is shown in Figure 5.

Figure 5: AI-based optimization methodology

At the beginning, the simulation baseline mode is hub and spoke topologies, where a number of client nodes start a TCP traffic generator on a central server node through a common router. In this step, NS-3 FlowMonitor produces flow-level performance metrics including latency, throughout, packet loss, and jitter for input to the feature engineering pipeline for generation of raw XML data.

Next, during the stress simulation stage the bandwidth of the router-to-server link is 10 Mbps and the propagation delay is 20ms. These settings are selected to allow the queue packets to continue to increase and to have a low rate of queue packets in the HIGH congestion conditions. It is important because if no other, it would be more or less all 'low congestion' data and the classifier much less useful for the detection of 'real congestion'.

Then, the data is extracted and features engineered according to the stages of the simulation. The XML outputs are read and then converted back to a tabular format. For each flow, there are 7 KPIs computed: the average end-to-end latency, the average throughput, the packet loss ratio, the jitter, the number of active clients, offered load and utilization ratio. Each of these metrics has a number of occurrences that are used to determine whether or not it is LOW or HIGH; thresholds are provided for each of these metrics.

These balanced proportions are then used to combine labeled baseline (LOW) recordings together with stress related recordings (HIGH), to form a balanced set of recordings to train a stress related classifier. The data set is used with the previously set data for training RF classifier. The 200 decision trees make a prediction using a different combination of KPI features and the result of the last prediction of the congestion state is made by majority voting.

During the optimization phase, the trained classifier is incorporated into the simulation loop. The classifier is asked to forecast the congestion for each test scenario using the current set of KPIs. Rate limitation and queue tweaking are likely to occur in a high state. The prediction of "LOW" indicates that the network should operate flawlessly without any intervention.

This is achieved by comparing the performance of an AI-optimized setup with a baseline configuration, both of which are simulated under severe stress using equivalent n=22 paired simulations. For the pairing, the only thing that varies between each pair of configurations within each comparison is the on/off state of the AI driven optimization function. The other parameters, such as traffic flow, topology, initial conditions etc. remain unchanged. Four metrics: latency, throughput, packet loss ratio, and jitter are used to evaluate

improvements.

III. RESULTS AND DISCUSSION

This section presents the results of a preliminary evaluation conducted to validate the feasibility of the proposed AI-based network optimization framework. To assess the impact of the framework, a paired experimental design was utilized (n = 22), executing baseline and AI-optimized simulation under identical high-stress conditions.

Performance Metric Comparison

The 22 test runs were carried out in conjunction with the baseline and AI optimized configuration, the latter of which is shown in Table 1.

Table 1: Paired comparison of network performance metrics (n = 22)

Metric	Baseline (Mean ± SD)	AI-Optimized (Mean ± SD)	Mean Difference	95% CI of Diff	p-value (t-test)	Cohen's dz	% Change
Latency (ms)	112.05 ± 14.00	70.11 ± 15.13	−41.94	[−43.88, −40.00]	2.26439629440582 35e-22	−9.60	-37.43%
Throughput (Mbps)	10.55 ± 0.06	10.56 ± 0.07	+0.011	[−0.010, +0.033]	0.278 (NS)	0.24	+0.11%
Packet Loss Ratio	0.0277 ± 0.0122	0.0289 ± 0.0109	+0.00116	[+0.0001, +0.0023]	0.039	0.47	+4.18%
Jitter (ms)	11.03 ± 6.03	11.07 ± 5.49	+0.043	[−0.36, +0.44]	0.827 (NS)	0.05	+0.39%
Utilization	0.0480 ± 0.0587	0.0479 ± 0.0583	−0.00009	[−0.0003, +0.0001]	0.398 (NS)	-0.18	-0.18%

The most interesting outcome is the latency result. Average end-to-end latency was reduced by 70.11ms approximately 37.43% with AI optimization compared to the average end-to-end latency of 112.05ms without AI optimization. This improvement is consistently observed in all 22 paired runs, and is statistically significant both from the paired t-test ($p < 0.001$) and Wilcoxon signed rank test ($p < 0.001$) and has Cohen's dz of -9.60, which is an extremely large effect size. As this classifier is able to successfully remove congestion with a high degree of reliability, it's not a "marginal" result. The amount of throughput is also significant, but on a different level. There was only an insignificant difference between the two configurations (+0.11%, $p = 0.278$) in the average throughput. This is important, as this indicate that the improvements in latency are achieved without sacrificing network capacity.

The packet loss was slightly higher, but significantly after optimization (mean difference +0.00116' $p = 0.039$). It is counter intuitive in some ways since the fewer packets in the queue, hopefully, the fewer that will be lost. This is like related only in part to the possibility that the network can sometimes be rate limited beyond what is necessary which can affect rates of retransmission. The magnitude of the changes is sufficiently small such that it is within an operationally acceptable range for most MSME applications. These were not significantly different from each other ($p = 0.827$ for jitter and $p = 0.398$ for utilization). Combined, these results are a description of a system that is doing

what it was designed to do- selectively alleviating congestion-induced delay without adversely impacting the overall system.

Framework Efficacy

These results show that selective optimization can be triggered and high-congestion conditions can be accurately identified by Random Forest classifier. The framework keeps a small footprint and avoids unnecessary overhead during predicted "LOW" congestion periods by only acting when the "HIGH" condition is predicted. Real word MSME deployments are resource-constrained and cost-sensitive which is consistent with this chosen methodology.

Contextual Discussion with Existing Studies

It is important to explain what the suggested framework is and is not. The majority of the work on using AI to optimize networks has been done in environments with a lot congestion events like data centers, carrier SDN deployments, research-based testbeds, etc. where data imbalance is not the primary factor and where overall control is centralized (Mestres et al., 2017). These methods are frequently resource-intensive and rely on an operational model that is unavailable in the majority of small businesses. The proposed framework is specifically designed for MSME networks, and focuses on optimizing select processes, as opposed to constant intervention, using AI features. This study proposes stress-based simulation which is different from previous studies that had used plent of congestion samples, to generate high congestion samples in a systematic way for supervised learning. Moreover, current literature has mostly focused on implementing machine learning algorithms for traffic classification or prediction only (Dhanya et al., 2019; Song et al., 2023), whereas this paper incorporates the congestion classification directly into the problem of triggering of optimization. This paper shows how lightweight and simple interpret AI-based models can be practically applied to MSME context to achive better latency without high computational and operational costs.

Limitations

The following are some restrictions placed on this study. First, the evaluation is based on discrete-event simulations in NS-3, but not all physical layer variables will exist in operational environments. Second, the analysis is conducted within a hub-and-spoke network topology and for certain high stress traffic patterns, which may affect the applicability of the results to other mesh network topologies. IN addition, supervised learning method needs to use pre-defined performance thresholds for labeling, which can be specific to the context. Lastly the optimization actions considered are purposefully limited to avoid too much overhead, and are a subset of possible adaptive control actions.

IV. CONCLUSION

This research introduced a lightweight network optimization framework for MSME with AI, which is based on NS-3 discrete-event simulation and supervised machine learning. The framework incorporates stress-based simulation regimes which effectively overcome the problem of data imbalance that is common to nominal network operations, enabling effective classification of congestion. Preliminary results show that the network performance is improved significantly when optimized by AI, with an end-to-end latency reduced by 37.43% while maintaining the throughput stability. Future studies will be directed towards

expanding the evaluation covering a variety of network topologies and heterogeneous traffic. In addition, the framework will be tested on real-time emulation platforms and real traffic data from networks to assess the viability and scalability of the framework to be applied in operational enterprise deployments.

Acknowledgments

This research was funded by the Universiti Pendidikan Sultan Idris under the Penyelidikan Translasional Universiti (TR@UPSI) [2024-0050-109-01].

Author Contributions

Rahman Md Maksudur: Conceptualization, Methodology design, Simulation development, Data curation, Original draft preparation and analysis. Muhamad Hariz Bin Muhamad Adnan: Supervision, Validation, Reviewing, Editing and Analysis. Satria Abadi: Proofreading.

Declaration of Generative AI

During the preparation of this work, the authors used ChatGPT (OpenAI) to enhance the clarity, grammar, and readability of the manuscript. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

REFERENCES

A. Mestres et al., (2017). Knowledge-Defined Networking. SIGCOMM Comput. Commun. Rev. 47, 3 (July 2017), 2–10. https://doi.org/10.1145/3138808.3138810

Ali, Q. I., & Awad, S. R. (2025). Enhancing SDN Performance: Machine Learning Integration with the POX Controller for Dynamic Routing and Congestion Management. International Transactions on Electrical, Electronics and Computer Science, 4(3), 152–160. https://doi.org/10.62760/iteecs.4.3.2025.132

Alqudah, N., & Yaseen, Q. (2020). Machine learning for traffic analysis: A review. Procedia Computer Science, 170, 911–916. https://doi.org/10.1016/j.procs.2020.03.111

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/a:1010933404324

Beshley, M., Klymash, M., Scherm, I., Beshley, H., & Shkoropad, Y. (2023). Emerging network technologies for digital transformation: 5G/6G, IoT, SDN/IBN, cloud computing, and blockchain. In Lecture notes in electrical engineering (pp. 1–20). https://doi.org/10.1007/978-3-031-24963-1_1

Bhatti, S. H., Ahmed, A., Ferraris, A., Hussain, W. M. H. W., & Wamba, S. F. (2022). Big data analytics capabilities and MSME innovation and performance: A double mediation model of digital platform and network capabilities. Annals of Operations Research, 350(2), 729–752. https://doi.org/10.1007/s10479-022-05002-w

Cisco. (2022). Cisco Annual Internet Report (2018–2023). Cisco Systems, Inc.

Dhanya R. Mathews and J. Lakshmi. 2019. Service Resilience Framework for Enhanced End-to-End Service Quality. In Proceedings of the 18th Workshop on Adaptive and Reflexive Middleware (ARM '19). Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/3366612.3368123

Evanita, S., & Fahmi, Z. (2023). Analysis of challenges and opportunities for

micro, small, and medium enterprises (MSMEs) in the digital era in a systematic literature review. JMK (Jurnal Manajemen Dan Kewirausahaan), 8(3), 227. https://doi.org/10.32503/jmk.v8i3.4190

Howard, M., Böhm, S., & Eatherley, D. (2022). Systems resilience and SME multilevel challenges: A place-based conceptualization of the circular economy. Journal of Business Research, 145, 757–768. https://doi.org/10.1016/j.jbusres.2022.03.014

Huang, P., Li, Y., Gong, H., & Koara, H. (2025). Robust and Interpretable Machine Learning for Network Quality Prediction with Noisy and Incomplete Data. Photonics, 12(10), 965. https://doi.org/10.3390/photonics12100965

Lu, C., Cao, Y., & Wang, Z. (2024). Research on intrusion detection based on an enhanced random forest algorithm. Applied Sciences, 14(2), 714. https://doi.org/10.3390/app14020714

Lu, H., Dong, Y., Wu, Z., Wei, H., & Lu, G. (2025). New class detection in network traffic classification using confidence information embedded cascade structure. IEEE Transactions on Network Science and Engineering, 12(3), 1692–1706. https://doi.org/10.1109/tnse.2025.3538564

OECD (2017). Enhancing the Contributions of SMES in a Global and Digitalised Economy. In Meeting of the OECD Council at Ministerial Level (pp. 1-24). Paris: OECD Publishing. https://www.oecd.org/mcm/documents/C-MIN-2017-8-EN.pdf

T. Song, D. Markudova, G. Perna and M. Meo, "Where Did My Packet Go? Real-Time Prediction of Losses in Networks," ICC 2023 - IEEE International Conference on Communications, Rome, Italy, 2023, pp. 3836-3841, https://doi.org/10.1109/icc45041.2023.10278583

Umoga, U. J., Sodiya, E. O., Ugwuanyi, E. D., Jacks, B. S., Lottu, O. A., Daraojimba, O. D., & Obaigbena, A. (2024). Exploring the potential of AI-driven optimization in enhancing network performance and efficiency. Magna Scientia Advanced Research and Reviews, 10(1), 368–378. https://doi.org/10.30574/msarr.2024.10.1.0028

World Bank (2025). Small and Medium Enterprises (SMEs) Finance. https://www.worldbank.org/en/topic/smefinance

Yuwono, T., Novandari, W., Suroso, A., & Sudarto, S. (2024). Information and communication technology among MSMEs: Drivers and Barriers. The Eastasouth Management and Business, 3(1), 101–109. https://doi.org/10.58812/esmb.v3i1.340