eISSN: 2477-7994

JMECS (Journal of Measurements, Electronics, Communications, and Systems)

Vol.11, No.1, Pg.22-29, June 2024

Interest Classification on Named Data Network Using the Supervised Learning Method

Sri Astuti[1*], Ratna Mayasari[2] and Ibnu Asror[3]

[1,2] *School of Electrical Engineering, Telkom University, Bandung, 40287, Indonesia*

[3] *School of Informatic Engineering, Telkom University, Bandung, 40287, Indonesia*

[*]*sriastuti@telkomuniversity.ac.id.*

Manuscript received May 20, 2024; revised June 16, 2024; accepted June 29, 2024.

Abstract

Named Data Network (NDN) is a next-generation network architecture that shifts the traditional data communications paradigm Unlike conventional networks that rely on IP addresses, NDN delivers content based on data names rather than specific locations. In NDN, consumers express their requests by sending interest packets containing content names. These names are then propagated through the network nodes, which forward them to the appropriate destinations. The forwarding strategy in an NDN network plays a crucial role in ensuring efficient data delivery. This strategy includes a set of rules that determine the next hop for each interest packet. These rules are designed to optimize the forwarding process, minimizing delays and improving network efficiency. However, if the forwarding strategy is implemented without accurately identifying the appropriate face (i.e., the network interface) to forward interests toward the producer or the nearest cache node, it can lead to significant delays and packet drops. This, in turn, negatively impacts Quality of Service (QoS) parameters and the overall performance of the NDN network. This study applies supervised learning to classify consumer-requested interests to overcome this issue. This technique leverages several related variables to accurately classify these interests. The outcomes of the conducted research demonstrated that raw data from the mini-NDN output can be processed and transformed into a usable dataset. This data is then utilized to train a classification model with supervised learning. In a scenario with 9 NDN nodes and varying numbers of interests, distributed both uniformly and according to Zipf's law, the Random Forest model performs effectively, achieving an accuracy rate of 86.2% with an error rate of 14.8%.

Keywords: *Named Data Network, Interest, QoS, Forwarding Strategy, Supervised Learning*

DOI: 10.25124/jmecs.v11i1.8100

1. Introduction

The existing internet architecture, built upon the traditional host-centric Internet Protocol (IP), faces significant challenges in content distribution and scalability [1] . The Named Data Network (NDN) architecture has been proposed as a solution to address these issues. NDN is an information-centric networking (ICN) architecture. designed to enhance data transfer efficiency, particularly in heterogeneous wireless networks, compared to traditional IP-based networks. In NDN, content is identified by its data name rather than the IP address of the hosting device [2].

Future network architectures must accommodate broadband multimedia content services with diverse characteristics. According to ITU-T, the next generation of networks will be data-aware and centered around content and data distribution. The Named Data Networking (NDN) architecture changes the way the Internet operates by focusing on data itself rather than on the devices that host it, enabling faster data transfers. In an NDN network, the system can be simulated and modeled using routing, forwarding, and caching mechanisms. These mechanisms are important, as they directly influence the network QoS [3].

The implementation of routing and forwarding mechanisms in an NDN network is crucial for determining its overall performance. Therefore, selecting the appropriate forwarding strategy is essential to ensure optimal performance in an NDN-based network. This paper proposes the use of supervised learning to classify forwarding strategies for NDN networks. The classification is based on data collected from an NDN network emulation using mini-NDN. The resulting classification serves as a recommendation for selecting the most suitable forwarding strategy for a given NDN network to achieve optimal performance.

This study focuses on the classification of content requests and provides a detailed analysis of the performance metrics for each classification model. The remainder of this paper is organized as follows: Section 2 reviews related work on forwarding strategies in NDN networks. Section 3 presents the proposed system model for classifying interfaces. Section 4 analyzes the performance of various classification models on the dataset. Finally, Section 5 concludes the paper and summarizes the key findings.

2. Research Method

This study employs an experimental approach involving multiple trials, result validation, performance evaluation of the classification model, and analysis of the findings.

Data was collected through the emulation process using the mini-NDN emulator, transforming the gathered data into a dataset for input for the machine learning model. The result serves as measurement parameters to assess the performance of the classification model. The research steps are described in further detail in the following subsection.

This study used an experimental method, which consisted of the following stages:

1. *Information literacy*: The researchers conducted a literature review to study the state-of-the-art classification models for their specific tasks of interest.
2. *Data collection*: The researchers collected a dataset of measurement data, which they pre-processed to a format suitable for input in the machine learning models.

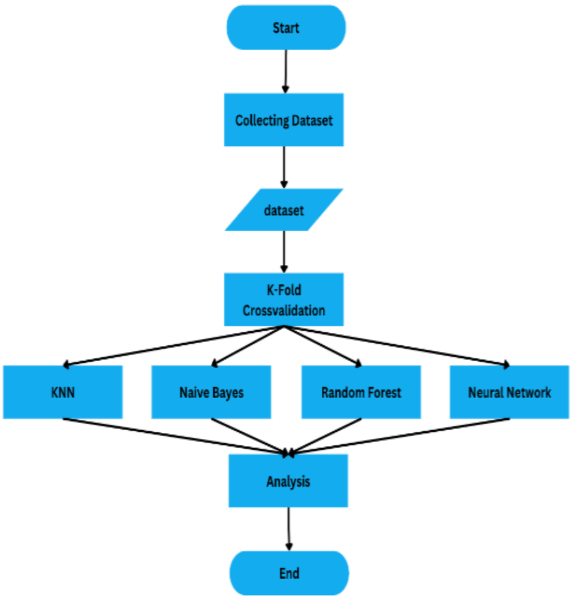

Fig. 1. Design System

3. *Data pre-processing*: The researchers pre-processed the data to improve its quality and make it more suitable for machine learning. This may have involved cleaning the data, removing outliers, or transforming the data into a different format.
4. *Evaluation of classification methods*: We trained and evaluated multiple machine learning classification models on the pre-processed data. The various evaluation metrics for model performance are accuracy, precision, recall, and F1 score.

The study's final result included comparing the performance of various classification methods on the given dataset. This information can guide the selection of the most suitable machine-learning classification model for the specific task.

2.1 Data Collection from NDN Emulator.

The data obtained from testing with the NDN emulator, known as Mini-NDN, is used as raw data This data is subsequently processed and transformed using pre-processing techniques to create a usable dataset. In this study, data collection is conducted by running multiple testing scenarios on a basic topology consisting of nine nodes. The topology is implemented in an NDN network environment. The basic topology consists of nine nodes: two nodes as producers, one node as a consumer, and six nodes as NDN nodes [4]. The scenarios were executed with varying interest request rates and prefixes, ranging from 100 interests per second to 10,000 interests per second.

Interest is the message that is requested by consumers to the NDN network. Consumers (users) send the interest to the network, and the content providers reply to the requested data to the consumers. Prefix is interpreted as a directory or folder in the system's file structure (e.g., NDN/Data-NDN/content4.pdf). Node C acts as the consumer node that requests interest, while nodes P1 and P2 serve as content producers. The remaining six nodes function as intermediary nodes, responsible for forwarding interests to the content producer nodes.

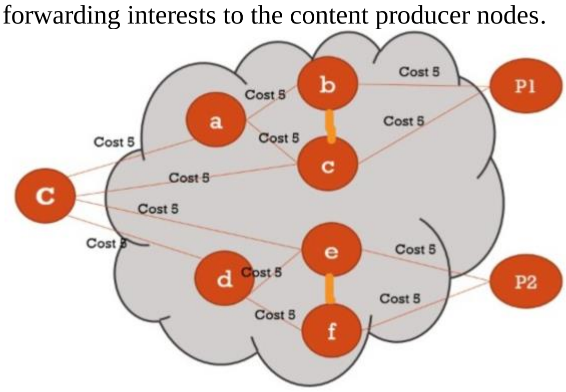

Fig. 2. NDN Topology

Fig. 3. Log Data of Emulator NDN

Node C (consumer) sends a request using an interest message to the NDN network. An intermediate node then forwards the interest message to the content provider node through a connection path, or interface, between nodes. The content data is then returned to the consumer via the same route or interfaces. Interface examples are 259, 260, 261, 262, 263, 264, etc. The shorter the distance between the content provider node and the consumer, the lower the delay (RTT). If the interest message requested by the consumer is available and successfully delivered, the interest status is successful (Acknowledgement = ACK). Conversely, if the interest message is unavailable or not received by the consumer, the interest status is marked as failed (Not-Acknowledgement = NACK).

2.2 Pre-Processing Data.

The pre-processing stage is crucial for ensuring that the data used in the model is clean, relevant, and in the appropriate format. It involves processing the data obtained from the mini-NDN emulator. In this research, the pre-processing steps include feature imputation, duplicate data, and feature reduction. The goal of these pre-processing steps is to prepare valid data for use in machine learning.

a) Feature Imputation
   Feature imputation is used to address missing data. If any data is missing, it is filled with the mode or median value of the dataset. However, in this study, no missing data is present.
b) Duplicate Data.
   Duplicate data is a common issue in data processing, particularly when data is collected from multiple sources or entered automatically. Handling duplicate data involves identifying and removing duplicates that could invalidate the dataset. In this study, duplicate data was removed from the dataset.
c) Feature Reduction.
   Feature reduction is a technique used to eliminate less relevant features from a dataset while preserving essential information. It aims to address issues such as overfitting, increased computation time, and model complexity. In this study, several non-essential attributes were removed from the dataset to improve model performance and efficiency.

Table 1. Data Attributes in Dataset

Data Attributes	Details
Node	The entity is responsible for processing and forwarding data packets and interests within the network.
Prefix	The initial part of a data name is used to identify and organise data within a certain hierarchy.
Interest Status	The status of the interest requested by the consumer on the network.
Interfaces	A communication channel or interface is used by a router or node to send and receive data packets.
Round-Trip Delay	The total time required for an interest to be sent from the source to the destination and back to the source.

2.3 Dataset

Our contribution to this research is the dataset. The data was collected using the NDN emulator (mini-NDN) in the .log format, as illustrated in Figure 3. The log data is processed into dot CSV (.csv) format for analysis. Several specific data attributes or features are selected to be included in the dataset. Some of these attributes are explained in Table 1.

Table 1 shows the attributes included in the dataset. The dataset comprises five key attributes: node, prefix, interfaces, interest status, and round-trip delay. Among these, the interest status serves as the class label, where NACK is represented by 0 and ACK by 1. The total dataset used in this study consists of 1,042 entries.

2.4 K-Fold Cross Validation.

Model validation is necessary to accurately assess each classification model's performance. This study uses K-Fold Cross-Validation, a common model validation technique in machine learning. K-Fold Cross-Validation is a model evaluation methode used to improve model accuracy. This technique divides the training data into K folds of equal size, then trains and tests the model K times. In each iteration, onefold is used as the testing data, while the remaining K−1 folds are used as the training data. The results of each iteration are averaged to estimate the model's overall performance.

Figure 4 illustrates K-Fold Cross-Validation. K-Fold Cross-Validation randomly splits the data into K-independent subsets, with each fold containing 1/K of the data. This study used a value of K = 10, so each fold comprises 1/10 of the data training.

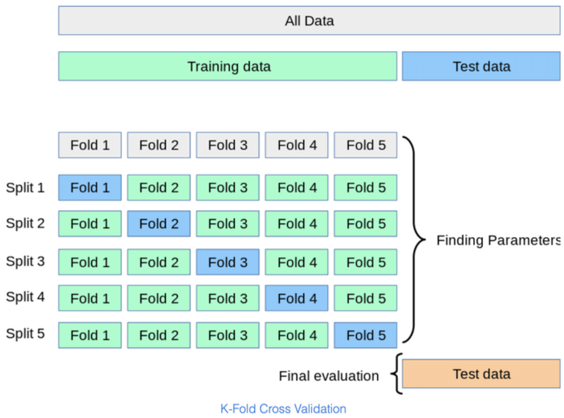

Fig. 4. Illustration of K-Fold Cross Validation

2.5 Named Data Networking

NDN represents an evolution in internet technology, shifting its focus from the traditional IP (Internet Protocol) approach to content- or data-centric communication. This shift fundamentally changes the way new technologies are developed and conceptualized [5]. Although NDN retains a similar architecture as IP, this evolution of NDN changes the paradigm of how it works, from transmitting packets to the target address to matching data identified by name. This shift provides several advantages, including easier data naming, stronger security, greater mobility, and a more efficient broadcasting process [6].

There are several key principles in designing the NDN architecture. First, it follows the hourglass model architecture similar to IP. Security must be integrated directly into the system, rather than being treated as a separate layer. NDN also adheres to the end-to-end principle. Additionally, network traffic must be self-regulating. There is a clear separation between routing and forwarding functions, and the architecture should ultimately support user choice[7].

NDN communication starts with the recipient or consumer. The consumer begins a request by transmitting an interest packet specifying their desired content. The router then records this information and network details and forwards the packet to its destination. Once the packet reaches the producer or content provider, the requested data is returned to the consumer along the same path, accompanied by the producer's signature key [8].

NDN router contains three main elements [9], namely:

1) *Forwarding Information Base (FIB)*: A table that stores interface information and indexes names used as references in sending packets on NDN.
2) *Pending Interest Table (PIT)*: A table that contains interest packets that have pending status or are not yet satisfied. The PIT that records the interest's name and incoming and outgoing interfaces.
3) *Content Store (CS)*: a temporary storage place for data content. Data content originating from the nearest producer or node will be cached on each router it passes to satisfy future Interests.

2.6 Forwarding in NDN

NDN forwarding relies on the optimal path selected by the prior routing procedure and the name contained within the interest packet. This name is a defining characteristic that sets the interest packet apart. This method mitigated several challenges faced by the IP architecture, namely NAT translation, address limitations, mobility, and network scalability. Therefore, routing and forwarding have a different role in NDN than in IP networks [10].

Regarding the forwarding mechanism, NDN has a smarter forwarding process than IP. In IP, the routing and forwarding process has a "smart routing, dumb forwarding" approach, so data packet delivery will focus on the routing process. On the other hand, adaptive forwarding, which creates the network's stability and flexibility, is much different and better than that on IP [11].

The network forwarder used in NDN is the NDN Forwarding Daemon (NFD), which is responsible for implementing the NDN communication protocol. According to the NFD Developer's Guide, the list of forwarding strategies implemented in NFD includes the following:

1) *Best Route*: forwards the consumer's interest to the network using the route with the lowest routing cost.
2) *Multicast*: forwards the consumer's interest to all upstream directions as specified by the Forwarding Information Base (FIB), except for the target interface in the FIB that corresponds to the requesting node.
3) *Access*: This strategy is specific to the local prefix on the access router, which reaches the producer using a single hop. It also utilizes multiple paths in the FIB and can recover packet loss at the last hop.
4) *Random*: randomly forwards interest to the next hop in the FIB, based on the Loop-Free Inport-Dependent (LFID) routing protocol.

2.7 Supervised Learning for Classification

Supervised learning methods provide a promising approach for automating the classification of content requests. These models learn from labeled data, capturing patterns and relationships that allow them to make predictions on previously unclassified content requests, as illustrated in Figure 5 [12].

This research compares the performance of various machine learning classification models, including Naive Bayes, Decision Tree, Random Forest, and Neural Network, to determine which model achieves the highest accuracy for the given task.

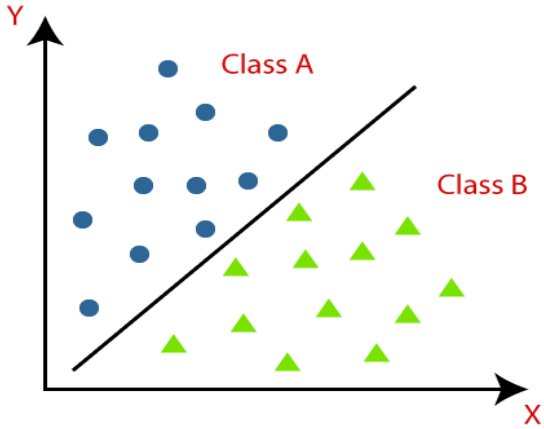

Fig. 5. Classification Model

1. Naïve Bayes
   A classification method that is easy to apply and based on the simple concept of probability, Naive Bayes is a technique grounded in basic probability principles, assuming that the explanatory variables are independent. Naive Bayes performs quite well when used with very large datasets. It uses conditional probability as defined by Bayes' theorem. One of the key advantages of the Naive Bayes algorithm is its lower error rate with large datasets, as well as its higher accuracy and faster performance when applied to larger data collections [13].
2. Random Forest
   The Random Forest method is an ensemble technique that improves the accuracy of classification methods by combining multiple classifiers. This algorithm integrates several decision trees into a single model. Typically, Random Forest is used for regression and classification problems involving large data sets, as it can handle many dimensions with varying scales while maintaining high performance. Random Forest employs decision trees in the selection process, where the trees are recursively split based on data belonging to the same class. In this case, using more trees results in more optimal accuracy. The classification in Random Forest is determined through a voting process based on the trees generated [14].
3. K-Nearest Neighbour
   The K-nearest neighbour algorithm identifies similarities between new and existing data by classifying data points based on their proximity to nearby groups of other data points. As a result, this algorithm produces competitive outcomes. Essentially, K-nearest neighbour retains all previous data and classifies new data points based on their similarity to the existing data. The K-NN training process involves testing multiple values of k to find theoptimal one, resulting in high accuracy. The use of pseudocode in the kNN algorithm demonstrate that K-NN is simple, easy to implement, and requires only one parameter, k, to be set [15].
4. Neural Network
   The human brain inspires Neural Networks. In humans, neurons receive signals or stimuli from outside and then send them to the brain for processing. After that, the body will act according to the commands sent by the brain through neurons. Neural networks utilize a Multilayer Perceptron (MLP) algorithm, capable of learning linear and non-linear models. MLP is the most commonly used form of neural network for simple classification problems, both in research and in practice [15].

3. Performance Evaluations

This part discusses metrics, analysis, and model performance evaluation. The performance metrics for this model include F1-score, accuracy, recall, and precision.

3.1 Metric Performance Analysis

This research evaluates the model by comparing actual outcomes with predicted values. The Confusion Matrix is a tool used to assess classification tasks in machine learning, where the output can fall into two or more categories. It presents four possible combinations of predicted and actual results: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Using the Confusion Matrix, we can calculate recall, accuracy, F1-score, and precision.

The F1-score is the harmonic mean of precision and recall, offering a balanced evaluation of the model's overall performance by considering both metrics. Accuracy reflects how well the predicted values align with the actual values, calculated by dividing some of the correct predictions by the total data. Recall assesses the system's effectiveness in retrieving relevant data, calculated by dividing some of the true positives (TP) by the total of actual positives. Precision measures the system's ability to deliver relevant information in response to user queries and is determined by dividing some of the true positives (TP) by the total positive predictions.

These metric values were determined and can be computed using the Confusion Matrix for each classification.

Table 2. The Confusion Matrix of Naïve Bayes Model

		Predicted		
	Class	**0**	**1**	**Total**
Actual	**0**	138	127	**265**
	1	82	695	**777**
	Total	**220**	**822**	**1042**

Table 3. The Confusion Matrix of Random Forest Model

	Predicted			
	Class	**0**	**1**	**Total**
	0	191	74	**265**
Actual	**1**	70	707	**777**
	Total	**250**	**792**	**1042**

Table 4. The Confusion Matrix of Neural Network Model

	Predicted			
	Class	**0**	**1**	**Total**
	0	157	108	**265**
Actual	**1**	69	708	**777**
	Total	**226**	**816**	**1042**

Table 5. The Confusion Matrix of K-Nearest Neighbour Model

	Predicted			
	Class	**0**	**1**	**Total**
	0	183	82	**265**
Actual	**1**	67	710	**777**
	Total	**226**	**816**	**1042**

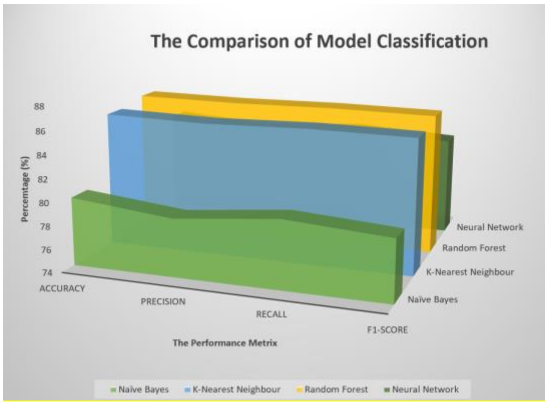

Fig. 6. The Comparison of Classification Model Performances

The confusion matrix for the Naïve Bayes model is shown in Table 2 above. Indicating that 833 data were correctly predicted, while 209 were misclassified. The confusion matrix for the Random Forest model is shown in Table 3 above. Showing that 896 data were correctly predicted, and 144 were misclassified.

Table 4 shows the Confusion Matrix for the Neural Network model. Indicating that 865 data were correctly predicted, while 177 data were misclassified. The confusion matrix for the K-Nearest Neighbour model is shown in Table 5 above. Showing that 893 data were correctly predicted, and 149 were misclassified. The Random Forest classification model is the most accurate, with 896 correctly predicted and 144 incorrectly predicted data points. Therefore, the Random Forest model is the best model for predicting the outcome of a particular input.

3.2 Performance Comparison of Classification Model

Based on the classification results, performance metrics were calculated for each of the four classification models used in this study: (Neural Network, Random Forest, K-Nearest Neighbor, and Naïve Bayes). Additionaly, a comparison analysis was conducted by evaluating the performance metrics of each classification model. The values for each classification were derived from the corresponding confusion matrix. Figure 6 above presents a comparison of the model performance.

As shown in Fig. 6., the accuracy levels of each model differ. The Naïve Bayes model has the lowest accuracy at 79.9%. While the Random Forest model achieves the highest accuracy at 86.2%. The error rate for the Random Forest model in the classification process is 14.8%. Based on these results, this study recommends the Random Forest model as the best classification model, with the highest accuracy. Additionally, applying 10-fold cross-validation to each classification model helps reduce variance and bias, ensuring that each fold has an equal opportunity to be used for both training and testing.

4. Conclusions

In this paper, we evaluated the performance of classification models on a forwarding dataset. Our research demonstrates that the raw data obtained from the mini-NDN emulator was processed into a dataset suitable for input into a supervised learning classification task. The classification models used Naïve Bayes, Random Forest, K-Nearest Neighbor, and Neural network models with a 10-fold cross-validation technique. In the scenario involving 9 NDN nodes with varying interest counts, both uniformly and Zipf-distributed, we recommended using the Random Forest model with an accuracy rate of 86.2% and an error rate of 14.8%.

References

[1] S. Fayyaz, M. Atif Ur Rehman, M. Salah Ud Din, Md. I. Biswas, A. K. Bashir, and B.-S. Kim, "Information-Centric Mobile Networks: A Survey, Discussion, and Future Research Directions," *IEEE Access*, vol. 11, pp. 40328–40372, 2023, doi: 10.1109/ACCESS.2023.3268775.

[2] "Named Data Networking in Vehicular Ad Hoc Networks: State-of-the-Art and Challenges | IEEE Journals & Magazine | IEEE Xplore." Accessed: Oct. 02, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/8624354

[3] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, "Recent Advances in Information-Centric Networking-Based Internet of Things (ICN-

IoT),” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2128–2158, Apr. 2019, doi: 10.1109/JIOT.2018.2873343.

[4] S. Astuti, T. A. Wibowo, R. Mayasari, I. Asror, and G. P. Satriawan, “Data Delay Classification with LFID forwarding strategy using Machine Learning to maximize NDN network performance”.

[5] “Named Data Networking: A Survey on Routing Strategies | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Oct. 02, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9864583

[6] “A Brief Introduction to Named Data Networking,” Named Data Networking (NDN). Accessed: Oct. 02, 2024. [Online]. Available: https://named-data.net/publications/ndn18/

[7] “Towards named data networking technology: Emerging applications, use cases, and challenges for secure data communication - ScienceDirect.” Accessed: Oct. 02, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0167739X23003631

[8] L. Mekinda and L. Muscariello, “Supervised Machine Learning-Based Routing for Named Data Networking,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6. doi: 10.1109/GLOCOM.2016.7842307.

[9] A. Tariq, R. A. Rehman, and B.-S. Kim, “Forwarding Strategies in NDN-Based Wireless Networks: A Survey,” *IEEE Commun. Surv. Tutor.*, vol. 22, no. 1, pp. 68–95, 2020, doi: 10.1109/COMST.2019.2935795.

[10] S. Ahdan, H. Situmorang, and N. R. Syambas, “Forwarding strategy performance in NDN network: A case study of palapa ring topology,” in *2017 3rd International Conference on Wireless and Telematics (ICWT)*, Jul. 2017, pp. 20–25. doi: 10.1109/ICWT.2017.8284131.

[11] “On Sharing an FIB Table in Named Data Networking.” Accessed: Oct. 02, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/9/15/3178

[12] L. Mekinda and L. Muscariello, “Supervised Machine Learning-Based Routing for Named Data Networking,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6. doi: 10.1109/GLOCOM.2016.7842307.

[13] “Machine Learning Fundamentals Handbook – Key Concepts, Algorithms, and Python Code Examples,” freeCodeCamp.org. Accessed: Oct. 02, 2024. [Online]. Available: https://www.freecodecamp.org/news/machine-learning-handbook/

[14] B. Taha Jijo and Adnan Mohsin Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning”, Accessed: Oct. 02, 2024. [Online]. Available: https://www.researchgate.net/publication/350386944_Classification_Based_on_Decision_Tree_Algorithm_for_Machine_Learning

[15] S. Smys, R. Bestak, R. Palanisamy, and I. Kotuliak, Eds., *Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021*, vol. 75. in Lecture Notes on Data Engineering and Communications Technologies, vol. 75. Singapore: Springer Nature, 2022. doi: 10.1007/978-981-16-3728-5.

Author Information

Sri Astuti is a lecturer in the School of Electrical Engineering, Telkom University. He received his master’s degree in Electrical Engineering of Telkom University in 2014. His research is focused on machine learning for networking.

Ratna Mayasari is a lecturer in the School of Electrical Engineering, Telkom University. He received his master’s degree in Electrical Engineering of Telkom University in 2013. His research is focused on machine learning for networking.

Ibnu Asror is a lecturer in the School of Informatics Engineering, Telkom University. He received his master’s degree in Electrical Engineering of Telkom University in 2011. His research is focused on machine learning.

Additional Information

Open Access. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. If material is not included in the article's Creative Commons license CC-BY-NC 4.0 and your intended use it, you will need to obtain permission directly from the copyright holder. You may not use the material for commercial purposes. To view a copy of this license, visit https://creativecommons.org/licenses/by-nc/4.0/