e-ISSN : 2829-4149
p-ISSN : 2829-4157

COELITE

JOURNAL OF COMPUTER ENGINEERING, ELECTRONICS AND INFORMATION TECHNOLOGY

OJS :

Publisher :

The Education University

Vol. 04| Issue. 01 | Apr 2025

Volume 4, Issue 1, April 2025

e-ISSN : 2829-4149
p-ISSN : 2829-4157

Journal of
Computer Engineering, Electronics and Information Technology
Universitas Pendidikan Indonesia

Jl. Pendidikan No.15, Cibiru Wetan, Kec. Cileunyi, Kab. Bandung, Jawa Barat 40625
Phone: (022) 7801840, Fax: (022) 7830426
E-Mail: coelite@upi.edu, Website: https://ejournal.upi.edu/index.php/COELITE

Editor in Chief
Dr. Eng. Munawir, S.Kom., M.T., (Scopus ID 57208758022)
Expertise: Data Mining, Text Mining.
Universitas Pendidikan Indonesia, Indonesia.

Executive Editor
Anugrah Adiwilaga, S.ST., M.T., (Scopus ID 57194714935)
Expertise: Robotic, Control and Intelligent Systems, AI, IoT.
Universitas Pendidikan Indonesia, Indonesia.

Editorial Board
Dr. Eng. Dinda Pramanta, M.Eng.
Expertise: Artificial Intelligence, FPGA, Robot and Neural Network.
Associate Professor at Kyushu Institute of Information sciences Japan.

Wirmanto Suteddy, S.T., M.T.,
Expertise: Embedded System, IoT, AI, VLSI, Computer Vision.
Universitas Pendidikan Indonesia, Indonesia.

Dr. Eng. Mia Rizkinia, S.T.,M.T.
Expertise: Optoelectrotechnique and Remote Sensing, Vision, Image and Graphics Processing.

Editorial Assistant
Deden Pradeka, S.T., M.Kom., (Scopus ID 57200731472)
Expertise: Software Engineering Development, Mobile Development, Cryptography, Steganography.
Universitas Pendidikan Indonesia, Indonesia.

Reviewers

Dr. Eng. Md. Zaman Molla
Expertise: Solar Cell.
Associate Professor of Physics at Ahsanullah University of Science and Technology Bangladesh.

Dr. Eng. Alfian Akbar Gozali, S.T., M.T
Expertise: Data Analytics, Machine Learning, Optimization Algorithm, Mobile Development.
Telkom University, Indonesia.

Dr. Eng. Bobby Kurniawan, S.T., M.T
Expertise: Production system and control, mathematical modeling, scheduling metaheuristics, AI.
Universitas Sultan Ageng Tirtayasa, Indonesia.

Syamsul Rizal S.T, M.T., PhD.
Expertise: Image Processing, IoT,Faulty Node, Fault Detection.
Telkom University, Indonesia.

Devi Aprianti Rimadhani Agustini, S.Si, M.Si
Expertise: Modeling and Simulation.
Universitas Pendidikan Indonesia, Indonesia.

Muhammad Taufik Dwi Putra, S.Tr.Kom., M.T.I
Expertise: Computer vision, deep learning, Machine Learning.
Universitas Pendidikan Indonesia, Indonesia.

Ana Rahma Yuniarti, S.T., M.Eng., (SCOPUS ID 57192948351)
Expertise: Biomedical Engineering, Machine Learning, Artificial Inteligence.
Universitas Pendidikan Indonesia, Indonesia.

Journal of Computer Engineering, Electronics and Information Technology
published by **Universitas Pendidikan Indonesia.**

Indexed by.
Google Scholar

Volume 4, Issue 1, April 2025

e-ISSN : 2829-4149
p-ISSN : 2829-4157

Journal of

Computer Engineering, Electronics and Information Technology

COELITE

Volume 4, Issue 1, April 2025
Table of Contents

Pneumonia Disease Detection in X-Ray Images Using A Deep Learning Approach with CNN and Alexnet Architecture 1-12
Anissa Putri A. Karim, Abdul Mubarak, Hairil Kurniadi Sirajudin

Hybrid Explainable AI Approach for DNA Sequence Classification: Feature Importance, Permutation Importance, and Local Explanations with LIME 13-24
Nur Alamsyah, Budiman Budiman, Reni Nursyanti, Elia Setiana, Venia Restreva Danestiara, Titan Parama Yoga

A Metaheuristic-Based Approach to Inflation Prediction in Indonesia with Support Vector Regression (SVR) 25-34
Putu Manik Sesa Arianti, Miftakhul Jannah, Noviyanti Santoso

Optimizing Logistic Regression for Digital Marketing Campaigns: Insights from Hyperparameter Tuning with Optuna 35-50
Budiman Budiman, Nur Alamsyah, Elia Setiana, Fikri Muhamad Fahmi

Digital Data Security Using a Combination of Base64 Encoding, Rail Fence Cipher, and GZIP Compression 51-60
Deden Pradeka, Zahra Khaerunnisa, Syifa Aqila Humaira, Aidha Salsa Billa, Murnawan Murnawan, Budiman Budiman

Published by:
Universitas Pendidikan Indonesia
E-Mail: coelite@upi.edu
Website: https://ejournal.upi.edu/index.php/COELITE

Journal of Computer Engineering, Electronics and Information Technology (COELITE)

Journal homepage: https://ejournal.upi.edu/index.php/COELITE

Pneumonia Disease Detection in X-Ray Images Using A Deep Learning Approach with CNN and Alexnet Architecture

Anissa Putri A. Karim[1], Abdul Mubarak[2], Hairil Kurniadi Sirajudin[3]*

[1,2,3] Program Studi Informatika, Fakultas Teknik, Universitas Khairun, Indonesia
Correspondence E-mail: anissaakarim@gmail.com

ABSTRACT

Effective treatment of pneumonia, a respiratory illness, depends on a prompt and precise diagnosis. However, using medical investigations to diagnose pneumonia can be laborious and rely on the skill of radiologists. To automatically identify pneumonia from chest X-ray pictures, this study intends to create an artificial intelligence model utilizing a Convolutional Neural Network (CNN) with the AlexNet architecture. 2,806 X-ray pictures that were classified as either normal or pneumonia were used. A variety of preprocessing methods were used to improve the quality of the data, and AlexNet, which had previously been trained on ImageNet, was used for transfer learning to increase the accuracy of detection. The model's accuracy, precision, and recall were 95.44%, 99%, and 94%, respectively. However, because Google Colab uses temporary sessions, speed varies a little when it is rerun. Despite this, the model continuously maintains an accuracy of over 90%. Furthermore, users can upload X-ray photos and get immediate results using a Gradio-based interface, which makes it accessible to those without technical knowledge. This study lays the groundwork for using AI in the diagnosis of pneumonia to increase the effectiveness and speed of medical imaging analysis.

ARTICLE INFO

Article History:
Submitted/Received 23 Feb 2025
First Revised 17 Mar 2025
Accepted 27 Mar 2025
First Available online 1 Apr 2025
Publication Date 1 Apr 2025

Keyword:
Pneumonia,
Convolutional Neural Network,
AlexNet,
X-Ray Image,
Deep Learning.

© 2025 Universitas Pendidikan Indonesia

1. INTRODUCTION

Pneumonia is a respiratory tract infection marked by lung inflammation caused by bacteria, viruses, or fungi. This condition disrupts oxygen absorption and leads to breathing difficulties due to inflammation and fluid buildup in the alveoli, the air sacs responsible for gas exchange [1]. It poses a significant health risk, particularly to vulnerable groups such as elderly individuals and children under five years old [2].

If not treated promptly, pneumonia can be life-threatening and result in acute respiratory distress [3]. X-rays, especially chest radiographs, offer a non-invasive, cost-effective, and widely accessible method to identify pneumonia [4]. Shoar and Musher [5] emphasized that community-acquired pneumonia remains a critical cause of mortality, underscoring the need for fast and accurate diagnosis methods. Environmental pollutants also elevate the risk and severity of respiratory diseases like pneumonia [6].

In this context, deep learning techniques—specifically Convolutional Neural Networks (CNNs)—have been increasingly adopted in medical image analysis for automated disease detection [7]. CNNs have demonstrated remarkable performance in analysing chest X-ray images to detect pneumonia and COVID-19 [8]. These models can automatically learn and extract key image features, reducing the dependency on manual diagnosis [9].
Optimization techniques such as standard deviation calculation have also been applied to improve CNN classification performance in pneumonia detection tasks [10]. CNNs' success in broader computer vision fields, including social media analysis, validates their flexibility and robustness in image classification [11].

2. METHODS

2.1. Convolutional Neural Network (CNN)

CNN is a specialized deep learning model designed for processing grid-like data such as images. It consists of three main types of layers: convolutional layers, pooling layers, and fully connected layers. These layers operate in a sequence to learn spatial hierarchies of features from input images [12].

Several well-known CNN architectures include LeNet, ZFNet, VGGNet, GoogleNet, and ResNet. Each provides improvements in depth, computation, and feature extraction capabilities [13]. CNNs are particularly effective in medical image classification tasks, including pneumonia detection in chest radiographs, due to their ability to learn from raw pixel input without needing hand-crafted features [14].

The training process of CNN involves passing image data through multiple layers, applying filters, and using activation functions like ReLU to introduce non-linearity. Pooling operations help reduce the spatial dimensions of the data, and the final classification is performed using fully connected layers and softmax activation [15]. **Figure 1** shows the structure of a Convolutional Neural Network (CNN).

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

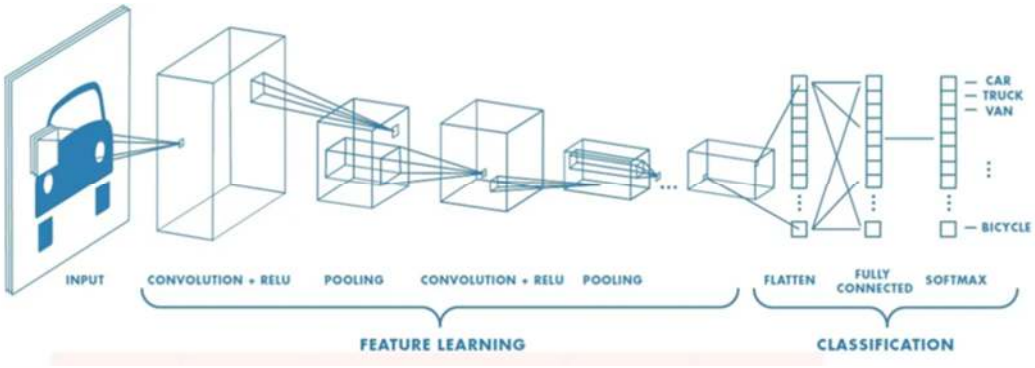


Figure 1. Convolutional Neural Network Layers

2.2. AlexNet Architecture

AlexNet is a deep CNN architecture that popularized deep learning by winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It consists of five convolutional layers followed by three fully connected layers and incorporates techniques such as ReLU activation, dropout regularization, and max-pooling [16].

For pneumonia detection, AlexNet is used with transfer learning, where pre-trained weights (from ImageNet) are fine-tuned to classify chest X-ray images [17]. This technique significantly improves model accuracy and training efficiency, especially when working with limited datasets [18].

To optimize the training process, the Adam optimizer is commonly applied. Adam combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp. It computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients [19].

In this study, Adam was used with a learning rate of 0.0001 and CrossEntropyLoss as the loss function, which helped improve convergence stability and classification performance [20]. **Figure 2** is a representation of the AlexNet architecture.

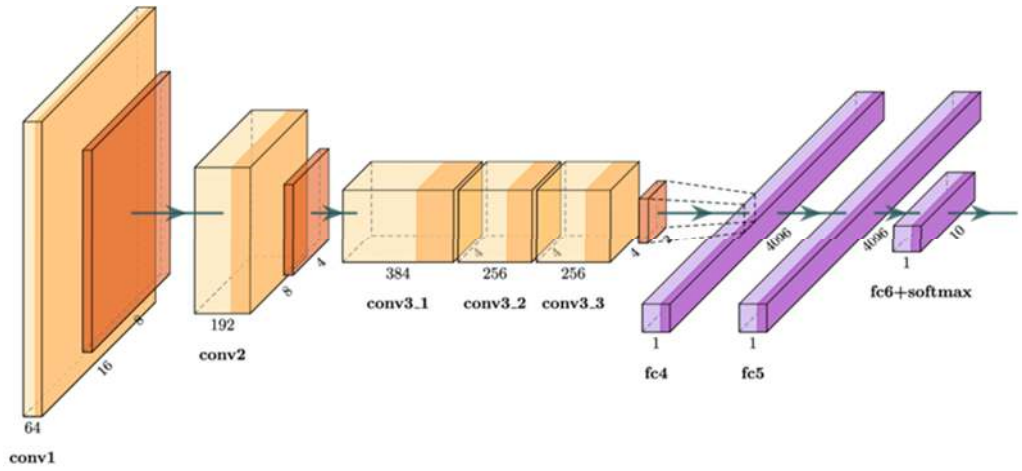


Figure 2. AlexNet Architecture

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

2.3. Research Flow

The first step in this investigation is gathering a dataset of chest X-rays, which contains both normal radiographs and cases of pneumonia. Two kinds of patients—those with pneumonia and those in normal conditions—are represented in the dataset. Following the collection of the dataset, the photos are pre-processed before being entered into the model. Preprocessing includes dividing the dataset into training and testing sets, which are used to train and assess the model's performance, and normalizing pixel intensity to make sure values fall within a suitable range for the model. The AlexNet architecture will then be used to create a Convolutional Neural Network (CNN) model. **Figure 3** shows a summary of the research workflow.

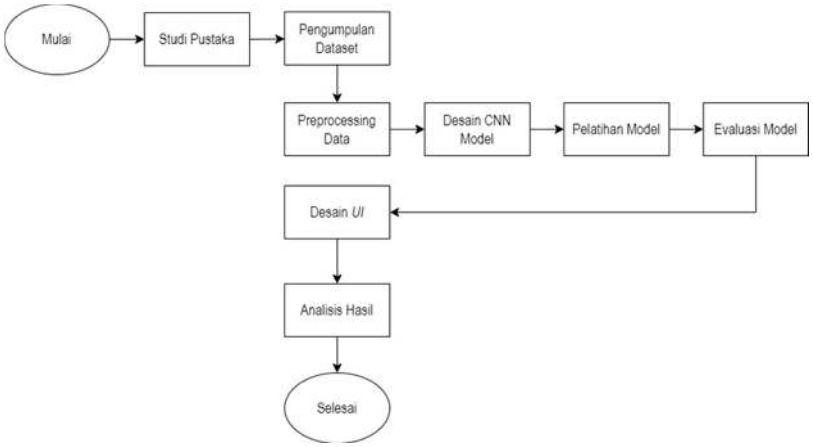


Figure 3. Research Flow

2.4. Research Object

The goal of this research is to use a Convolutional Neural Network (CNN) with the AlexNet architecture to detect pneumonia in chest X-ray pictures. A chest X-ray dataset from the Kaggle platform, comprising 2,804 X-ray images divided into two primary categories—normal and pneumonia—is used in this study. 25% of the dataset is used for testing, and the remaining 75% is used for training.

Based on earlier studies by Sedhy and associates in 2023, titled Deep Learning for the Detection of COVID-19, Pneumonia, and Tuberculosis in Chest X-ray Images, the dataset size was established. Employing CNN's AlexNet Architecture and 2,304 chest X-ray pictures. The study recommended expanding the dataset size for subsequent studies to attain better performance metrics and increased accuracy. By using a statistical method that adheres to the Central Limit Theorem (CLT), 500 more photos can improve the model's performance and reliability if the sample size is large enough.

2.5. Research Object

The UI design will include an input element where users can select and upload X-ray images from their devices. Once the image is uploaded, the system will automatically run the

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

trained Convolutional Neural Network (CNN) model to detect pneumonia. The prediction results will be displayed as text, indicating whether the image represents a normal condition or pneumonia. The visual layout of the user interface will be straightforward to use, and it will include instructions to help users at every stage of the prediction process. The main objective of this user interface design is to guarantee pneumonia detection efficiency and convenience of use, making it available to researchers and medical professionals alike. **Figure 4** shows the user interface design.

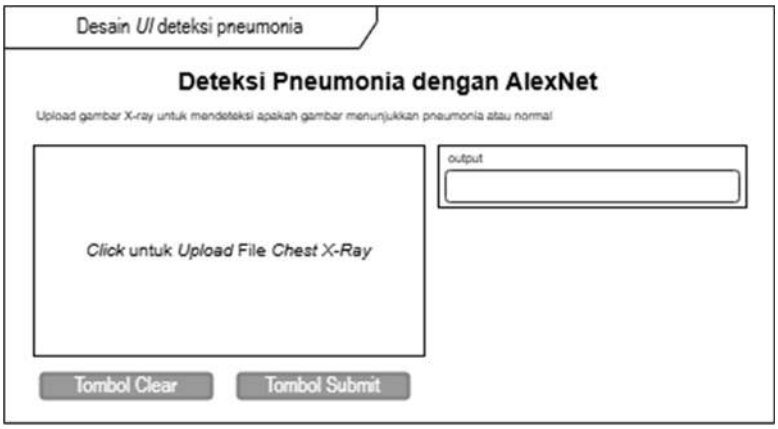
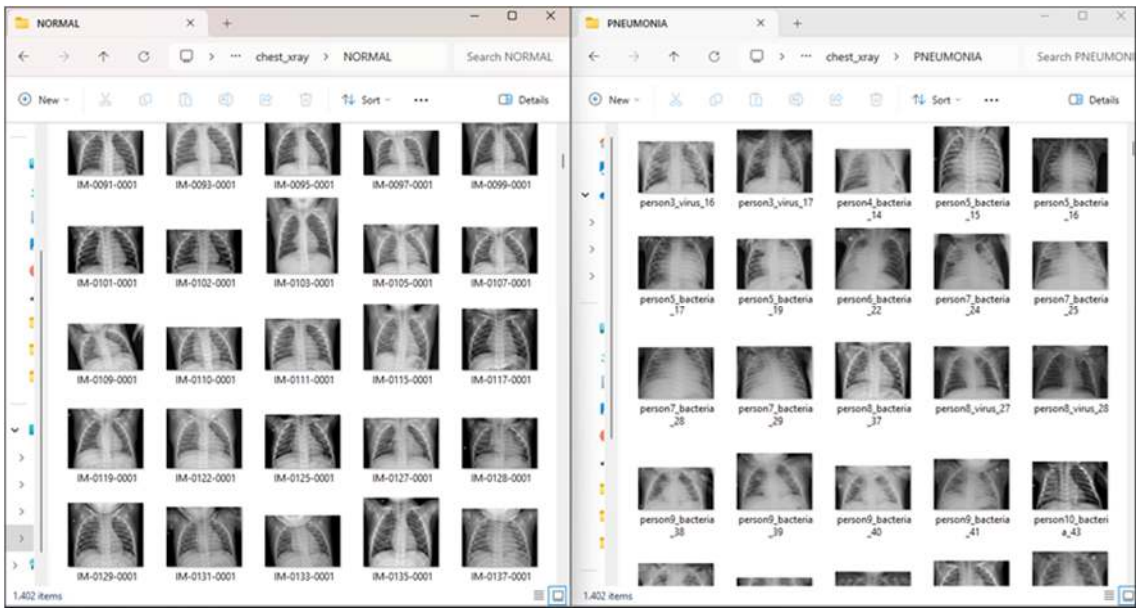


Figure 4. User Interface Design

3. RESULTS AND DISCUSSIONS

3.1. Dataset Description

Daniel S. Kermany's study, Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning, was the first to exploit this dataset. There are 1,403 JPEG images in each of the two classes of chest X-ray images in the dataset: pneumonias and normal images. **Figure 5** displays the images from the chest X-ray collection.



DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

Figure 5. Chest X-Ray Dataset

3.2. Preprocessing Data

The first step in preparing the X-ray dataset is to resize the images to 224x224 pixels to comply with the AlexNet model's input specifications. To enable the model to process the data numerically, the photos are resized and then transformed into tensor format using PyTorch's ToTensor function. To guarantee uniform pixel distribution, normalization is next applied using the mean and standard deviation values from the ImageNet dataset ([0.485, 0.456, 0.406] for the mean and [0.229, 0.224, 0.225] for the standard deviation). Data augmentation techniques like random rotations between -10 and 10 degrees and random horizontal flips are used to increase data diversity and avoid overfitting, which improves the model's capacity to identify various visual patterns.

Following the transformation, the dataset is divided into 75% training data and 25% testing data to make sure the model is tested on unseen images. PyTorch's DataLoader function is used to process data efficiently in batches, with the training set's images being shuffled in each epoch to expose the model to a variety of variations, and the testing set maintaining a fixed order for consistent evaluation. Each batch comprises 32 images, optimizing the training and testing process for better performance.

3.3. Design CNN Models

The model used to detect pneumonia from X-ray images is AlexNet, a Convolutional Neural Network (CNN) architecture. The model design process includes initialization and modification of the final layer for binary classification. PyTorch is used as the primary library for building the model, as it provides various functions for constructing fully connected layers and initializing weights. Additionally, Torchvision, a supporting PyTorch library, simplifies image data processing and provides pretrained models, including AlexNet.

Using the AlexNet_Weights class, which leverages previously trained weights to improve accuracy, the model is initialized with pretrained weights from the ImageNet dataset. AlexNet's classification layer is initialized and then adjusted to the binary classification task, allowing it to differentiate between images of pneumonia and normal X-rays. Two neurons, which stand in for the two categories, are used in place of the final layer in this study, even though AlexNet is by default built for 1000-class classification. After this adjustment, the Xavier initialization method is used to initialize the final layer weights, which speeds up training and preserves consistent learning performance. **Figure 6** shows the CNN AlexNet model's output.

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

```
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace=True)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace=True)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace=True)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
  (classifier): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace=True)
    (3): Dropout(p=0.5, inplace=False)
    (4): Linear(in_features=4096, out_features=4096, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=4096, out_features=2, bias=True)
  )
```

Figure 6. CNN AlexNet Model Output

Several essential layers make up the model architecture, such as MaxPool to lower the dimensionality of the data, ReLU activation for non-linearity, and convolutional layers. Multiple fully connected layers, adaptive average pooling, and dropout to avoid overfitting are all incorporated into the final layers, which result in an output with two neurons that corresponds to the binary classification problem.

3.4. Model Training

Prediction errors in pneumonia classification are measured during the AlexNet model's training process using the CrossEntropyLoss function, and model weights are adaptively updated using the Adam optimizer, which has a learning rate of 0.0001. Multiple batches of chest X-ray pictures processed by the model make up each epoch throughout the training process. To provide predictions, the model does a forward pass. CrossEntropyLoss is then used to compare the predictions with the real labels. To improve prediction accuracy, the Adam optimizer uses the gradients of the loss function, which are then computed in a backward pass. The running loss value is recorded at each epoch as a measure of model performance.

To observe patterns in loss reduction, the CNN model's training using the AlexNet architecture was assessed over 20 and 30 epochs. The model's capacity to identify patterns in training data improved as the mean loss steadily dropped from 0.3434 to 0.0448 after 20 epochs. A continuous decrease in mean loss from 0.0417 to 0.0019 was also observed after 30 epochs of training, indicating a notable improvement in prediction accuracy. The overall loss decrease indicates that the model became more and more tuned for pneumonia classification from X-ray pictures, with gradually lower error rates, despite slight oscillations in certain epochs. **Figure 7** shows the outcome of the mean loss.

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

```
Mulai pelatihan di cpu
Epoch 1/10 dimulai
Epoch [1/10], Rata-rata Loss: 0.3143
Epoch 2/10 dimulai
Epoch [2/10], Rata-rata Loss: 0.1610
Epoch 3/10 dimulai
Epoch [3/10], Rata-rata Loss: 0.1345
Epoch 4/10 dimulai
Epoch [4/10], Rata-rata Loss: 0.1123
Epoch 5/10 dimulai
Epoch [5/10], Rata-rata Loss: 0.1069
Epoch 6/10 dimulai
Epoch [6/10], Rata-rata Loss: 0.1134
Epoch 7/10 dimulai
Epoch [7/10], Rata-rata Loss: 0.0789
Epoch 8/10 dimulai
Epoch [8/10], Rata-rata Loss: 0.0886
Epoch 9/10 dimulai
Epoch [9/10], Rata-rata Loss: 0.0743
Epoch 10/10 dimulai
Epoch [10/10], Rata-rata Loss: 0.0842
Pelatihan selesai!
```

Figure 7. Output Mean Loss

3.5. Model Evaluation

Based on the test dataset, the AlexNet model's evaluation results show 95.44% accuracy, 99% precision, and 92% recall. The model's efficacy in classifying X-ray images for the detection of pneumonia is demonstrated by its accuracy, which is determined by dividing the number of correct predictions by the total test data. While the comparatively high recall shows the model's strong ability to detect the majority of pneumonia cases, the high precision suggests that the model rarely generates false positive predictions. The evaluation results of the AlexNet model can be seen in **Figure 8**.

```
Correct so far: 640/672
Processing batch 22/22
Predictions: tensor([1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1,
        1, 1, 0, 0, 1])
Correct so far: 669/701
Accuracy on test set: 95.44%
Precision: 0.99
Recall: 0.92
```

Figure 8. Evaluation Results of the AlexNet Model

Because the trainisng and test datasets are split randomly, the evaluation findings could change every time the model is run again. Furthermore, because Google Colab is used as a cloud computing platform in this study, the model must be run again for every new session, which may result in differences in recall, accuracy, and precision. However, the model's ability to identify pneumonia from X-ray pictures is consistently dependable.

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

3.6. User Interface Display

The interactive interface built with Gradio is used to facilitate pneumonia detection using the AlexNet Using the AlexNet model, the interactive Gradio-built interface makes it easier to detect pneumonia by letting users upload X-ray images and get immediate prediction results. **Figure 9** shows the user interface display. There are three possible outputs for the detection results; **Figure 10** displays the Normal output. **Figure 11** displays the detection result for pneumonia. As seen in **Figure 12**, if the input is not a legitimate chest X-ray, the image cannot be processed.

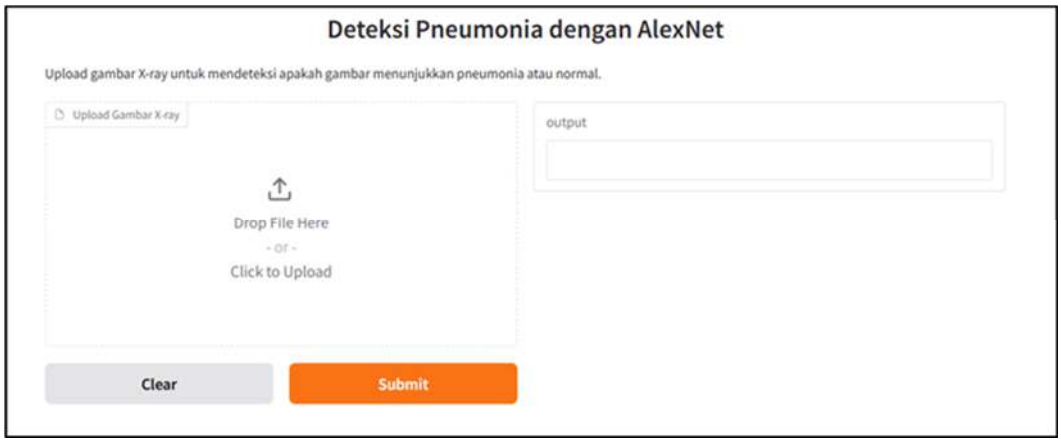


Figure 9. User Interface Display



Figure 10. UI Display for Detection Result as Normal



Figure 11. UI Display for Detection Result as Pneumonia



DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

Figure 12. UI Display of Detection Result for Invalid Input

X-ray data from the archives of RSUD Chasan Boesari Ternate were used to verify the model's ability to accurately identify pneumonia in medical photos. The testing X-ray pictures are shown in **Figure 13**.

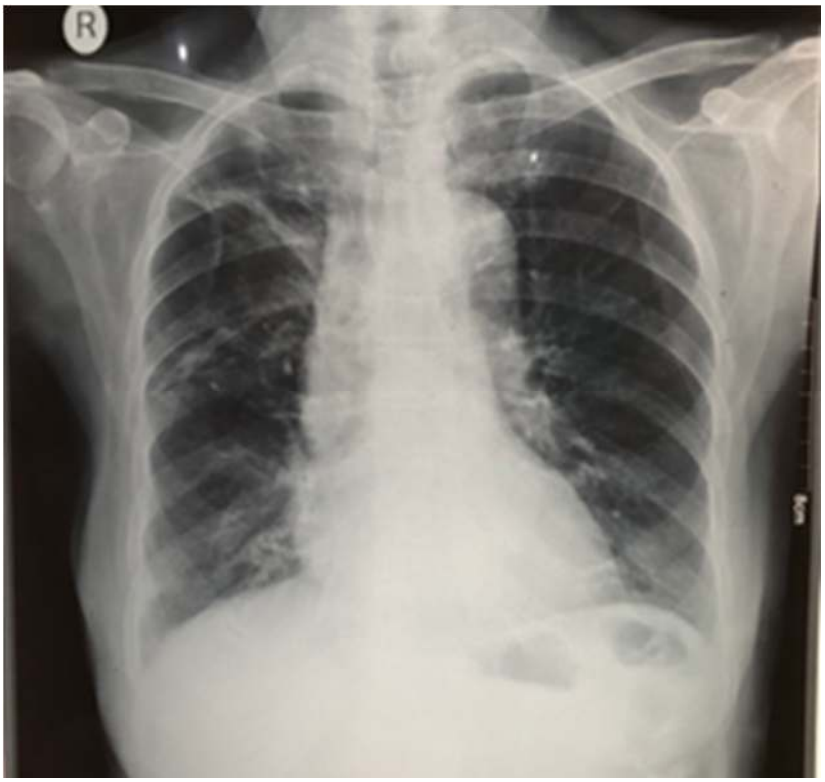


Figure 13. X-Ray Pneumonia District Hospital

4. CONCLUSION

This study evaluates the AlexNet CNN architecture for detecting pneumonia from chest X-ray images through several key stages. The process begins with the collection and preprocessing of 2,806 X-ray images, categorized as normal or pneumonia, using techniques such as resizing, normalization, and augmentation to enhance data quality. The AlexNet model was modified with pretrained weights from ImageNet to adapt it for binary classification. The model training was conducted using the Adam optimizer and CrossEntropyLoss, enabling the model to effectively recognize patterns in the data. Evaluation results showed an accuracy of 95.44%, precision of 99%, and recall of 94%, although slight variations may occur due to the nature of the Google Colab platform. All things considered, this study shows that AlexNet has great promise as a trustworthy tool for X-ray imaging-based pneumonia diagnosis.

5. AUTHORS' NOTE

For future research, it is recommended to use a GPU for the model training process. This study utilized a CPU, which resulted in a longer runtime. By using a GPU, the training process is expected to be faster and more efficient, allowing for experiments with more complex models or larger datasets.

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

6. REFERENCES

[1] Lin, J., & Irsyad, H. (2021). Classification of Pneumonia in Lung X-Ray Images Using GLCM and LVQ. *Jurnal Algoritme*, 1(2), 184-194. ISSN 2775-8796. https://doi.org/10.35957/algoritme.v1i2.897.

[2] Nuraeni, M. (2020). Prediction of Pneumonia Using Big Data, Deep Learning, and Machine Learning Techniques. *International Research Journal of Modernization in Engineering Technology and Science*. https://doi.org/10.56726/irjmets29620.

[3] Yopento, J., Ernawati, & Coastera, F. F. (2022). Identification of Pneumonia in Lung X-Ray Images Using Convolutional Neural Network (CNN) Method Based on Sobel Feature Extraction. *Rekursif: Jurnal Informatika*, 10(1), 40-47. ISSN 2777-0427. Unib Press. https://doi.org/10.33369/rekursif.v10i1.17247.

[4] Yugaswara, H., Jounio, R., & Fathurahman, M. (2022). Pneumonia Detection in COVID-19 Patients Based on Chest X-Ray Images Using Transfer Learning Method. *Media Elektrika*, 15(2), 95. ISSN 2579-972X. LPPM Universitas Muhammadiyah Semarang. https://doi.org/10.26714/me.v15i2.10731.

[5] Shoar, S., & Musher, D. M. (2020). Etiology of Community-Acquired Pneumonia in Adults: A Systematic Review. *Pneumonia*, 12(1). ISSN 2200-6133. Springer Science and Business Media LLC. https://doi.org/10.1186/s41479-020-00074-3.

[6] Abdullah, I. A. A., Sadikin, R., Rianto, Y., & Priansyah, D. (2022). Segmentation of Visceral Fat and Lung Cancer Using Deep Learning. *Buletin Poltanesa*, 23(2). ISSN 2614-8374. Politeknik Pertanian Negeri Samarinda.

[7] Wilyani, F., Arif, Q. N., & Aslimar, F. (2024). Introduction to Basic Python Programming Using Google Colaboratory. *Jurnal Pelayanan dan Pengabdian Masyarakat Indonesia*, 3(1), 08–14. https://doi.org/10.55606/jppmi.v3i1.1087.

[8] Modi, A. R., & Kovacs, C. S. (2020). Hospital-Acquired and Ventilator-Associated Pneumonia: Diagnosis, Management, and Prevention. *Cleveland Clinic Journal of Medicine*, 87(10), 633-639. ISSN 0891-1150. https://doi.org/10.3949/ccjm.87a.19117.

[9] Directorate General of Higher Education, Research, and Development. (2018). Accredited Sinta Rank 4 (Vol. 3, Issue 1). Pneumonia Detection Using Deep Learning Based on Convolutional Neural Network (CNN) Model. *International Research Journal of Modernization in Engineering Technology and Science*. https://doi.org/10.56726/irjmets32669.

[10] Sanjaya, U. P., Alawi, Z., Zayn, A. R., & Dirgantoro, G. P. (2023). Optimization of Convolutional Neural Network with Standard Deviation for Pneumonia Classification on X-ray Images. *Generation Journal*, 7(3), 40-47.

[11] Kholik, A. (2021). Classification Using Convolutional Neural Network (CNN) on Instagram Page Screenshots. *Jurnal Data Mining dan Sistem Informasi*, 2(2), 10. ISSN 2745-8458. Universitas Teknokrat Indonesia. https://doi.org/10.33365/jdmsi.v2i2.1345.

[12] Rahman, T., Chowdhury, M. E. H., Khandakar, A., & Islam, K. R. (2020). Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray. Applied Sciences, 10(9), 3233. https://doi.org/10.3390/app10093233

[13] Vinoth, R., & Subalakshmi, S. (2023). Pneumonia detection from chest X-ray using AlexNet image classification technique. 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS). https://doi.org/10.1109/ICICCS58233.2023.10142405

[14] Jain, R., Nagrath, P., Kataria, G., Kaushik, V. S., & Hemanth, D. J. (2020). Pneumonia detection in chest X-ray images using convolutional neural networks and transfer

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

learning. Measurement, 165, 108046. https://doi.org/10.1016/j.measurement.2020.108046

[15] Alapat, D. J., Menon, M. V., & Ashok, S. (2022). A review on detection of pneumonia in chest X-ray images using neural networks. Journal of Biomedical Physics and Engineering, 12(5), 551–564.

[16] Salih, S. Q., Abdulla, H. K., & Ahmed, Z. S. (2020). Modified AlexNet convolution neural network for COVID-19 detection using chest X-ray images. Kurdistan Journal of Applied Research, 5(1), 45–51.

[17] Jamil, H., Mohamed, M., & Joret, A. (2021). Chest X-ray pneumonia classification using AlexNet. Evolution in Electrical and Electronic Engineering, 2(1), 294–303.

[18] Jawarneh, M., Kalash, H. S., & Al Amri, M. (2023). Enhancing pneumonia detection in chest X-rays through transfer learning with AlexNet and adversarial training. IRE Journals, 7(8), 234–239.

[19] Athar, A., Asif, R. N., Saleem, M., & Munir, S. (2023). Improving pneumonia detection in chest X-rays using transfer learning approach (AlexNet) and adversarial training. IEEE Conference Proceedings. https://doi.org/10.1109/ICBATS57792.2023.10111193

[20] Jenefa, J., & Vetriveeran, D. (2025). A comparative analysis of AlexNet and ResNet for pneumonia detection. In Monitoring Using IoT Systems. https://doi.org/10.1002/9781394270392.ch11

DOI: https://doi.org/10.17509/coelite.v4i1.80881
p- ISSN 2829-4157 e- ISSN 2829-4149

Journal of Computer Engineering, Electronics and Information Technology (COELITE)

Journal homepage: https://ejournal.upi.edu/index.php/COELITE

Hybrid Explainable AI Approach for DNA Sequence Classification: Feature Importance, Permutation Importance, and Local Explanations with LIME

Nur Alamsyah[1], Budiman[2], Reni Nursyanti[3], Elia Setiana[4], Venia Restreva Danestiara[5], Titan Parama Yoga[6]*

[1]Department of Technology and Informatics, Universitas Informatika Dan Bisnis Indonesia, Indonesia
Correspondence E-mail: nuralamsyah@unibi.ac.id

ABSTRACT

Understanding the contribution of features in DNA sequence classification is crucial for enhancing model interpretability and reliability. This study proposes a Hybrid Explainable AI (XAI) approach that integrates Feature Importance (FI), Permutation Importance (PI), and Local Interpretable Model-Agnostic Explanations (LIME) to analyse the most influential features in a Random Forest classifier. FI is utilized to determine the most significant features contributing to the model, while PI validates their impact by assessing performance changes when features are shuffled. Additionally, LIME provides local explanations, offering insight into how specific feature values affect classification decisions. Experimental results on a publicly available DNA sequence dataset reveal a strong correlation between FI and PI rankings, validating the stability of key features such as A84, A89, and A92. LIME further enhances interpretability by highlighting individual instance contributions, reinforcing the relevance of specific nucleotide positions in sequence classification. This hybrid approach provides a more comprehensive understanding of feature importance, improving trust and transparency in DNA sequence classification models.

ARTICLE INFO

Article History:
Submitted/Received 11 Mar 2025
First Revised 18 Mar 2025
Accepted 28 Mar 2025
First Available online 1 Apr 2025
Publication Date 1 Apr 2025

Keyword:
Explainable AI,
Feature Importance,
Permutation Importance,
LIME,
DNA Classification.

© 2025 Universitas Pendidikan Indonesia

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

1. INTRODUCTION

The classification of DNA sequences plays a crucial role in bioinformatics, particularly in identifying genetic variations, disease-associated mutations, and evolutionary relationships [1]. With the growing availability of genomic data, machine learning techniques have been increasingly applied to improve classification accuracy [2]. However, while machine learning models such as Random Forest (RF), Support Vector Machines (SVM), and Deep Learning architectures have shown remarkable performance, their black-box nature limits interpretability, making it difficult for researchers to understand the factors influencing predictions [3]. This lack of transparency poses challenges in critical applications such as genetic disease prediction and mutation impact analysis, where interpretability is essential for validating model decisions [4].

To address this issue, Explainable Artificial Intelligence (XAI) has emerged as a framework to interpret complex machine learning models [5]. Several XAI techniques have been introduced to provide insights into model behaviour, including Feature Importance (FI), Permutation Importance (PI), and Local Interpretable Model-Agnostic Explanations (LIME) [6]. FI is a widely used approach that measures the contribution of each feature based on decision trees' splits, providing a global explanation of feature relevance [7]. However, FI may be biased toward correlated features, necessitating Permutation Importance (PI) as a validation method. PI quantifies feature impact by measuring performance degradation when a feature's values are randomly shuffled, offering a more robust validation mechanism for identifying key features [8]. While both FI and PI provide global feature explanations, they do not provide instance-level interpretability.

To complement these global explanations, LIME (Local Interpretable Model-Agnostic Explanations) has been introduced to provide localized feature insights by perturbing input data and fitting a surrogate model to approximate local decision boundaries [9]. LIME has been effectively applied in biomedical research to understand mutation impact, disease classification, and sequence pattern contributions [10]. However, its application in DNA sequence classification remains limited, and no previous studies have integrated FI, PI, and LIME into a hybrid explainability framework for genetic data analysis.

Several studies have focused on applying individual explainability techniques to machine learning models for genomic analysis. Zhou *et al.* [11] and Zubair *et al.* [12] explored FI in genomic classification, demonstrating its effectiveness in ranking influential features. Cantor *et al.* [13] highlighted PI's ability to validate feature significance, making it more reliable for high-dimensional biological datasets. Meanwhile, Labory *et al.* [14] introduced LIME as a method for explaining predictions at an instance level, which was later adapted in genomic research for mutation-based classification Puiu *et al.*, [15] and genomic sequence interpretation Lee *et al.*[16]. However, the existing literature lacks a comprehensive hybrid approach that integrates FI, PI, and LIME into a unified framework.

This study proposes a Hybrid Explainable AI (XAI) approach for DNA sequence classification by integrating Feature Importance (FI), Permutation Importance (PI), and LIME. The proposed framework aims to:

1. Determine the most influential genomic features using FI,
2. Validate feature significance through PI, and
3. Provide local instance-based explanations with LIME.

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

By combining these three explainability techniques, this research enhances interpretability, robustness, and transparency in machine learning-based DNA classification, enabling a more reliable understanding of feature contributions and their impact on model predictions. The results of this hybrid approach will not only improve model trustworthiness but also contribute to advancing interpretable AI solutions in bioinformatics and genomic research.

2. METHODS

This section describes the methodology used in this study, which follows a structured pipeline consisting of data collection, preprocessing, model training, and explainability analysis. The workflow begins with data collection, where a DNA sequence dataset is obtained and prepared for machine learning. The data preprocessing phase includes feature-target separation, normalization, and data splitting to ensure optimal model performance. The processed data is then utilized in the model training phase, where a machine learning classifier is trained to predict DNA sequence classifications.

Following model training, the explainability analysis is performed using a hybrid Explainable AI (XAI) framework comprising Feature Importance (FI), Permutation Importance (PI), and Local Interpretable Model-Agnostic Explanations (LIME). FI identifies globally important features by assessing their impact on model decisions, while PI validates the robustness of these feature contributions by measuring performance degradation when features are permuted. Lastly, LIME provides instance-level insights, explaining how individual features contribute to specific predictions.

The methodology is visualized in **Figure 1**, which outlines the main steps involved in this research. The subsequent subsections provide a detailed description of each stage, starting with data collection.

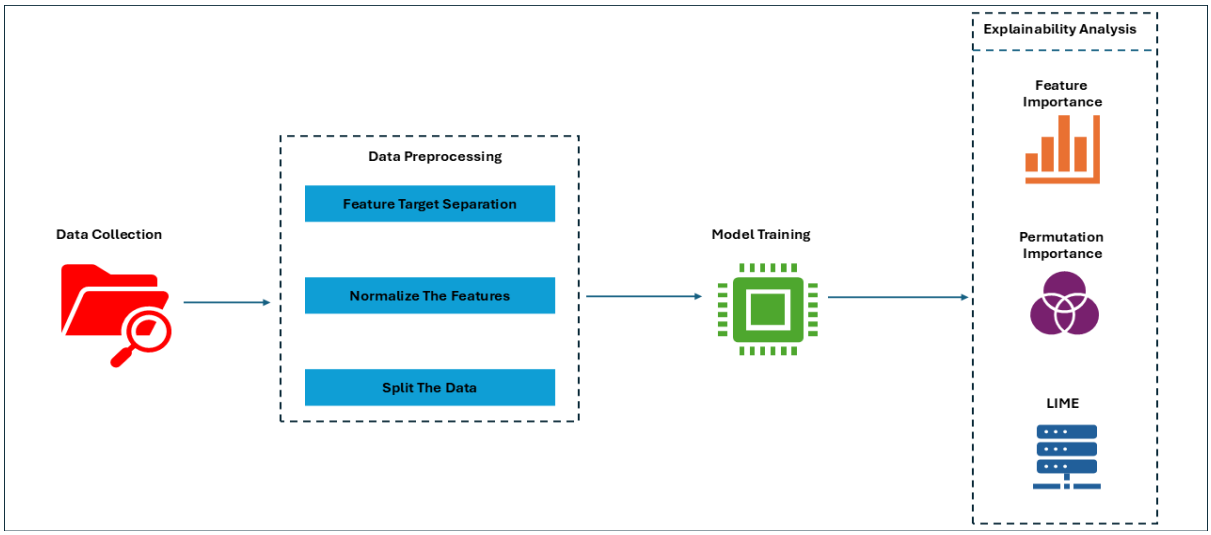


Figure 1. Propose Method.

2.1. Data Collection

The dataset used in this study was obtained from Kaggle, containing 3,187 DNA sequences. Each instance in the dataset represents a DNA segment, with features encoding nucleotide

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

presence in specific positions. The dataset consists of 180 feature columns representing nucleotide encoding and one target column (class), which categorizes DNA sequences into distinct classes.

To facilitate the explainability analysis, we process the dataset by encoding the nucleotide sequences into numerical values. Each feature represents a specific nucleotide position in the sequence, encoded as a binary value. The target column (class) denotes the classification of the DNA sequence into different biological categories. A summary of the dataset features is provided in **Table 1.**

Table 1. Feature Description of the DNA Dataset

Feature	Description
A0 - A179	Encoded nucleotide presence (binary)
class	DNA sequence classification label

This dataset serves as the foundation for training a classification model and conducting explainability analysis using Feature Importance (FI), Permutation Importance (PI), and Local Interpretable Model-agnostic Explanations (LIME).

2.2. Data Preprocessing

Data preprocessing is a crucial step to ensure that the dataset is properly formatted and optimized for training machine learning models [17]. In this study, the preprocessing phase consists of three main steps: Feature Target Separation, Feature Normalization, and Data Splitting. The first step, Feature Target Separation, involves isolating the features from the target variable. All DNA sequence attributes serve as input features, while the classification label represents the target variable. This separation ensures a clear distinction between input and output data for the machine learning model.

Next, Feature Normalization is performed to standardize the scale of the features. Min-Max Scaling is applied to transform all feature values into a normalized range between 0 and 1, ensuring consistent feature representation. This normalization process enhances model stability and improves training convergence. The final step, Data Splitting, partitions the dataset into three subsets: Training Set, Validation Set, and Test Set. The training set is used to train the model, the validation set is used for hyperparameter tuning, and the test set evaluates model performance on unseen data. The data is split using a 70% training, 15% validation, and 15% testing ratio while maintaining class distribution using stratified sampling. Through this preprocessing pipeline, the dataset is structured and optimized for training, ensuring better model generalization and improved classification performance.

2.3. Model Training

The model training phase involves developing a classification model to predict DNA sequence categories based on extracted features. In this study, a Random Forest Classifier was chosen due to its robustness and capability to handle high-dimensional data effectively [18]. The training process follows these steps:

1. Model Initialization: A Random Forest classifier is initialized with optimized hyperparameters.
2. Model Training: The classifier is trained using the processed dataset.
3. Performance Evaluation: The trained model is assessed using standard classification metrics.

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

A Random Forest Classifier consists of multiple decision trees, where each tree contributes to the final prediction through majority voting. The model's prediction function can be mathematically expressed as:

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} \quad f_i(X) \tag{1}$$

Where $(\hat{y})$ Is the final predicted class? (N) Is the number of decision trees and $(f_i(X))$ Is the prediction from the (i)-th decision tree. The training process aims to minimize the classification error by optimizing the splitting criteria in each tree, typically using the Gini impurity:

$$G = 1 - \sum_{i=1}^{C} \quad p_i^2 \tag{2}$$

Where (G) Is the Gini impurity, (C) Is the number of classes and (p_i) Is the probability of a sample belonging to a class(i).

2.4. Explainability Analysis

To enhance the interpretability of the classification model, we employ a comprehensive Explainability Analysis framework that integrates Feature Importance (FI), Permutation Importance (PI), and Local Interpretable Model-Agnostic Explanations (LIME). These techniques provide a deeper understanding of how features contribute to the model's decision-making process [19].

Feature Importance (FI)

Feature Importance quantifies the contribution of each feature to the model's predictive performance by measuring the reduction in impurity across decision trees [20]. The importance of a feature (f_i) A decision tree model is given by:

$$FI(f_i) = \sum_{t \in T} \quad I_t \cdot w_t \tag{3}$$

Where (T) represents all nodes in the decision trees where (f_i) is used, (I_t) Is the impurity reduction at the node (t) and (w_t) Is the weighted number of samples that reach the node (t).

Permutation Importance (PI)

Permutation Importance is a model-agnostic technique that validates feature relevance by randomly shuffling individual features and measuring the corresponding decrease in model accuracy. The PI score for a feature (f_i) Is computed as:

$$PI(f_i) = \frac{1}{N}\sum_{j=1}^{N} \quad \left[A - A_{f_i}^{(j)}\right] \tag{4}$$

Where (A) Is the baseline accuracy of the trained model? $(A_{f_i}^{(j)})$Is the accuracy after randomly permuting the feature? (f_i) and (N) Is the number of iterations. Higher PI scores indicate a greater impact on classification performance when the feature is shuffled, thus validating its importance.

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

Local Interpretable Model-Agnostic Explanations (LIME)

LIME generates local surrogate models to approximate the behaviour of the classifier for individual predictions. It perturbs the input data and learns a linear approximation around a given instance.(x). The explanation model is given by:

$$\hat{f}(x) = arg\ arg\ L\ (f, g, \pi_x) + \Omega(g) \quad (5)$$

Where (g) is the interpretable model (e.g., a linear regression model), $(L(f, g, \pi_x))$Is the loss function ensuring approximations (f) locally, (π_x)represents the proximity measure defining locality and $(\Omega(g))$enforces complexity constraints on (g) For better interpretability.

By integrating FI, PI, and LIME, this explainability framework provides both global feature significance (FI, PI) and local interpretability (LIME), offering a transparent and accountable explanation for DNA sequence classification.

3. RESULTS AND DISCUSSION

3.1. Result

Feature Importance (FI) was employed to determine the most influential features in predicting DNA sequence classifications. The model utilized a Random Forest classifier to rank features based on their contribution to the classification task. **Table 2** presents the top 15 most important features, while **Figure 2** visualizes their relative importance scores. The results indicate that feature A89 holds the highest importance score of 0.091248, followed by A84 and A92, with scores of 0.080648 and 0.064267, respectively. Features A104 and A99 also show notable influence, suggesting their relevance in distinguishing DNA sequence classifications. The remaining features exhibit progressively lower importance scores but still contribute to the model's performance.

These findings provide a global understanding of how different features impact the classification model. However, to validate these results and ensure their robustness, we extend our analysis by incorporating Permutation Importance (PI) in the subsequent section.

Table 2. Top 15 most important features.

Rank	Feature	Importance Score
1	A89	0.091248
2	A84	0.080648
3	A92	0.064267
4	A104	0.061665
5	A99	0.037118
6	A82	0.033560
7	A88	0.016974
8	A83	0.016850
9	A93	0.015391
10	A87	0.015057
11	A95	0.014192
12	A81	0.013620
13	A85	0.013004
14	A96	0.011621

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

15	A86	0.010789

Figure 2 illustrates the Feature Importance distribution, highlighting the relative significance of each feature in the model.

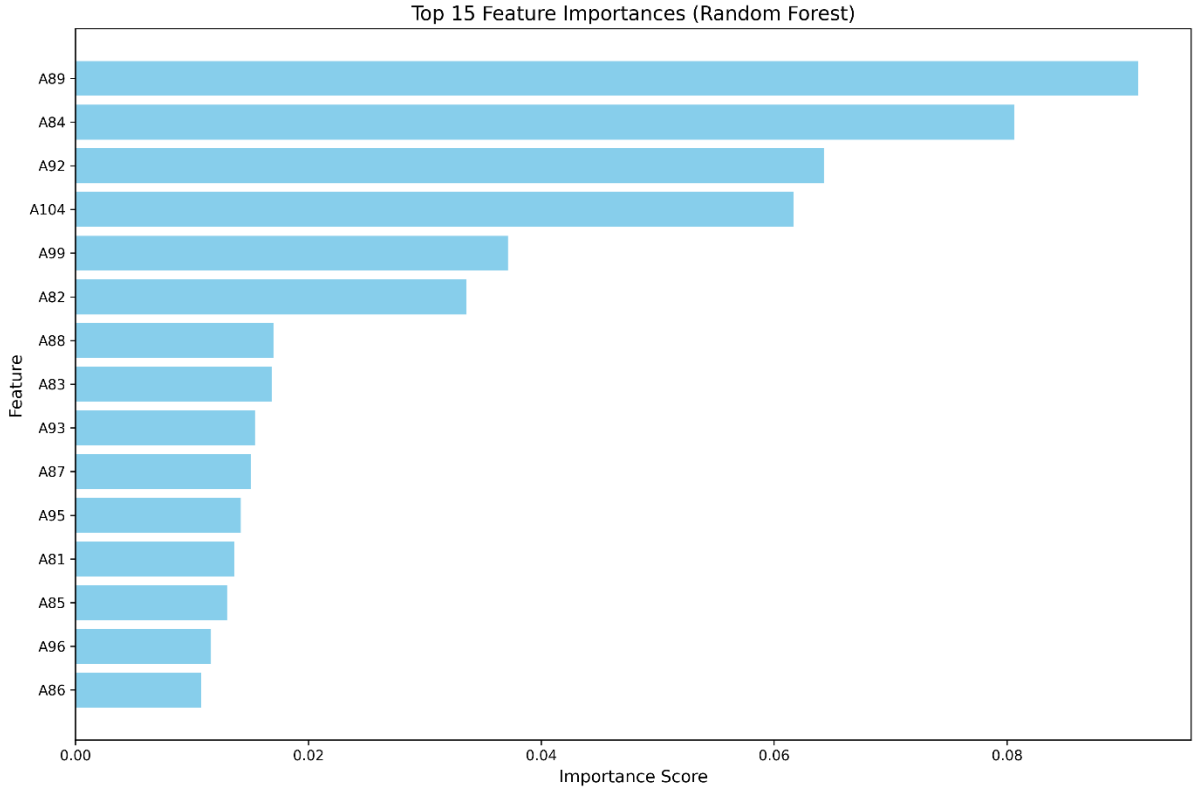


Figure 2. Importance Score

To validate the results obtained from Feature Importance (FI), we employed Permutation Importance (PI) as an additional validation method. PI measures the impact of each feature by shuffling its values and observing the drop in model performance. This method provides a robust verification of the relative importance of features. The results presented in **Table 3** and visualized in Figure 3 confirm the consistency between FI and PI rankings. The top three features remain unchanged, with A84, A89, and A92 maintaining their high importance scores. However, some variations can be observed in lower-ranked features, where A104 and A93 show slightly different rankings. The standard deviation values indicate the stability of feature contributions, with lower deviations implying more consistent impact.

Table 3. Top 15 Permutation Importance Scores.

Rank	Feature	Importance Score	Standard Deviation
1	A84	0.098536	0.009516
2	A89	0.092050	0.013065
3	A92	0.068410	0.007942
4	A104	0.048536	0.009013
5	A93	0.017573	0.004879
6	A99	0.015900	0.005056
7	A95	0.010460	0.005038

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

Rank	Feature	Importance Score	Standard Deviation
8	A83	0.008368	0.004184
9	A94	0.007741	0.003110
10	A87	0.005439	0.002838
11	A96	0.004603	0.002440
12	A74	0.004184	0.001871
13	A103	0.003347	0.003268
14	A85	0.003347	0.002510
15	A88	0.003138	0.003771

Overall, the validation through PI strengthens the reliability of the feature importance findings, confirming that the identified key features significantly contribute to the classification of DNA sequences. **Figure 3** depicts the Permutation Importance analysis, highlighting the top 15 most influential features and their variations.

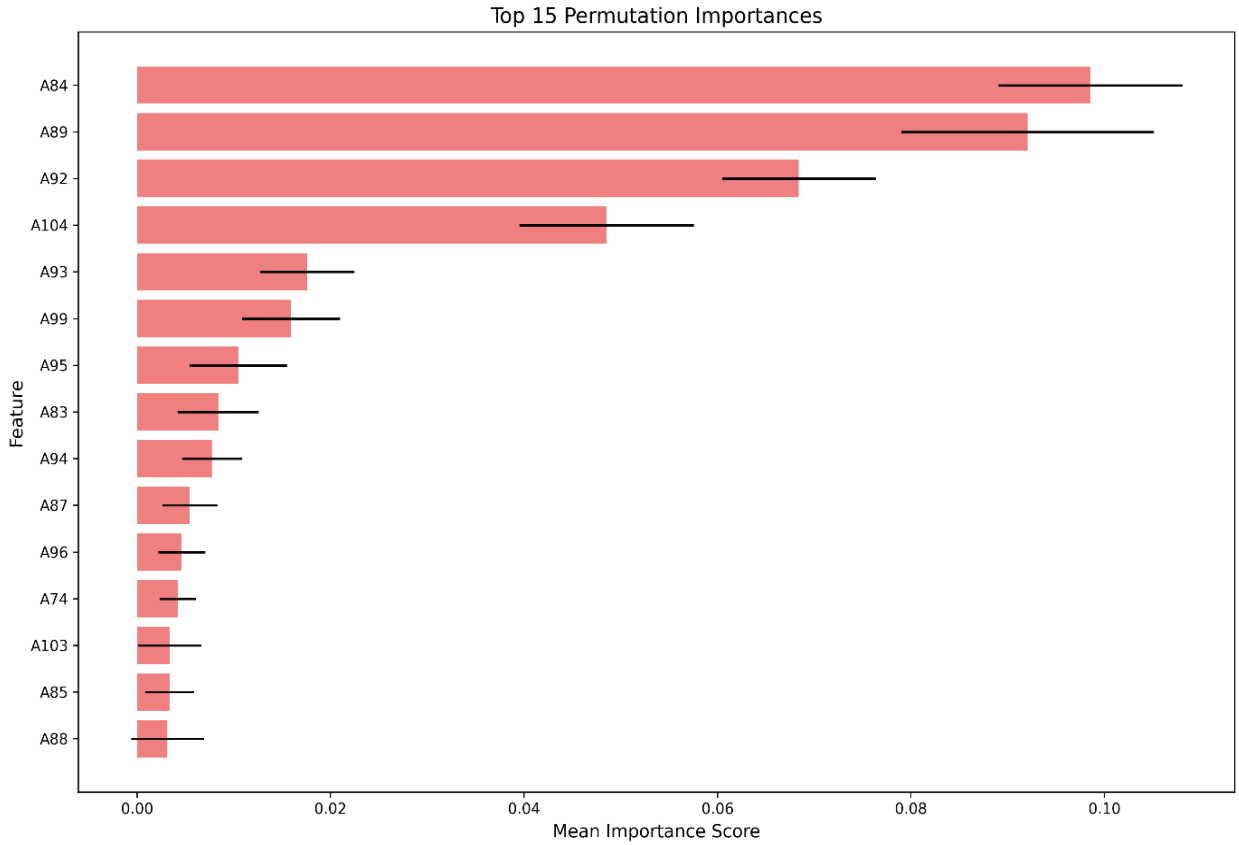


Figure 3. Permutation Importance Score

To complement the Feature Importance (FI) and Permutation Importance (PI) analyses, we conducted a Local Interpretable Model-Agnostic Explanations (LIME) analysis. LIME provides an interpretable, instance-based explanation by approximating the model's decision boundary with a simpler, interpretable model.

Figure 4 presents the local explanation for a single test sample. The horizontal bars represent the contribution of each feature to the classification decision. The red bars indicate

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

negative contributions (reducing the probability of a specific class), while the green bars denote positive contributions (increasing the probability of the predicted class).

- A84 and A89 exhibit the highest negative influence on the classification outcome, indicating their strong contribution to shifting the prediction away from certain DNA classes.
- Conversely, A83 and A82 show significant positive contributions, reinforcing the model's classification decision.
- Other features such as A81, A88, and A86 play minor but noticeable roles in adjusting the classification outcome.

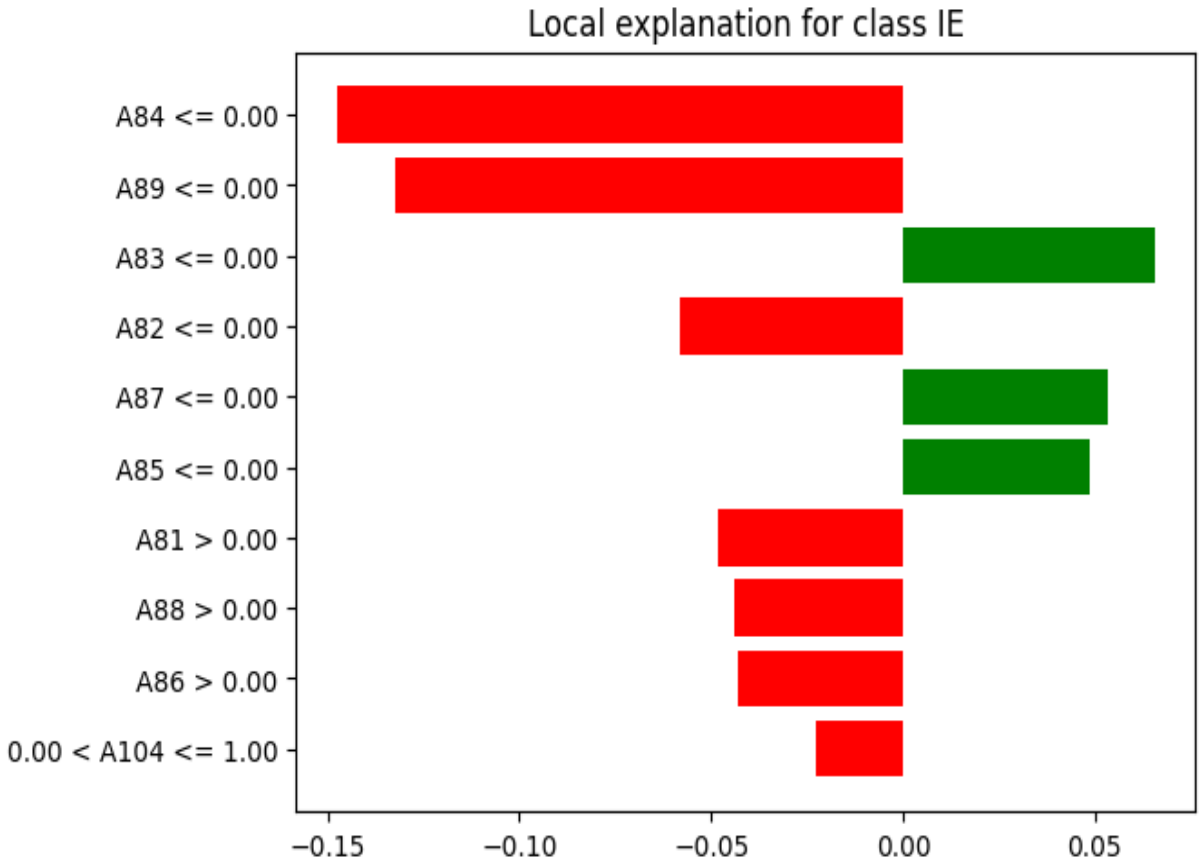


Figure 4. Local Explanation For Class IE

These results further validate the global Feature Importance (FI) and Permutation Importance (PI) findings by providing an instance-based justification for the significance of key features. This combination of global and local explainability techniques ensures a comprehensive understanding of the DNA sequence classification model.

4. CONCLUSION

This study presented a Hybrid Explainable AI Approach for DNA sequence classification, integrating Feature Importance (FI), Permutation Importance (PI), and Local Interpretable Model-Agnostic Explanations (LIME) to enhance the interpretability of machine learning predictions. The results demonstrated that features A84, A89, and A92 were consistently identified as the most significant across multiple explainability techniques. Feature

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

Importance (FI) provided a global understanding of the dominant features influencing model decisions, while Permutation Importance (PI) validated their impact on classification performance. Furthermore, LIME offered localized explanations, allowing a more detailed analysis of individual predictions, which is crucial in understanding how specific DNA sequence characteristics contribute to classification outcomes. The combination of these explainability methods ensures a transparent and interpretable machine learning model, which is essential for applications in genomic research and biological analysis.

Although this approach offers valuable insights, there are several directions for future work. First, exploring more advanced models such as deep neural networks or Transformer-based architectures could improve classification performance while maintaining interpretability. Additionally, integrating domain-specific feature engineering may enhance feature representation and improve model accuracy. Further research can also investigate additional explainability techniques, such as SHAP (Shapley Additive Explanations) and counterfactual analysis, to gain deeper insights into feature contributions and interactions. Expanding the dataset with more diverse DNA sequences can improve the model's generalization ability, making it more robust for practical applications. Finally, applying this explainability framework in genomic medicine, disease prediction, and mutation analysis could extend its impact beyond classification tasks, contributing to advancements in computational biology and precision medicine.

5. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Faculty of Technology and Informatics and the Institute for Research and Community Service (LPPM), Universitas Informatika dan Bisnis Indonesia (UNIBI), for their continuous support and encouragement in conducting this research. Their invaluable assistance and resources have greatly contributed to the successful completion of this study.

6. AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. Authors confirmed that the paper was free of plagiarism. Furthermore, all authors have significantly contributed to the research, including data collection, model development, explainability analysis, and manuscript writing. This research represents original work and has not been submitted or published elsewhere.

7. REFERENCES

[1] D. Bendigeri, L. Sakri, S. Mural, S. Hukkeri, and P. Tayannavar, "Human Genetic-based Disease Identification," in *2024 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, 2024, pp. 486–491.

[2] H. Song *et al.*, "A novel approach utilizing domain adversarial neural networks for the detection and classification of selective sweeps," *Adv. Sci.*, vol. 11, no. 14, p. 2304842, 2024.

[3] F. M. Talaat, A. Aljadani, M. Badawy, and M. Elhosseini, "Toward interpretable credit scoring: integrating explainable artificial intelligence with deep learning for credit card default prediction," *Neural Comput. Appl.*, vol. 36, no. 9, pp. 4847–4865, 2024.

[4] A. Husnain, A. Shiwlani, M. N. Gondal, A. Ahmad, A. Saeed, and others, "Utilizing machine learning for proactive detection of cardiovascular risks: A data-driven

approach," *Int. J. Sci. Res. Arch.*, vol. 13, no. 1, pp. 1280–1290, 2024.
[5] E. Hikmawati and N. Alamsyah, "Supervised Learning for Emotional Prediction and Feature Importance Analysis Using SHAP on Social Media User Data," *Ingénierie Systèmes Inf.*, vol. 29, no. 6, 2024.
[6] V. Vimbi, N. Shaffi, and M. Mahmud, "Interpreting artificial intelligence models: a systematic review on the application of LIME and SHAP in Alzheimer's disease detection," *Brain Inform.*, vol. 11, no. 1, p. 10, 2024.
[7] U. Michelucci, "Feature Importance and Selection," in *Fundamental Mathematical Concepts for Machine Learning in Science*, Springer, 2024, pp. 229–242.
[8] N. Alamsyah, B. Budiman, T. P. Yoga, and R. Y. R. Alamsyah, "XGBOOST HYPERPARAMETER OPTIMIZATION USING RANDOMIZEDSEARCHCV FOR ACCURATE FOREST FIRE DROUGHT CONDITION PREDICTION," *J. Pilar Nusa Mandiri*, vol. 20, no. 2, pp. 103–110, 2024.
[9] J. Narkhede, "Comparative Evaluation of Post-Hoc Explainability Methods in AI: LIME, SHAP, and Grad-CAM," in *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, IEEE, 2024, pp. 826–830.
[10] T. Khater *et al.*, "Explainable Machine Learning Model for Alzheimer Detection Using Genetic Data: A Genome-Wide Association Study Approach," *IEEE Access*, 2024.
[11] W. Zhou, Z. Yan, and L. Zhang, "A comparative study of 11 non-linear regression models highlighting autoencoder, DBN, and SVR, enhanced by SHAP importance analysis in soybean branching prediction," *Sci. Rep..*, vol. 14, no. 1, p. 5905, 2024.
[12] I. M. Zubair, Y.-S. Lee, and B. Kim, "A New Permutation-Based Method for Ranking and Selecting Group Features in Multiclass Classification," *Appl. Sci.*, vol. 14, no. 8, p. 3156, 2024.
[13] E. Cantor, S. Guauque-Olarte, R. León, S. Chabert, and R. Salas, "Knowledge-slanted random forest method for high-dimensional data and small sample size with a feature selection application for gene expression data," *BioData Min.*, vol. 17, no. 1, p. 34, 2024.
[14] J. Labory, E. Njomgue-Fotso, and S. Bottini, "Benchmarking feature selection and feature extraction methods to improve the performances of machine-learning algorithms for patient classification using metabolomics biomedical data," *Comput. Struct. Biotechnol. J.*, vol. 23, pp. 1274–1287, 2024.
[15] A. Puiu, C. Gómez Tapia, M. E. Weiss, V. Singh, A. Kamen, and M. Siebert, "Prediction uncertainty estimates elucidate the limitation of current NSCLC subtype classification in representing mutational heterogeneity," *Sci. Rep..*, vol. 14, no. 1, p. 6779, 2024.
[16] H. Lee, U. Ozbulak, H. Park, S. Depuydt, W. De Neve, and J. Vankerschaver, "Assessing the reliability of point mutation as data augmentation for deep learning with genomic data," *BMC Bioinformatics*, vol. 25, no. 1, p. 170, 2024.
[17] N. Alamsyah, A. P. Kurniati, and others, "A Novel Airfare Dataset To Predict Travel Agent Profits Based On Dynamic Pricing," in *2023 11th International Conference on Information and Communication Technology (ICoICT)*, IEEE, 2023, pp. 575–581.
[18] N. Alamsyah, B. Budiman, T. P. Yoga, and R. Y. R. Alamsyah, "Comparison of LINEAR REGRESSION AND RANDOM FOREST MODELS FOR PREDICTION OF UNDERGROUND DROUGHT LEVELS IN FOREST FIRES," *J. Techno Nusa Mandiri*, vol. 21, no. 2, pp. 81–86, 2024.
[19] N. Alamsyah, A. Hendra, E. Setiana, T. P. Yoga, V. R. Danestiara, and others, "Improved Prediction Of Global Temperature Via LSTM Using ReLU Activation And Hyperparameter Optimization," in *2024 International Conference on Information Technology Research and Innovation (ICITRI)*, IEEE, 2024, pp. 41–46.

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

[20] A. G. Putrada, I. D. Oktaviani, M. N. Fauzan, and N. Alamsyah, "CNN Pruning for Edge Computing-Based Corn Disease Detection with a Novel NG-Mean Accuracy Loss Optimization," *Telematika*, vol. 17, no. 2, pp. 68–83, 2024.

DOI: https://doi.org/10.17509/coelite.v4i1.81356
p- ISSN 2829-4157 e- ISSN 2829-4149

Journal of Computer Engineering, Electronics and Information Technology (COELITE)

Journal homepage: https://ejournal.upi.edu/index.php/COELITE

A Metaheuristic-Based Approach to Inflation Prediction in Indonesia with Support Vector Regression (SVR)

Putu Manik Sesa Arianthi[1], Miftakhul Jannah[2], Noviyanti Santoso[3]*
[1,2,3]Department of Statistic Business, Institute Technology Sepuluh Nopember, Indonesia
Correspondence E-mail: sesaarianthi@gmail.com

ABSTRACT

Inflation prediction plays a vital role in individual, corporate, and government economic decision-making. This study evaluates the performance of the Support Vector Regression (SVR) algorithm in predicting monthly inflation using 251 data points from Bank Indonesia. Two parameter optimization methods, Grid Search Optimization and Particle Swarm Optimization (PSO), are applied to enhance prediction accuracy. Data pre-processing includes normalization using Min-Max Scaler and splitting into training and testing sets with ratios of 85:15 and 90:10. The optimization results indicate that PSO with a 90:10 ratio outperforms other approaches, achieving a Mean Absolute Percentage Error (MAPE) of 20.27% and an R-squared value of 89.76%. These findings highlight the significance of effective parameter optimization methods in improving prediction model performance, especially for non-linear data such as inflation. The results also demonstrate machine learning techniques' potential in analyzing time series data for economic and financial applications. This study provides insights into developing more accurate prediction systems, which can contribute to better economic planning and policy making.

ARTICLE INFO

Article History:
Submitted/Received 20 Dec 2024
First Revised 27 Feb 2025
Accepted 27 Mar 2025
First Available online 1 Apr 2025
Publication Date 1 Apr 2025

Keyword:
Support Vector Regression,
Particle Swarm Optimization,
Parameter Optimization,
Inflation Prediction,
Grid Search Optimization.

© 2025 Universitas Pendidikan Indonesia

1. INTRODUCTION

Inflation is a critical indicator of economic performance that directly impacts a country's stability. It reflects the general increase in prices over time and the corresponding decline in purchasing power. Inflation influences individual purchasing power, corporate investment decisions, currency value, and overall economic growth. High and unpredictable inflation often disrupts economic stability, leading to decreased consumer confidence, reduced foreign investment, and impaired economic planning. In developing countries like Indonesia, where economic volatility is more pronounced, the need for accurate inflation prediction is particularly crucial to support effective policy-making and safeguard macroeconomic stability. Inflation instability can harm people's purchasing power, investment, and exchange rate stability, thereby affecting overall economic performance [1]. Understanding inflation trends and making accurate forecasts is essential for enabling governments and financial institutions to formulate appropriate strategies. These forecasts provide critical insights that help mitigate inflation's adverse effects, ensuring sustainable economic growth while maintaining financial stability.

Inflation prediction is inherently complex due to the non-linear relationships between its influencing factors. Traditional statistical models often fall short in capturing these complexities, necessitating the use of advanced computational approaches. Machine learning, particularly Support Vector Regression (SVR), has emerged as a promising tool for time series forecasting. SVR is a regression method from SVM (Support Vector Machine), which is commonly used to overcome overfitting and has good performance for regression cases [2]. It works by finding a hyperplane that best fits the data while minimizing the prediction error. However, the accuracy of SVR models depends heavily on hyperparameter tuning, specifically the penalty parameter (C) and kernel function parameters. Selecting appropriate values for these parameters is vital for achieving high prediction accuracy.

To enhance prediction quality, leveraging advanced optimization techniques is necessary to refine machine learning models. Hyperparameter optimization techniques such as Grid Search Optimization and Particle Swarm Optimization (PSO) provide robust solutions for improving model performance. Grid Search is a systematic hyperparameter search technique where parameters are explored through a predefined grid of values, determined by the lower bound, upper bound, and step size [3]. Grid Search systematically evaluates all possible parameter combinations within a defined range, ensuring a thorough exploration of the parameter space. Meanwhile, Particle Swarm Optimization (PSO), inspired by the social behaviour of bird flocks and fish schools, efficiently searches for optimal solutions by leveraging collective intelligence. PSO has demonstrated strong performance in enhancing machine learning models, including SVR, for inflation prediction in Indonesia [4].

This research aims to integrate SVR with Grid Search and PSO to predict monthly inflation rates in Indonesia using historical data from Bank Indonesia. The study seeks to evaluate and compare the performance of these optimization techniques in improving the accuracy of SVR models. The dataset comprises 251 monthly inflation records, which undergo pre-processing steps including normalization and division into training and testing sets. The models are then optimized using Grid Search and PSO to identify the best-performing configuration.

By leveraging advanced optimization techniques, this study intends to address the challenges associated with inflation prediction and provide actionable insights for policymakers. Accurate forecasts can enable more effective economic planning, better resource allocation, and improved financial stability. The findings are also expected to contribute to the broader application of machine learning in economic modelling and highlight the potential of metaheuristic algorithms in enhancing predictive analytics.

DOI: https://doi.org/10.17509/coelite.v4i1.78101
p- ISSN 2829-4157 e- ISSN 2829-4149

2. METHODS

The analytical steps used in this study, starting from preparing the dataset to optimizing the model, are illustrated through the flowchart in **Figure 1.**

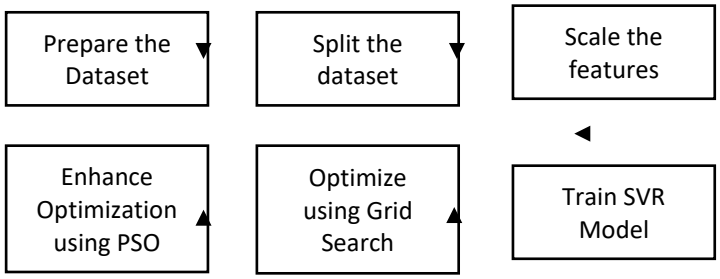


Figure 1. Flowchart the Method

Based on the flowchart, the first step is to prepare the dataset by adjusting the date-time format and data types, as well as defining the input and output features. Next, the data is split with a ratio of 0.85, followed by training the SVR model using default parameters. After obtaining the model evaluation results, predictions are made on the training data to assess whether the SVR with default parameters has a low error rate. The subsequent step involves optimization using grid search by testing splitting ratios of 0.85 and 0.90. The final step is to further enhance the SVR model using a metaheuristic algorithm, specifically Particle Swarm Optimization (PSO).

2.1. Support Vector Regression

SVM is a kernel-based method designed to determine the best hyperplane for separating classes. In contrast, SVR, which is an extension of SVM for regression problems, focuses on identifying a hyperplane that approximates the data while allowing for a permissible margin of error, rather than segmenting it into distinct categories as SVM does [5]. SVR is a method that can overcome overfitting, so it will produce a good performance [6]. The objective of SVR is to determine a regression function *f(x)* that effectively captures the relationship between the input dataset $x = \{x_1, x_2, x_3, ..., x_n\}$ and the target values $y = \{y_1, y_2, y_3, ..., y_n\}$ [7] as follows:

$$f(x) = w\phi(x) + b \tag{1}$$

W is a representation of the function's weight vector, b is the offset factor, and ϕ denotes the nonlinear mapping. To prevent overfitting the calibration data samples, SVR utilizes an objective function (Equation 2) and a loss function (Equation 3) to derive the regression parameters in (Equation 1):

$$\min \frac{1}{2} \| w \|^2 + C \sum_{1}^{n} (\xi_i + \xi_i^*) \tag{2}$$

$$\text{subject to: } \xi_i^* \begin{cases} y_i - (w\phi(x) + b) \le \varepsilon + \xi_i \\ (w\phi(x) + b) - y_i \le \varepsilon + \xi_i^* \\ \xi_i \ge 0, \xi_i^* \ge 0, i = 1, ..., n \end{cases} \tag{3}$$

The goal of SVM is to divide the dataset (classification) into 2 zones, while the goal of SVR is the opposite, namely, how to get all the dataset into one zone, while still minimizing the epsilon value (ε)[2]. The steps for using SVR are as follows [8].

a. Preparing training data
b. Selecting the kernel and parameters and their regulation

DOI: https://doi.org/10.17509/coelite.v4i1.78101
p- ISSN 2829-4157 e- ISSN 2829-4149

c. Creating a model to obtain coefficients
d. Using coefficients, then making estimates

The model's prediction accuracy is evaluated using the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE), calculated using the following formulas in Equations (4) and (5) [9].

$$MAPE = \frac{1}{T}\sum_{i=1}^{T}\left|\frac{d_i - \hat{d}_i}{d_i}\right| \tag{4}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(d_i - \hat{d}_i)^2} \tag{5}$$

The description of each index in the formulas is as follows: T represents the total number of test samples, d denotes the actual sample value, and $\hat{d}$ refers to the predicted value.

2.2. Grid Search Optimization

Optimize Parameter Grid is an effective method for determining optimal parameter values in an operator subprocess. This method involves evaluating the performance for each combination of selected parameter values. By communicating the optimal parameter values through the parameter set port, this method can be applied to various types of operators such as Support Vector Classifier, Lasso, Random Forest, and others [10]. The Optimize Parameter Grid has two modes: synchronous and asynchronous. Synchronous mode uses a list of predefined parameter value combinations, while asynchronous mode generates all possible parameter value combinations. By using Optimize Parameter Grid, we can systematically and efficiently find the best parameter values [10].

Grid search has been successfully applied in many inflation forecasting studies. For example, Alfiyatin et al. [11] combined Extreme Learning Machine (ELM) with improved Particle Swarm Optimization to forecast inflation accurately. Isnaeni et al. [12] applied SVR with RBF kernels and optimized parameters using grid search to predict inflation in Indonesia. In addition, non-parametric approaches such as Nadaraya-Watson kernel regression and B-spline have also been compared with parametric models like ARIMA in regional inflation forecasting, as discussed by Wulandari et al. [13].

2.3. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is based on the behaviour of flocks of birds or fish. The PSO algorithm mimics the social behaviour of these organisms. Social behaviour consists of individual actions and the influence of other individuals in a group. The word particle refers, for example, to a bird in a flock of birds. Each individual or particle behaves by using its intelligence and is also influenced by the collective behaviour of the group. Thus, if one particle or a bird finds the right or shortest path to a food source, the rest of the group will also be able to follow the path immediately, even though they are far away in the group. In Particle Swarm Optimization (PSO), the swarm is assumed to have a certain size, with each particle initially located at a random location in a multidimensional space. Each particle is assumed to have two characteristics: position and velocity. Each particle moves in a certain space and remembers the best position it has ever travelled to, or found the food source, or the value of the objective function. Each particle communicates its best position or information to the other particles and adjusts its position and velocity based on the information it receives about the position [4].

DOI: https://doi.org/10.17509/coelite.v4i1.78101
p- ISSN 2829-4157 e- ISSN 2829-4149

The application of PSO has been widely explored in various forecasting models. Fauziah et al. [14] applied ELM for inflation prediction in Indonesia, showing that optimization methods improve model performance. In Egypt, El-Daly [15] employed neural networks for inflation forecasting, highlighting the effectiveness of machine learning. Ponziani [16] used ARDL models, showing comparative accuracy with newer models. Bandara and De Mel [17] evaluated multiple supervised learning models in Sri Lanka, demonstrating PSO's viability. In Indonesia, Raharja et al. [18] implemented ANFIS for inflation prediction, while Aysun and Ekinci [19] explored various machine learning techniques in Turkey. Notably, Nurjanah and Sumarna [20] applied PSO to train a feedforward neural network for Indonesian inflation prediction, validating the utility of metaheuristics in economic modelling.

3. RESULTS AND DISCUSSION

3.1. Descriptive Statistics

The analysis will be conducted using monthly inflation rate data obtained from Bank Indonesia (BI). The dataset consists of 251 data points, with the structure of input (inflation rate at n-1 and n-2) and output (inflation rate at n) features for the SVR model in **Table 1**.

Table 1. Inflation Rate

Month (n)	Inflation Rate		
	n	n-1	n-2
March 2004	5,11%	4,60%	4,82%
April 2004	5,92%	5,11%	4,60%
May 2004	6,47%	5,92%	5,11%
June 2004	6,83%	6,47%	5,92%
July 2004	7,20%	6,83%	6,47%
August 2004	6,67%	7,20%	6,83%
November 2024	1,55%	1,71%	1,84%

A descriptive analysis will then be conducted as an initial step to observe the inflation rate trends over time, with the analysis results presented in a time series plot in **Figure 2**.

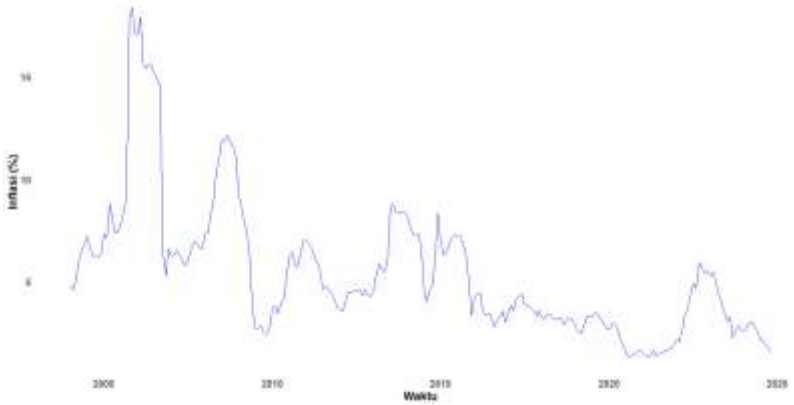


Figure 2. Inflation Rate Time Series Plot

Figure 2 shows a decreasing trend in the inflation rate from January 2004 to November 2024. Additionally, the inflation rate appears to fluctuate each year, with no clear linear pattern, indicating that the data is non-linear. The highest inflation rate occurred in November 2005 at 18.38%, while the lowest was in August 2020. The next step in the analysis is to prepare the data for processing using Support Vector Regression (SVR).

3.2. Support Vector Regression

Analysis using SVR is sensitive to data scale, so a pre-processing step is required. First, the data is split into training and testing sets with a ratio of 0.85, while maintaining the sequence of the data since it is a time series. Then, a min-max scaler is applied to standardize

DOI: https://doi.org/10.17509/coelite.v4i1.78101
p- ISSN 2829-4157 e- ISSN 2829-4149

the data, resulting in a value range from 0 to 1. The model was built using the ***caret*** package, focusing on two main parameters, c and sigma. By default, the *caret* package uses a value of C = 1. The results of the inflation rate predictions on the training data using C = 1 and Sigma = 0.1 are shown in **Figure 3**.

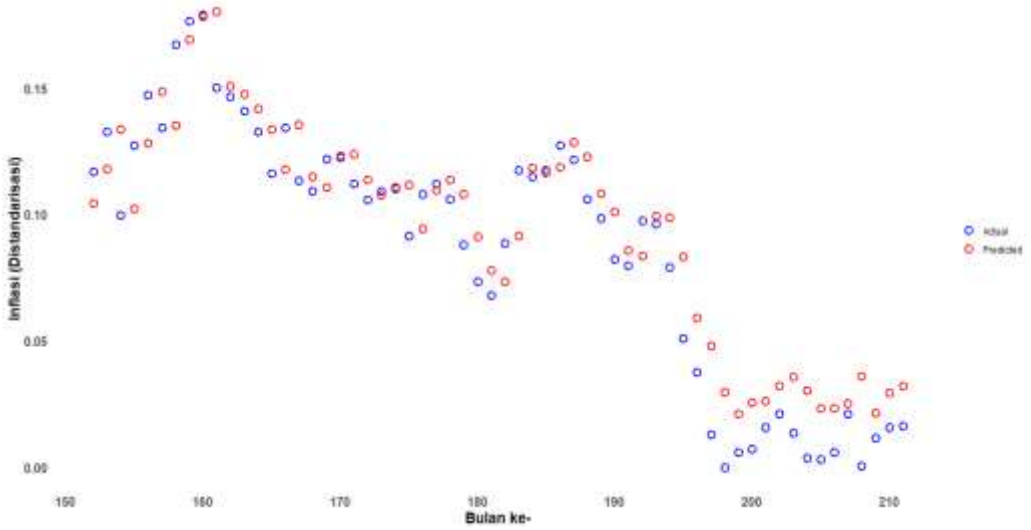


Figure 3. Prediction Results on the Training Data Using the RBF Kernel and Default Parameters

Figure 3 shows the prediction results on the training data, where the blue points represent the actual data and the red points represent the predicted values. The plot is used to assess how well the model captures the underlying pattern and to visually evaluate the magnitude of the error. Based on the figure, it can be observed that some patterns are not captured well. This is further supported by the model evaluation on the testing data using parameters C = 1 and Sigma = 0.1, which resulted in an MAPE of 29% and an accuracy of 89.38%.

3.3. Optimization

To obtain the best SVR model with the smallest error, parameter optimization will be conducted using two approaches: grid search and metaheuristic algorithms. The model was built using the *caret* package, focusing on two main parameters c and sigma. Optimization was performed through k-fold validation using the *SVM Radial* kernel, also known as the *Radial Basis Function* (RBF). The results of the grid search optimization for various combinations of C and sigma with a splitting ratio of 0.85 and 0.90 are presented in Table 2.

Table 2. Grid Search Optimization

Splitting Ratio	k	c	sigma	RMSE
0.85	2	13.5	0.051	0.059
	3	16.1	0.051	0.057
	4	3.9	0.051	0.056
	5	19.5	0.051	0.054
0.90	2	17.5	0.051	0.056
	3	18.7	0.051	0.057
	4	**9.1**	**0.051**	**0.050**
	5	24.5	0.201	0.052

Table 2 shows that the best model is achieved using a splitting ratio of 0.90, an SVR RBF kernel with 4-fold cross-validation, C = 9.1, and Sigma = 0.051. The best parameters obtained

DOI: https://doi.org/10.17509/coelite.v4i1.78101
p- ISSN 2829-4157 e- ISSN 2829-4149

through grid search optimization resulted in a MAPE value of 28.437%. A comparison will be made using a metaheuristic algorithm, specifically Particle Swarm Optimization (PSO), to determine if there is a difference in performance. The description of the parameters used in the PSO is shown in Table 3, and the results are presented in **Table 3**.

Table 3. PSO Parameters

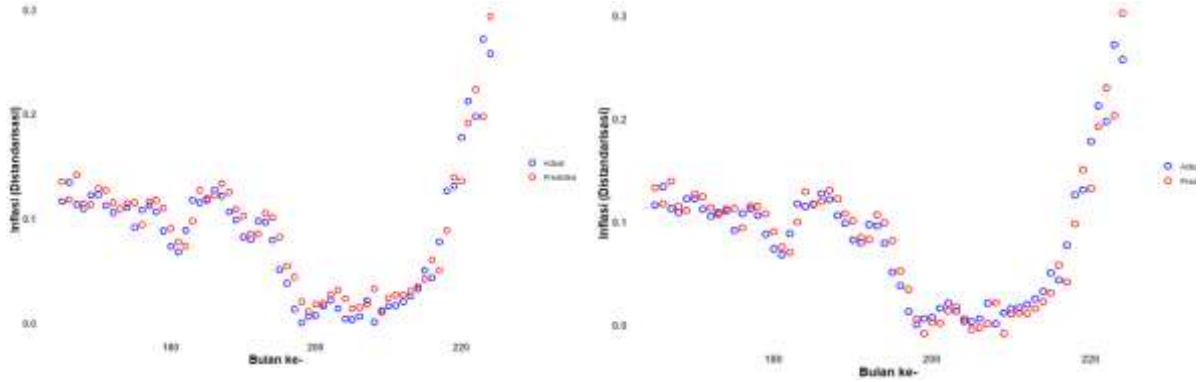
Parameters	Description
par	random from lower and upper
lower	C = 0.1, Sigma = 0.001
upper	C = 20, Sigma = 1
Control max iteration	50, 100

Table 3 presents the parameters used in PSO, while the optimization results with a maximum of 50 and 100 iterations, as well as splitting ratios of 0.85 and 0.90, are shown in **Table 4**.

Table 4. Result of PSO

Splitting Ratio	Max Iteration	Best C	Best Sigma	MAPE	RMSE	R-sq
0.85	50	2.959	0.592	21.71%	0.027	87.89%
	100	2.903	0.600	21.69%	0.027	87.90%
0.90	**50**	**18.274**	**1**	**20.27%**	**0.027**	**87.89%**
	100	18.717	0.999	20.28%	0.023	89.76%

Table 4 shows that the best SVR performance is achieved with a splitting ratio of 0.90, a maximum iteration of 50, C = 18.274, and Sigma = 1. A comparison of training data predictions using grid search optimization and metaheuristics is presented in **Figure 4**.



(a) Grid Search Optimization **(b)** Particle Swarm Optimization

Figure 4. Prediction Using Grid Search and PSO

Figure 4 shows that, visually, the error produced is not significantly large for both the grid search and PSO methods. However, based on the evaluation metric, namely the MAPE value, the model chosen to predict the inflation rate for the next 12 months is SVR-PSO with a splitting ratio of 0.90, maximum iteration = 50, C = 18.274, and Sigma = 1.

3.4. Prediction Result

The predicted inflation rate for the next 12 months using the best model from SVR-PSO is presented in **Table 5**.

Table 5. Prediction Result for 12 Months

Month	Inflation Rate
December 2024	3.74%
January 2025	3.98%
February 2025	4.31%

DOI: https://doi.org/10.17509/coelite.v4i1.78101
p- ISSN 2829-4157 e- ISSN 2829-4149

March 2025	4.67%
April 2025	5.00%
May 2025	5.24%
June 2025	5.36%
July 2025	5.41%
August 2025	5.39%
September 2025	5.35%
October 2025	5.31%
November 2025	5.28%

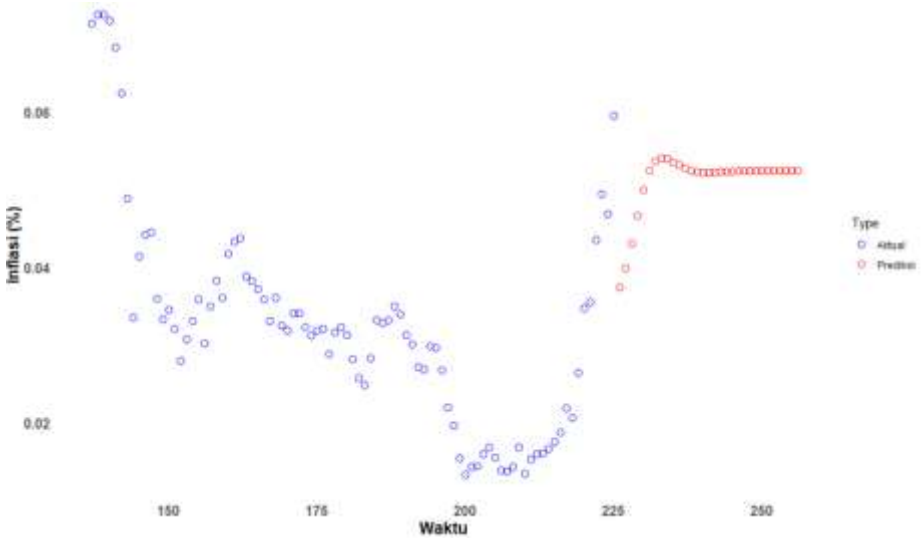


Figure 5. Inflation Rate Prediction

Figure 5 shows that the prediction results exhibit an upward trend at the beginning, which resembles the pattern observed in the previous months, characterized by significant fluctuations.

4. CONCLUSION

The results highlight that SVR performance is highly influenced by the choice of parameters, necessitating pre-processing and parameter optimization. Grid Search and Particle Swarm Optimization (PSO) were applied to determine the optimal parameters, with PSO yielding the best performance. The final model, SVR-PSO with a splitting ratio of 0.90, maximum iterations of 50, C = 18.274, and sigma = 1, achieved the lowest MAPE value and produced predictions closely aligned with the actual data. These findings suggest that the SVR-PSO approach can serve as a reliable method for inflation rate forecasting, supporting more accurate economic decision-making.

5. REFERENCES

[1] A. S. Fadila and A. Purnamasari, "Pengaruh Inflasi Terhadap Pertumbuhan Ekonomi Indonesia," *Ekonomica Sharia: Jurnal Pemikiran dan Pengembangan Ekonomi Syariah*, vol. 7, no. 1, pp. 17–27, 2021, [Online]. Available: www.bps.go.id,

[2] A. Aulia, B. Aprianti, Y. Supriyanto, and C. Rozikin, "Prediksi Harga Emas dengan Menggunakan Algoritma Support Vector Regression (SVR) dan Linear Regression (LR)," *Jurnal Ilmiah Wahana Pendidikan*, vol. 8, no. 5, pp. 84–88, 2022.

[3] R. Khalid and N. Javaid, "A Survey on Hyperparameters Optimization Algorithms of Forecasting Models in Smart Grid," *Sustain Cities Soc*, 2020, [Online]. Available: http://www.njavaid.com

DOI: https://doi.org/10.17509/coelite.v4i1.78101
p- ISSN 2829-4157 e- ISSN 2829-4149

[4] Priliani, L. A., Pramudito, E. N., & Hartati, S. (2022). Forecasting inflation rate using support vector regression (SVR) based weight attribute particle swarm optimization (WAPSO). Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 6(1), 149–154. https://doi.org/10.29207/resti.v6i1.3689

[5] M. Beniwal, A. Singh, and N. Kumar, "Forecasting Long-Term Stock Prices Of Global Indices: A Forward-Validating Genetic Algorithm Optimization Approach For Support Vector Regression," *Appl Soft Comput*, vol. 145, Sep. 2023, doi: 10.1016/j.asoc.2023.110566.

[6] Lubis, Z., Arisandi, D., Ginting, B. N., & Syahputra, S. (2021). Perbandingan Kinerja Algoritma Support Vector Regression (SVR) dan Long Short Term Memory (LSTM) dalam Memprediksi Inflasi. Jurnal Teknologi Informasi dan Ilmu Komputer, 8(5), 903-912.

[7] B. Bargam, A. Boudhar, C. Kinnard, H. Bouamri, K. Nifa, and A. Chehbouni, "Evaluation Of The Support Vector Regression (SVR) And The Random Forest (RF) Models Accuracy For Streamflow Prediction Under A Data-Scarce Basin In Morocco," *Discover Applied Sciences*, vol. 6, no. 6, June. 2024, doi: 10.1007/s42452-024-05994-z.

[8] A. Arfan and L. ETP, "Perbandingan Algoritma Long Short-Term Memory dengan SVR Pada Prediksi Harga Saham di Indonesia," *PETIR*, vol. 13, no. 1, pp. 33–43, Mar. 2020, doi: 10.33322/petir.v13i1.858.

[9] Fahrudin, R., & Sumitra, I. D. (2020). Peramalan Inflasi menggunakan Metode SARIMA dan Single Exponential Smoothing (Studi Kasus: Kota Bandung). Majalah Ilmiah UNIKOM, 17(2), 111-120.

[10] RapidMiner, "Optimize Parameter Grid," RapidMiner Documentation.

[11] Alfiyatin, A. N., Mahmudy, W. F., Ananda, C. F., & Widodo, A. W. (2021). Inflation Rate Forecasting Using Extreme Learning Machine and Improved Particle Swarm Optimization. International Journal of Intelligent Engineering & Systems, 14(6), 333-342.

[12] Isnaeni, R., Sudarmin, S., & Rais, Z. (2022). Analisis Support Vector Regression (Svr) Dengan Kernel Radial Basis Function (Rbf) Untuk Memprediksi Laju Inflasi Di Indonesia. Variansi: Journal Of Statistics And Its Application On Teaching And Research, 4(1), 30-38. DOI: 10.35580/variansiunm13

[13] Wulandari, N., Putri, R. A., Nazhirah, A., & Novia, S. A. (2025). Inflation Forecasting for Riau Province: A Comparison of Parametric ARIMA with Nonparametric Nadaraya-Watson Kernel Regression and B-Spline Methods. Jurnal Matematika, Statistika dan Komputasi, 21(3), 763-774.

[14] Fauziah, R., Handayani, R., & Santoso, A. (2022). Penerapan Extreme Learning Machine (ELM) untuk Prediksi Tingkat Inflasi di Indonesia. Jurnal Rekayasa Informasi, 11(1), 22-29.

[15] El-Daly, S. H. (2021). Forecasting Inflation Rates Using Artificial Neural Networks: Evidence from Egypt. The Journal of Economic and Administrative Sciences, 37(1), 164-184.

[16] Ponziani, R. M. (2023). Inflation Forecasting Using Autoregressive Distributed Lag (ARDL) Models. Jurnal Ekonomi & Studi Pembangunan, 24(2), 316–330.

[17] Bandara, W. M. S., & De Mel, W. A. R. (2024). Evaluating the Efficacy of Supervised Machine Learning Models in Inflation Forecasting in Sri Lanka. Am. J. Appl. Stat. Econ, 3, 51-60.

[18] Raharja, M. A., Darmawan, I., Nilakusumawati, D. P. E., & Supriana, I. W. (2021). Analysis of membership function in implementation of adaptive neuro fuzzy inference

system (ANFIS) method for inflation prediction. Journal of Physics: Conference Series, 1722(1), 012005.

[19] Aysun, N., & Ekinci, E. (2022). Inflation Forecasting Using Machine Learning Algorithms in Turkey. Journal of Business, Economics and Finance (JBEF), 11(2), 173-183.

[20] Nurjanah, A., & Sumarna, H. (2022). Prediksi Inflasi di Indonesia Menggunakan Jaringan Saraf Tiruan Feedforward Backpropagation dengan Particle Swarm Optimization. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), 10(1), 10-18.

DOI: https://doi.org/10.17509/coelite.v4i1.78101
p- ISSN 2829-4157 e- ISSN 2829-4149

Journal of Computer Engineering, Electronics and Information Technology (COELITE)
Journal homepage: https://ejournal.upi.edu/index.php/COELITE

Optimizing Logistic Regression for Digital Marketing Campaigns: Insights from Hyperparameter Tuning with Optuna

Budiman[1], Nur Alamsyah[2], Elia Setiana[3], Fikri Muhamad Fahmi[4]*
[1,3]Informatika, Universitas Informatika dan Bisnis Indonesia, Indonesia
[2,4]Sistem Informasi, Universitas Informatika dan Bisnis Indonesia, Indonesia
Correspondence E-mail: budiman@unibi.ac.id

ABSTRACT

The shift from mass media advertising to digital marketing driven by the internet revolution highlights the need for data-driven strategies and emerging technologies like Artificial Intelligence (AI). This study aims to develop effective strategies for digital marketing campaigns that enhance customer engagement and increase conversion rates. Key factors such as CampaignChannel, CampaignType, AdSpend, and demographic characteristics (Age, Gender, Income) were analyzed about performance metrics like ConversionRate and WebsiteVisits. A dataset with 20 customer demographics and campaign details features was processed and evaluated using machine learning models, including Logistic Regression, Random Forest, and XGBoost. Pre-processing involved handling missing values, feature selection, and splitting data into training and testing sets. Hyperparameter tuning using Optuna optimized the Logistic Regression Model, achieving the best performance with 89% accuracy. The findings reveal significant relationships between campaign factors and customer behaviour, providing actionable insights to enhance ROI (ROI). This study contributes to a machine learning-based framework for effective segmentation, personalized interactions, and efficient marketing budget management. The study advances AI applications in digital marketing by addressing challenges like data dynamics and shifting business conditions, paving the way for adaptive and data-driven strategies.

© 2025 Universitas Pendidikan Indonesia

ARTICLE INFO

Article History:
Submitted/Received 01 Mar 2025
First Revised 15 Mar 2025
Accepted 25 Mar 2025
First Available online 01 Apr 2025
Publication Date 01 Apr 2025

Keyword:
Digital Marketing,
Customer Engagement,
Machine Learning,
Logistic Regression,
Campaign Optimization,
Hyperparameter,
Optuna

1. INTRODUCTION

The Internet revolution and digital marketing have shifted the dominance of mass media advertising towards more modern promotions through the internet and electronic devices in the digital environment [1][2]. In the ever-evolving social network advertising landscape, accurate and quality data play a critical role in the success of predictive models [3]. Businesses must understand customer behaviour and preferences to increase revenue and create better customer engagement [4]. In addition, artificial intelligence (AI) as an emerging technology has become a significant force in various fields, such as healthcare, security, industry, big data, the modern economy, and digital marketing [5]. Studies show that AI will increasingly influence marketing operations by helping companies increase customer loyalty through more personalized interactions [6][7].

AI-powered digital marketing is revolutionizing how organizations create campaign content, generate leads, reduce customer acquisition costs, and manage customer experience efficiently [8]. Algorithms used by online promotion platforms can identify the best strategies to improve campaign performance, although some companies prefer to build customized systems internally [9]. Challenges in digital marketing include dynamic user data and changing business conditions that can affect campaign success [10][11]. Therefore, companies must be able to conduct in-depth customer segmentation by discovering their behavioural patterns to market products more effectively [12].

The integration of artificial intelligence (AI) and machine learning (ML) in digital marketing brings significant changes in traditional marketing strategies [13]. AI helps companies predict consumer demand, optimize advertising campaigns, and find the optimal target audience to increase marketing effectiveness [14][15]. Deep learning through AI allows marketers to tap into user behaviour patterns on websites or mobile apps to respond quickly and refine advertising offers [16][17]. By utilizing these technologies, digital marketing becomes necessary for companies to keep up with evolving consumer needs and opens up new domains for the future of marketing strategies [18].

This study aims to develop effective strategies for digital marketing campaigns that enhance customer engagement and increase conversion rates. This study analyses the influence of factors such as campaign channel, campaign type, ad spend, and customer demographic characteristics (age, gender, and income) on key performance metrics such as conversion rate and website visits. In addition, this study aims to implement and evaluate machine learning models to predict customer behaviour patterns and provide strategic insights that can be applied to improve campaign efficiency and return on investment (ROI).

As a contribution, this study is expected to provide a machine learning-based framework that improves the accuracy of customer behaviour prediction and enables companies to optimize campaign strategies through more effective segmentation, personalization of interactions, and more efficient management of marketing budgets. As such, this study contributes to the scientific literature on digital marketing and the application of artificial intelligence technology in improving the effectiveness of marketing campaigns.

2. METHODS

The methodology depicted in **Figure 1** outlines the sequential steps for optimizing digital marketing campaigns. It starts with a comprehensive review of relevant literature to explore current strategies and identify areas for improvement.

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

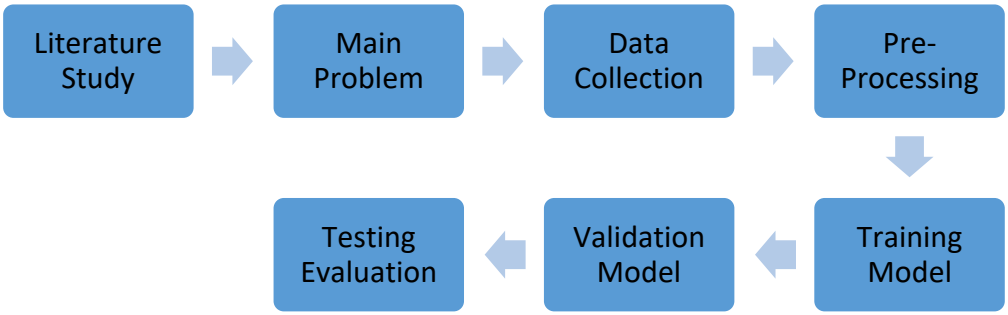


Figure 1. Proposed Method

2.1. Literature Study

This study draws on relevant literature on digital marketing campaign optimization. Previous studies show that metrics such as ClickThroughRate (CTR), ConversionRate, WebsiteVisits, and AdSpend are crucial in understanding customer behaviour and campaign effectiveness. Studies have also highlighted the importance of segmenting campaigns based on demographic features such as Age, Gender, and Income. Furthermore, tools like predictive modelling and machine learning have been widely used to identify patterns in customer interactions and optimize campaign performance [19][20].

2.2. Main Problem

The main problem to be addressed is developing a strategy that maximizes customer engagement and Conversion in digital marketing campaigns. The objective is to analyze the impact of factors such as CampaignChannel, CampaignType, AdSpend, and demographic characteristics (Age, Gender, Income) on key performance metrics like ConversionRate and WebsiteVisits. This study aims to provide actionable insights to enhance campaign efficiency and ROI (Return on Investment).

2.3. Data Collection

We utilized a digital marketing campaign dataset from Kaggle, consisting of 20 features capturing customer demographics, campaign details, and performance metrics. The dataset contains 8,000 records, providing a comprehensive view of customer interactions and campaign performance. The dataset includes the following key features in **Table 1**:

Table 1. Features of a Digital Marketing Campaign

Feature	Description
CustomerID	Unique identifier for each customer.
Age	Age of the customer.
Gender	Gender of the customer (Male/Female).
Income	Income level of the customer.
CampaignChannel	The channel used for the campaign (Social Media, Email, PPC).
CampaignType	Type of campaign (Awareness, Retention, Conversion).
AdSpend	Advertising expenditure in the campaign.
ClickThroughRate	The proportion of ad clicks relative to total impressions.
ConversionRate	The ratio of successful conversions to total interactions.

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

WebsiteVisits	Number of visits to the website.
PagesPerVisit	Average number of pages viewed per visit.
TimeOnSite	Time spent on the website during visits.
SocialShares	Number of shares on social media.
EmailOpens	Count of emails opened by customers.
EmailClicks	Count of clicks on links within emails.
PreviousPurchases	Total purchases made by the customer before the campaign.
LoyaltyPoints	Customer loyalty points accumulated.
AdvertisingPlatform	Platform used for advertisements.
AdvertisingTool	Specific tools or technologies used in the campaign.
Conversion	Binary indicator for conversion success (1 for success, 0 for failure).

2.4. Pre-Processing

Pre-processing of the dataset involves several steps to ensure data quality and prepare it for analysis:

a. Handling missing values to ensure completeness.
b. Normalizing numerical features like AdSpend, Income, and TimeOnSite to ensure uniformity.
c. Encoding categorical variables such as CampaignChannel and Gender for machine learning compatibility.
d. Feature selection is used to identify the most relevant predictors for Conversion and other key metrics.
e. Splitting the dataset into training (80%) and testing (20%) sets to allow independent Model evaluation.

2.5. Training Model

We trained machine learning models using the pre-processed dataset to predict customer conversions and optimize campaign strategies. Models such as Logistic Regression, Random Forest, XGBoost, and Support Vector Machine were employed to analyze the relationship between demographic and campaign features and key performance indicators. Hyperparameter tuning and cross-validation were applied to enhance model accuracy and generalizability.

2.6. Validation Model

Cross-validation techniques such as k-fold cross-validation were employed to evaluate the robustness of the predictive models. This method helps prevent overfitting by ensuring that each data subset is used for training and testing. The averaged results across folds provide a reliable estimate of model performance on unseen data.

2.7. Testing Evaluation

The final step involved evaluating model performance using accuracy, precision, recall, and F1-score metrics. These metrics provided a comprehensive view of the model's ability to predict conversions and optimize campaign outcomes. Specific focus was given to precision (to minimize false positives in conversion prediction) and recall (to ensure all potential

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

conversions are identified). The evaluation results inform recommendations for campaign improvement and customer engagement strategies.

3. RESULTS AND DISCUSSION

3.1. Pre-Processing

Table 2 provides a comprehensive overview of the Digital Marketing Campaign Dataset, capturing key customer attributes and performance metrics. The dataset includes details such as Age, which ranges from 32 to 69 years, and Gender, where Male and Female customers are represented, reflecting a diverse target audience. CampaignChannel shows the distribution of marketing strategies across Social Media, Email, and PPC, highlighting the varied platforms utilized for customer engagement.

AdSpend demonstrates significant variation, ranging from $539.53 to $6497.87, indicating differences in budget allocation across campaigns. Metrics like ClickThroughRate and ConversionRate further illustrate campaign effectiveness, with ClickThroughRate varying from 0.0439 to 0.2774 and ConversionRate ranging from 0.0764 to 0.1827. These values suggest diverse levels of customer interaction and conversion success. Additionally, customers with higher AdSpend tend to exhibit higher engagement rates, as observed in Social Media and PPC campaigns.

Engagement indicators such as WebsiteVisits, PagesPerVisit, and TimeOnSite provide insights into customer behaviour on digital platforms. For example, customers with higher TimeOnSite often view more PagesPerVisit, indicating a deeper exploration of campaign content. While some customers exhibit high levels of interaction, others display minimal engagement. SocialShares highlights the extent of content dissemination on social media, varying widely among campaigns, with specific customers sharing content up to 89 times.

Email performance is measured through EmailOpens and EmailClicks, revealing varying levels of customer responsiveness to email marketing. Notably, campaigns with higher EmailClicks generally correlate with better ConversionRate, emphasizing the importance of effective email Communication strategies. LoyaltyPoints, ranging from 688 to 4345, underscore differences in customer loyalty and past interactions, suggesting that loyal customers are more likely to engage with campaigns and convert successfully.

Lastly, the Conversion column identifies whether a customer was successfully converted (1) or not (0), with the majority achieving conversions, reflecting the overall effectiveness of these campaigns. This dataset forms the basis for analyzing and optimizing digital marketing strategies to enhance customer engagement and campaign outcomes. By identifying patterns within this data in **Table 2**, businesses can tailor campaigns to meet customer preferences better and maximize return on investment (ROI).

Table 2. Digital Marketing Campaign Dataset

index	Age	Gender	CampaignChannel	AdSpend	ClickThroughRate	ConversionRate
0	56	Female	Social Media	6497.87	0.0439	0.0880
1	69	Male	Email	3898.67	0.1557	0.1827
2	46	Female	PPC	1546.43	0.2774	0.0764
3	32	Female	PPC	539.53	0.1376	0.0880
4	60	Female	PPC	1678.04	0.2529	0.1099
298	45	Male	Social Media	1234.67	0.1673	0.0642
299	38	Male	Email	2345.78	0.1267	0.0493

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

index	Age	Gender	CampaignChannel	AdSpend	ClickThroughRate	ConversionRate
300	32	Female	PPC	3467.98	0.2100	0.0810
301	58	Female	Social Media	5467.43	0.1900	0.1023
302	61	Male	Email	2845.65	0.1450	0.0710

index	WebsiteVisits	PagesPerVisit	TimeOnSite	SocialShares	EmailOpens
0	0	2.40	7.40	19	6
1	42	2.92	5.35	5	2
2	2	8.22	13.79	0	11
3	47	4.54	14.69	89	2
4	0	2.05	13.99	6	6
298	15	1.78	6.23	4	4
299	20	2.89	5.15	8	5
300	10	3.12	7.85	15	6
301	30	4.30	10.45	20	10
302	25	3.80	8.40	12	4

index	EmailClicks	LoyaltyPoints	Conversion
0	9	688	1
1	7	3459	1
2	2	2337	1
3	2	2463	1
4	6	4345	1
298	2	1234	0
299	3	2100	0
300	4	2340	0
301	7	3000	0
302	5	2500	0

Table 3 shows the descriptive statistics of the dataset that provide deep insights into customer attributes and digital marketing campaign performance. Regarding demographics, the average age of customers is 43.6 years old, with an age range from 18 to 69 years old, reflecting a wide range of targets. Customer income varied significantly, with an average of $84,664.20, but spread from $20,014 to $149,986, suggesting segmentation by purchasing power. Average advertising expenditure was recorded at $5,000.94 with significant variations (standard deviation of $2,838.04), reflecting different budget allocations between campaigns. The average ClickThroughRate (CTR) was 0.1548, and the average ConversionRate was 0.1044, highlighting the moderate effectiveness of campaigns in attracting and converting customers. These metrics show that campaigns with a higher CTR tend to generate a higher ConversionRate, although there is some variation.

Customer engagement is evident from the average website visit (24.75 visits) and average time spent (7.73 minutes). Customers who spend more time on the site tend to visit more pages per visit (5.55 pages on average). Social media sharing activity was also significant, with an average of 49.8 shares per campaign, demonstrating the appeal of the content on social media platforms. Customer response to email campaigns was also good, with an average of

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

9.48 email opens and 4.47 clicks on email links. Customers with more email engagement tend to have a higher likelihood of Conversion, as seen from the positive correlation between EmailClicks and ConversionRate. Customer loyalty was also an essential factor, with an average of 2,490.27 loyalty points, indicating an ongoing relationship between the customer and the brand. The overall campaign conversion rate was very high, with an average of 87.65% of subscribers successfully converted. This reflects excellent success in the overall digital campaign strategy. This analysis shows that optimizing ad spend, email marketing, and website engagement can improve campaign performance to maximize ROI and strengthen customer relationships.

Table 3. Descriptive Statistics of the Digital Marketing Campaign Dataset

Feature	mean	std	min	25%	50%	75%	max
Age	43.63	14.90	18	31	43	56	69
Income	84664.20	37580.39	20014	51744.5	84926.5	116815.75	149986
AdSpend	5000.94	2838.04	100.05	2523.2	5013.4	7407.99	9997.91
ClickThroughRate	0.15	0.08	0.01	0.08	0.15	0.23	0.30
ConversionRate	0.10	0.05	0.01	0.06	0.10	0.15	0.20
WebsiteVisits	24.75	14.31	0	13	25	37	49
PagesPerVisit	5.55	2.61	1.00	3.30	5.53	7.84	10.00
TimeOnSite	7.73	4.23	0.50	4.07	7.68	11.48	15.00
SocialShares	49.80	28.90	0	25	50	75	99
EmailOpens	9.48	5.71	0	5	9	14	19
EmailClicks	4.47	2.86	0	2	4	7	9
PreviousPurchases	4.49	2.89	0	2	4	7	9
LoyaltyPoints	2490.27	1429.53	0	1254.75	2497	3702.25	4999
Conversion	0.88	0.33	0	1	1	1	1

Figure 2 is a heatmap image that visualizes the correlation relationship between the various variables in the digital marketing campaign dataset. The correlation values range from -1 to 1, with 1 indicating a perfect positive correlation, -1 indicating a perfect negative correlation, and 0 indicating no relationship. Some important patterns can be observed from this heatmap. The ConversionRate variable shows a moderate positive relationship with ClickThroughRate, which suggests that the more customers click on an advert, the more likely they are to convert. In addition, PagesPerVisit and TimeOnSite have a positive correlation, indicating that customers who spend more time on the website tend to browse more pages, which can increase the chances of engagement.

AdSpend has a low correlation with ConversionRate, which may indicate that ad spend alone is insufficient to ensure a high conversion rate. This underlines the importance of other factors, such as the quality of ad content and audience segmentation strategies. The LoyaltyPoints variable has a moderate positive correlation with Conversion, suggesting that more loyal customers tend to convert more quickly. Some variables, such as Income and SocialShares, show low or almost zero correlation with key performance variables such as

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

ConversionRate and ClickThroughRate, indicating that customer income and social media sharing activity may not directly affect campaign performance in this dataset. Overall, this heatmap shows that while some variables such as ClickThroughRate, PagesPerVisit, and LoyaltyPoints have a significant influence on campaign performance, many other variables show weaker relationships, highlighting the need for additional analysis to identify key factors that influence marketing campaign success.

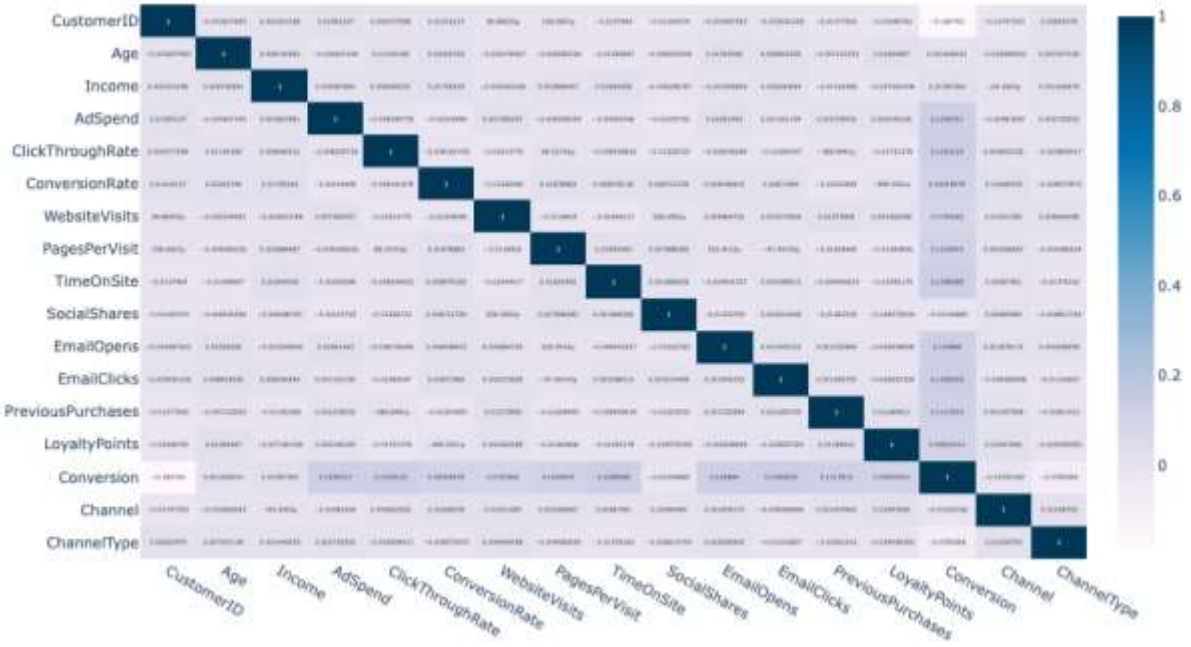


Figure 2. Heatmap

Figure 3 above shows the results of the Chi-Square analysis to measure the strength of the relationship between categorical variables in the dataset. Higher Chi-Square values indicate a stronger relationship between that variable and the target or dependent variable in the analysis. In this graph, Gender, CampaignChannel, and Age have the highest Chi-Square values, indicating that these variables significantly influence the target or campaign outcome. This suggests that demographic factors such as gender and age, as well as the campaign channel used, play an essential role in determining the success of a digital marketing campaign. In contrast, variables such as SocialShares, ChannelType, and PreviousPurchases have lower Chi-Square values, suggesting that the relationship between these variables and campaign outcomes is relatively weak or insignificant. Variables such as AdvertisingPlatform and AdvertisingTool showed almost no relationship, indicating that they may not be relevant or have little impact on campaign success. These results provide valuable insights for identifying essential variables that should be prioritized in marketing strategies, such as focusing on customer demographics and selecting appropriate campaign channels while reducing attention to variables with weak influence for better efficiency.

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

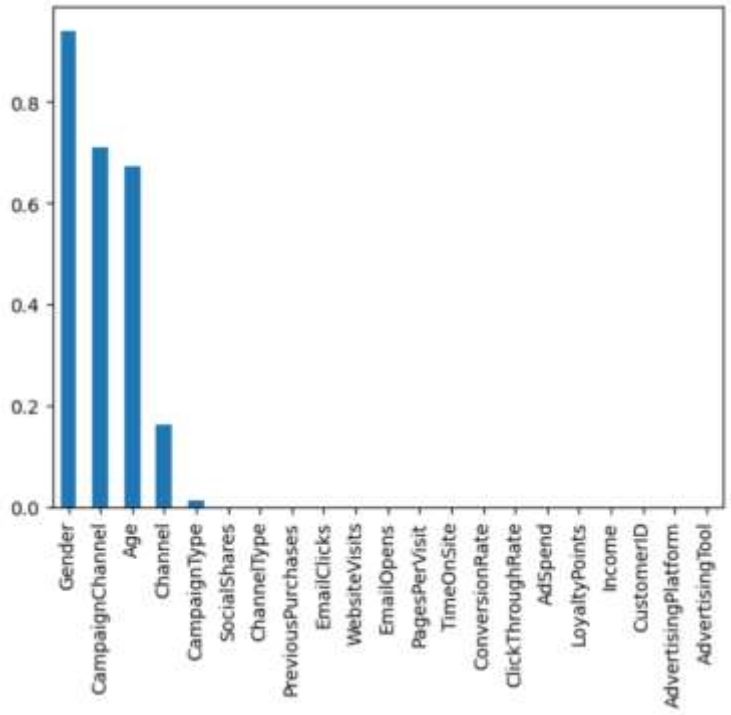


Figure 3. Chi-Square analysis

3.2. Training Model

Table 4 compares the classification reports; the performance of the tested models shows significant variation in their ability to handle predictions. Support Vector Classification (SVC) showed great weakness in predicting class 0, with precision, recall, and f1-score being 0.00 for this class. This indicates that the model could not detect or predict class 0 despite performing very well for class 1 (precision: 0.88, recall: 1.00, f1-score: 0.93). The model's overall accuracy is 0.88, but the imbalance in class prediction makes it less reliable in applications that require detection in both classes.

In contrast, Random Forest Classification showed better ability in detecting class 0 with a precision of 0.97, although its recall was low at 0.14, resulting in an f1-score of 0.25. For class 1, the model performed almost flawlessly with a recall of 1.00 and an f1-score of 0.94, resulting in an overall accuracy of 0.89. While the accuracy and performance of class 1 are pretty high, the challenge remains in improving the recall of class 0 to provide more balanced results. While both models show excellence in handling class 1 (majority), Random Forest Classification has the edge in accommodating class 0 predictions, although there is still room for improvement. Support Vector Classification, however, requires further tuning to handle significant class imbalance. In real applications, model selection depends on business priorities, such as whether it is more critical to maximize minority class predictions or maintain high accuracy on the majority class.

Table 4. Comparison of Classification Report

Metric	LR	SVM	RF	KNN	DT
Precision (Class 0)	0.77	0	0.97	0.52	0.54
Recall (Class 0)	0.17	0	0.14	0.17	0.56

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

Metric	LR	SVM	RF	KNN	DT
F1-Score (Class 0)	0.28	0	0.25	0.25	0.55
Precision (Class 1)	0.9	0.88	0.89	0.89	0.94
Recall (Class 1)	0.99	1	1	0.98	0.93
F1-Score (Class 1)	0.94	0.93	0.94	0.93	0.93
Accuracy	0.89	0.88	0.89	0.88	0.89
Macro Avg Precision	0.83	0.44	0.93	0.71	0.74
Macro Avg Recall	0.58	0.5	0.57	0.57	0.75
Macro Avg F1-Score	0.61	0.47	0.6	0.59	0.74
Weighted Avg F1-Score	0.86	0.82	0.86	0.85	0.89

The learning curve for Logistic Regression shows a steady increase in validation accuracy as the dataset size increases, reaching a maximum accuracy of around 89%. Training accuracy is slightly higher than validation accuracy, but the gap between the two is relatively small, indicating that the model has low bias and only slight overfitting. The model shows consistent performance, making it a good choice for datasets with fairly linear or straightforward patterns.

The Support Vector Classifier (SVC) has a more volatile learning curve. The training accuracy remains high, but the validation accuracy is relatively lower and does not show significant improvement as the data grows. The large gap between training and validation accuracy indicates serious overfitting. This suggests that the SVC is less than optimal in generalization on this dataset, mainly if no further parameter tuning is performed.

Random Forest shows almost perfect training accuracy across all dataset sizes, but validation accuracy tends to plateau at around 89%. The large gap between training and validation accuracy indicates that the model suffers from significant overfitting. Despite the high validation accuracy, the model may be less efficient because it learns too many details from the training data, which may affect the generalization of new data.

The K-Nearest Neighbor (KNN) learning curve shows a promising upward trend in validation accuracy as the data grows, approaching 91% at the maximum dataset size. Training accuracy is slightly higher than validation, but the gap between the two is relatively small, indicating that the model has a good balance between bias and variance. KNN appears to be one of the best models for generalization on this dataset.

Decision Tree has a similar learning curve to Random Forest, with training accuracy close to 100%, but validation accuracy tends to plateau at around 89%. The large gap between training and validation accuracy suggests that the Decision Tree also suffers from overfitting, even though it is a simpler model than Random Forest. This indicates that Decision Tree may be less effective for this dataset without additional regularization methods.

Based on the learning curve analysis in **Figure 4**, K-Nearest Neighbor (KNN) and Logistic Regression performed best in generalization, with a small gap between training and validation accuracy and high validation accuracy. Random Forest and Decision Tree have overfitting issues, although their validation accuracy is quite good. Meanwhile, SVC performs suboptimal due to low validation accuracy and significant fluctuations. Since Logistic Regression has better accuracy than SVC at 0.89, Logistic Regression is the best choice for this dataset, depending on the complexity of the data and the computational resources available.

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

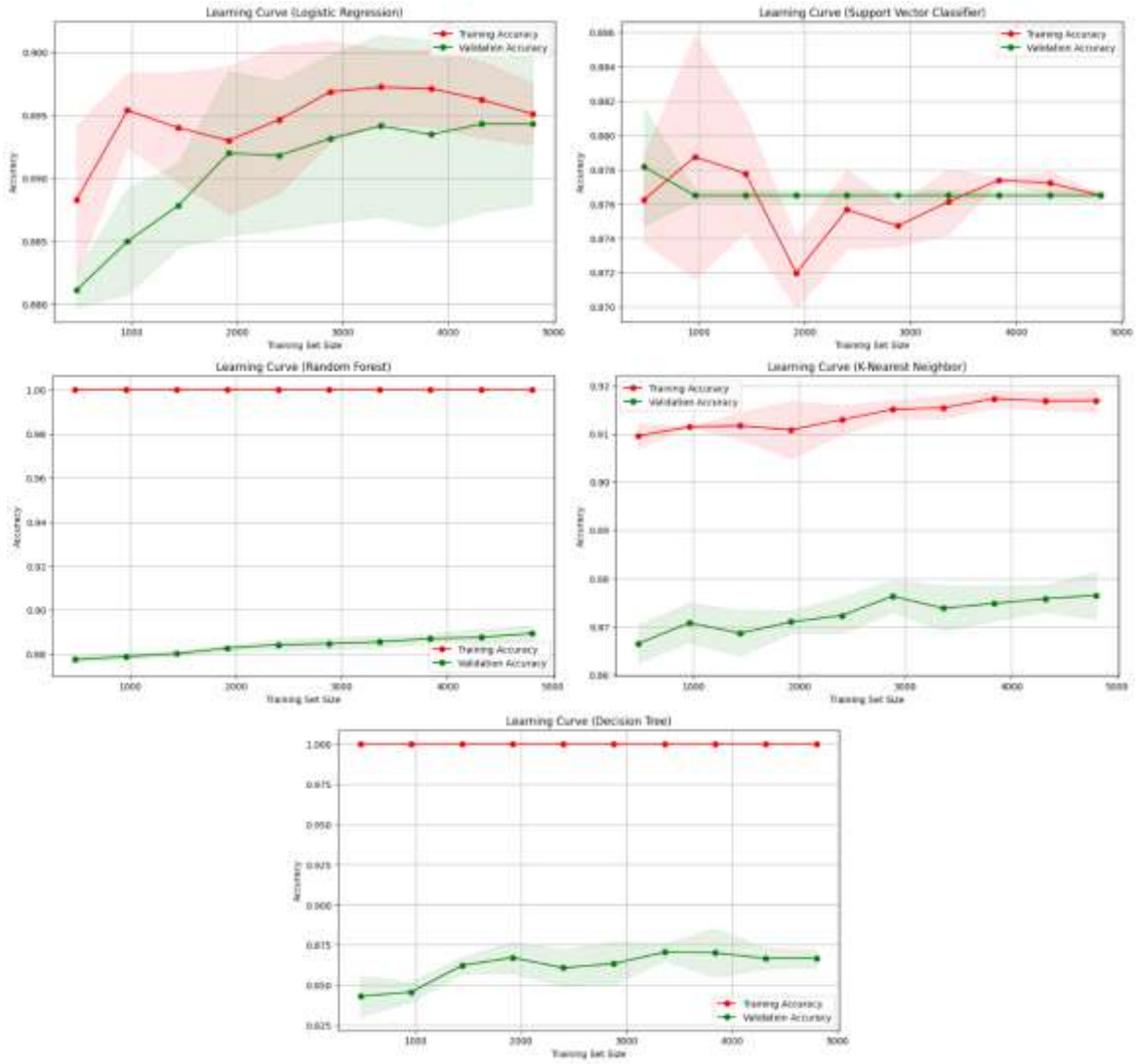


Figure 4. Learning Curve

After the analysis shows that Logistic Regression is the best model based on the learning curve, the next step is to improve the performance of this model through hyperparameter tuning. Hyperparameter tuning aims to find the optimal combination of parameters so that the model can maximize accuracy and generalization. **Figure 5** shows the learning curve analysis of the hyperparameter tuning process of the Logistic Regression model using Grid Search CV, Bayes Search CV, and Optuna, showing significant differences in model performance concerning training and validation accuracy. The first graph with Grid Search CV shows a stable trend with training and validation accuracy slowly approaching equilibrium as the dataset size increases. However, the gap between training and validation accuracy is still visible at small dataset sizes, indicating the model has a slight overfitting that decreases with larger datasets.

The learning curve of Bayes Search CV shows better performance than Grid Search, with more stable and higher validation accuracy. The training and validation accuracies on larger datasets tend to converge, indicating that Bayesian Optimisation successfully finds better parameter combinations to improve model generalization. In addition, the fluctuation in validation accuracy is smaller than that of Grid Search, reflecting more consistent results in tuning iterations.

Optuna, as a more adaptive hyperparameter tuning approach, showed very competitive results. The validation accuracy is higher than the other methods on almost all dataset sizes,

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

with a small gap between training and validation accuracy. This shows that the Logistic Regression model tuned using Optuna successfully minimizes overfitting while maintaining high validation accuracy. Optuna's adaptivity in efficiently exploring the parameter space provides a significant advantage over Grid Search and Bayes Search CV.

Optuna produced the most optimal model for Logistic Regression in generalisability and stability, followed by Bayes Search CV. Grid Search CV, although providing decent results, is less efficient and performs less than the other two methods. Thus, Optuna is the recommended method for hyperparameter tuning in Logistic Regression based on this dataset and learning curve analysis.

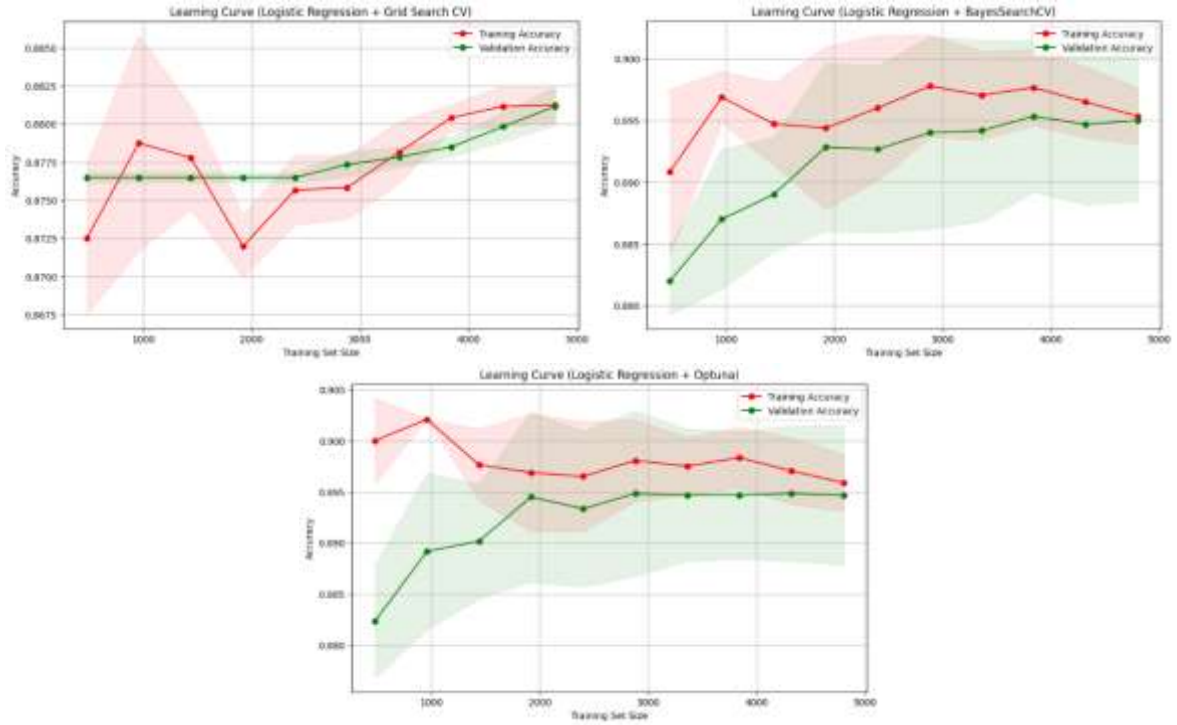


Figure 5. Learning Curve Analysis of the Hyperparameter Tuning Process of the Logistic Regression

After hyperparameter tuning using Optuna on Logistic Regression, model performance was slightly improved, especially for Class 0. Precision for Class 0 decreased from 0.77 to 0.75, but recall increased from 0.17 to 0.18, which resulted in an increase in F1-Score from 0.28 to 0.29. This shows that although the model slightly sacrificed precision in predicting Class 0, the model's ability to detect more Class 0 samples increased. For Class 1, the precision, recall, and F1-Score remained consistent at 0.90, 0.99, and 0.94, respectively, indicating that the model's performance for Class 1 was unaffected by tuning. The model accuracy remained at 89%, with no significant change. The best parameters found by Optuna are {'penalty': 'l1', 'C': 294.9937556473663, 'solver': 'saga'}, which provides the optimal configuration for the Logistic Regression model.

Regarding macro averages, recall increased from 0.58 to 0.59, indicating a slight improvement in the model's ability to detect samples on average in both classes. The F1-Score macro average also saw a slight increase from 0.61 to 0.62, reflecting a small improvement in the balance of precision and recall. However, the weighted average of precision, recall, and F1-Score remained stable at 0.88, 0.89, and 0.86, respectively, indicating that the contribution of Class 1 dominating the final model results was still very strong.

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

Compared to Logistic Regression before tuning, hyperparameter tuning with Optuna slightly improved, especially in detecting samples in Class 0, which was previously a significant weakness of the model. However, this improvement was insignificant and did not affect the overall accuracy. In conclusion, although hyperparameter tuning provides a slight improvement, additional strategies, such as handling data imbalance by oversampling or undersampling, are needed to improve performance on Class 0 more substantially.

3.3. Validation Model

Table 5 above is cross-validation results show fairly consistent performance across folds, with metrics such as precision, recall, and f1-score showing good stability. The model performed exceptionally well in the majority class (Class 1), with an average accuracy of 0.89 and an average recall of 0.99. This shows the model can minimize false positives while detecting almost all Class 1 samples well. The average F1-score for Class 1, which reached 0.94, reflects the optimal balance between precision and recall. In contrast, the performance of the minority class (Class 0) was much lower, with an average recall of only 0.20, indicating that the model could only detect about 20% of the total Class 0 samples. Precision for Class 0 also averaged 0.77, suggesting that predictions for Class 0 still generated many false positives.

The model's overall accuracy averages 89.3%, indicating that most samples can be classified correctly. However, this accuracy is heavily influenced by the dominant performance of Class 1, which has a much larger data distribution than Class 0. This imbalance means that high accuracy does not necessarily reflect good performance for both classes equally. Analysis of the mean and standard deviation shows that the performance for Class 1 is very stable, with almost insignificant variation between folds. In contrast, the recall for Class 0 has a higher standard deviation (0.0126), indicating that the model performance for this class is more sensitive to data variations.

Table 5. Classification Metrics by Fold

Metric	Fold 0	Fold 1	Fold 2	Fold 3	Mean	Std. Dev
tn	45	52	51	53		
fp	202	195	196	194		
fn	6	18	15	26		
tp	1747	1735	1738	1727		
prec_1	0.90	0.90	0.90	0.90	0.90	0.0011
prec_0	0.88	0.74	0.77	0.67	0.78	0.076092
recall_1	0.99	0.99	0.99	0.99	0.99	0.004081
recall_0	0.18	0.21	0.21	0.21	0.20	0.012601
accuracy	0.90	0.89	0.89	0.89	0.89	0.002208
f1_score	0.94	0.94	0.94	0.94	0.94	0.001347

In conclusion, while the model performs very well in Class 1, its performance in Class 0 requires further attention. To improve the performance of this minority class, measures such as handling data imbalance using oversampling (such as SMOTE) or undersampling, as well as adjusting the class weights in the loss function, can be applied. In addition, hyperparameter tuning that focuses more on improving recall for Class 0 can help to correct this performance imbalance. Using these strategies, the model is expected to achieve a better balance between the two classes, resulting in a more optimized classification.

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

3.4. Testing Evaluation

The final step involved evaluating model performance using accuracy, precision, recall, and F1-score metrics. These metrics provide a comprehensive view of the model's ability to predict conversions and optimize campaign outcomes. Specific focus was given to precision, which minimizes false positives in conversion prediction, and recall, which ensures that all potential conversions are identified. The model exhibited slight improvements after hyperparameter tuning with Optuna, particularly for Class 0. Precision for Class 0 decreased slightly from 0.77 to 0.75, but recall improved from 0.17 to 0.18, leading to an F1-score increase from 0.28 to 0.29. For Class 1, the precision, recall, and F1-score remained stable at 0.90, 0.99, and 0.94, respectively. This stability indicates that the tuning did not negatively affect the performance of the majority class while enhancing the detection of the minority class.

The learning curve analysis from the hyperparameter tuning process showed that Optuna consistently outperformed other methods, such as Grid Search CV and Bayes Search CV, by achieving higher validation accuracy and minimizing overfitting. Optuna successfully balanced training and validation performance, making it the most effective tuning method for Logistic Regression in this context. Despite these improvements, overall accuracy remained at 89%, highlighting the importance of further strategies to address performance in minority classes. Cross-validation results reaffirmed the model's stability, with minimal variations across folds for Class 1 metrics but a higher standard deviation for recall on Class 0, indicating sensitivity to data variations.

These results suggest that while the model performs exceptionally well for Class 1, additional measures are needed to improve performance in Class 0. Techniques such as handling data imbalance through oversampling (e.g., SMOTE) or undersampling and adjusting class weights in the loss function can help address this imbalance. Moreover, targeted hyperparameter tuning focusing on increasing recall for Class 0 could further enhance the model's ability to detect conversions within the minority class. The evaluation insights from this process directly inform recommendations for optimizing campaign strategies and improving customer engagement outcomes. By implementing these strategies, the model is expected to balance both classes better, resulting in a more effective and generalized classification system.

4. CONCLUSION

This study successfully achieved its goal of developing optimal strategies for digital marketing campaigns by increasing customer engagement and conversion rates. The analysis showed that factors such as campaign channel (CampaignChannel), campaign type (CampaignType), advertising spend (AdSpend), as well as customer demographic characteristics (age, gender, and income) have a significant influence on key performance metrics such as ConversionRate and WebsiteVisits. Using a comprehensive digital marketing campaign dataset, this study identifies customer behaviour patterns and generates strategic insights that can be used to improve campaign efficiency and return on investment (ROI).

Machine learning model evaluation results show that Logistic Regression is the best model based on the learning curve and overall performance, with 89% accuracy. Logistic Regression showed a good balance between accuracy, recall, and generalisability. The model was also more stable after hyperparameter tuning using the Optuna method, which resulted in optimal parameter configurations such as {'penalty': 'l1', 'C': 294.9938, 'solver': 'saga'}.

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

Hyperparameter tuning successfully improved the detection of the minority class (Class 0), although the improvement was still limited.

As a contribution, this study provides a machine learning-based framework that enables the personalization of customer interactions, more effective segmentation, and more efficient marketing budget management. The results of this study not only reinforce the relevance of digital marketing in the modern era but pave the way for the application of artificial intelligence in creating more adaptive and data-driven marketing strategies.

5. ACKNOWLEDGMENT

The authors thank all the individuals and organizations who contributed to this study. Thanks to Universitas Informatika dan Bisnis Indonesia for providing the necessary resources and support throughout the study. The authors are also thankful to the reviewers and peers for their valuable feedback, which greatly enhanced the quality of this work.

6. AUTHORS' NOTE

The authors declare no conflicts of interest related to this study. All data and methodologies used in this study comply with ethical standards and were transparently conducted. The authors take full responsibility for errors or oversights and welcome constructive feedback from the academic and professional community.

7. REFERENCES

[1] B. Peyravi, J. Nekrošienė, and L. Lobanova, 'Revolutionised Technologies for Marketing: Theoretical Review With Focus On Artificial Intelligence', *Verslas: teorija ir praktika*, vol. 21, no. 2, pp. 827–834, 2020.

[2] Q. Yang, P. Li, X. Xu, Z. Ding, W. Zhou, and Y. Nian, 'A Comparative Study on Enhancing Prediction in Social Network Advertisement Through Data Augmentation', in *2024 4t,'International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, Jun. 2024, pp. 214–218. doi: 10.1109/MLISE62164.2024.10674203.

[3] K. Jin, Z. Z. Zhong, and E. Y. Zhao, 'Sustainable Digital Marketing Under Big Data: An AI Random Forest Model Approach,' *IEEE Transactions on Engineering Management*, vol. 71, pp. 3566–3579, 2024, doi: 10.1109/TEM.2023.3348991.

[4] B. S. Arasu, B. J. B. Seelan, and N. Thamaraiselvan, 'A Machine Learning-Based Approach To Enhancing Social Media Marketing,' *Computers & Electrical Engineering*, vol. 86, p. 106723, Sep. 2020, doi: 10.1016/j.compeleceng.2020.106723.

[5] M. Boujrad and Y. el M. N. lamlili, 'A New Artificial Intelligence-Based Strategy for Digital Marketing Reinforcement', in *Innovations in Smart Cities Applications Volume 4*, M. Ben Ahmed, İ. Rakıp Karaş, D. Santos, O. Sergeyeva, and A. A. Boudhir, Eds., Cham: Springer International Publishing, 2021, pp. 689–699. doi: 10.1007/978-3-030-66840-2_52.

[6] R. Sarath Kumar Boddu, A. A. Santoki, S. Khurana, P. Vitthal Koli, R. Rai, and A. Agrawal, 'An Analysis To Understand The Role Of Machine Learning, Robotics And Artificial Intelligence In Digital Marketing', *Materials Today: Proceedings*, vol. 56, pp. 2288–2292, Jan. 2022, doi: 10.1016/j.matpr.2021.11.637.

[7] M. Marić, O. Grljević, and L. Gluščević, 'Application Of Artificial Intelligence In Digital Marketing', *Anali Ekon fak Subotica*, no. 52, pp. 21–37, 2024, doi: 10.5937/AnEkSub2300033M.

[8] P. van Esch and J. Stewart Black, 'Artificial Intelligence (AI): Revolutionizing Digital

Marketing', *Australasian Marketing Journal*, vol. 29, no. 3, pp. 199–203, Aug. 2021, doi: 10.1177/18393349211037684.

[9] D. Dumitriu and M. A.-M. Popescu, 'Artificial Intelligence Solutions for Digital Marketing', *Procedia Manufacturing*, vol. 46, pp. 630–636, Jan. 2020, doi: 10.1016/j.promfg.2020.03.090.

[10] N. Saud Abd, O. Salim Atiyah, M. Taher Ahmed, and A. Bakhit, 'Digital Marketing Data Classification by Using Machine Learning Algorithms', *IJEEE*, vol. 20, no. 1, pp. 245–256, Jun. 2024, doi: 10.37917/ijeee.20.1.23.

[11] K. Singhal, J. N. Singh, and V. Sharma, 'Enabling Autonomous Digital Marketing: A Machine Learning Approach for Consumer Demand Forecasting', in *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, Feb. 2024, pp. 1903–1908. doi: 10.1109/IC2PCT60090.2024.10486327.

[12] C. Wang, 'Efficient Customer Segmentation In Digital Marketing Using Deep Learning With Swarm Intelligence Approach', *Information Processing & Management*, vol. 59, no. 6, p. 103085, Nov. 2022, doi: 10.1016/j.ipm.2022.103085.

[13] V. Singh, B. Nanavati, A. K. Kar, and A. Gupta, 'How to Maximize Clicks for Display Advertisement in Digital Marketing? A Reinforcement Learning Approach', *Inf Syst Front*, vol. 25, no. 4, pp. 1621–1638, Aug. 2023, doi: 10.1007/s10796-022-10314-0.

[14] M. Zeeshan and K. Saxena, 'Explorative Study of Artificial Intelligence in Digital Marketing', in *Proceeding of the International Conference on Computer Networks, Big Data and IoT (ICCBI - 2019)*, A. P. Pandian, R. Palanisamy, and K. Ntalianis, Eds., Cham: Springer International Publishing, 2020, pp. 968–978. doi: 10.1007/978-3-030-43192-1_107.

[15] Q. Han, C. Lucas, E. Aguiar, P. Macedo, and Z. Wu, 'Towards Privacy-Preserving Digital Marketing: An Integrated Framework For User Modeling Using Deep Learning On A Data Monetization Platform', *Electron Commer Res*, vol. 23, no. 3, pp. 1701–1730, Sep. 2023, doi: 10.1007/s10660-023-09713-5.

[16] Y. Lahbabi, S. Raki, H. C. Lamrani, and S. Dehbi, 'Machine Learning In Digital Marketing.', *MENACIS2021*, Nov. 2021, [Online]. Available: https://aisel.aisnet.org/menacis2021/28

[17] M. S. Ullal, I. T. Hawaldar, R. Soni, and M. Nadeem, 'The Role of Machine Learning in Digital Marketing', *Sage Open*, vol. 11, no. 4, p. 21582440211050394, Oct. 2021, doi: 10.1177/21582440211050394.

[18] G. Gürsoy, A. Varol, and S. Varol, 'Impact of Machine Learning in Digital Marketing Applications', in *2022 3rd International Informatics and Software Engineering Conference (IISEC)*, Dec. 2022, pp. 1–6. doi: 10.1109/IISEC56263.2022.9998211.

[19] Imani, M., & Arabnia, H. R. (2023). Hyperparameter Optimization and Combined Data Sampling Techniques In Machine Learning For Customer Churn Prediction: *A Comparative Analysis. Technologies, 11*(6), 167. https://doi.org/10.3390/technologies11060167

[20] Feroz, S. B., Sharmin, N., & Sevas, M. S. (2024). An Empirical Analysis Of Hyperparameter Tuning Impact On Ensemble Machine Learning Algorithm For Earthquake Damage Prediction. Asian Journal of Civil Engineering, 25(4), 3521–3547.

DOI: https://doi.org/10.17509/coelite.v4i1.81020
p- ISSN 2829-4157 e- ISSN 2829-4149

Journal of Computer Engineering, Electronics and Information Technology (COELITE)

Journal homepage: https://ejournal.upi.edu/index.php/COELITE

Digital Data Security Using a Combination of Base64 Encoding, Rail Fence Cipher, and GZIP Compression

Deden Pradeka[1], Zahra Khaerunnisa[2], Syifa Aqila Humaira[3], Aidha Salsa Billa[4], Murnawan[5], Budiman[6]*

[1,2,3,4]Department of Computer Engineering, Universitas Pendidikan Indonesia, Indonesia
[5]Department of Information Systems, Universitas Widyatama, Indonesia
[6]Department of Information System, Universitas Informatika dan Bisnis Indonesia, Indonesia
Correspondence E-mail: dedenpradeka@upi.edu

ABSTRACT

Digital data security is essential in protecting sensitive information in the current information era. This research implements a hybrid security approach by combining Base64 encoding, the Rail Fence cipher, and GZIP compression. Base64 encoding converts digital files into a standardized text format for easier processing, while the Rail Fence cipher applies transposition to further scramble the encoded data, enhancing confidentiality. GZIP compression reduces file size while adding a layer of complexity, making unauthorized data recovery more difficult. The experiment involved ten trials using files with extensions such as XLSX, DOCX, JPG, PDF, and PPTX. Pearson correlation analysis revealed coefficients close to 0.00 or slightly negative, indicating no significant correlation between plaintext and ciphertext, which signifies strong data randomness and security. Additionally, GZIP compression achieved an average file size reduction of approximately 1 KB, although some smaller files showed minimal compression effects. Overall, the combined method effectively enhances both data security and storage efficiency.

© 2025 Universitas Pendidikan Indonesia

ARTICLE INFO

Article History:
Submitted/Received 22 Mar 2025
First Revised 29 Mar 2025
Accepted 30 Mar 2025
First Available online 1 Apr 2025
Publication Date 1 Apr 2025

Keyword:
Data Security, Base64 Encoding, Rail Fence Cipher, GZIP Compression, Cryptography

1. INTRODUCTION

The development of information technology has had a significant impact on the increase in the flow of digital data transmitted over the internet. This intensive data exchange presents its challenges in maintaining the security and privacy of sensitive information from various threats such as theft, manipulation, or data breaches [1]. This issue is crucial given the increasing number of data security breaches, which result in both material and non-material losses for individuals and institutions.

Various data protection techniques have been developed to safeguard digital information, including encryption, encoding, and data compression methods. However, relying on a single technique alone is sometimes insufficient to ensure optimal security. An integrative solution that combines multiple methods is needed to strengthen security layers against digital data threats. For instance, Base64 encoding has a weakness due to the availability of many online generators [2], making it necessary to further scramble the encoded data using the Rail Fence Cipher method [3]. From another perspective, handling large files can be challenging for transmission or storage, making GZIP compression a suitable alternative [4].

This study proposes a solution that combines three methods: Base64 Encoding, Rail Fence Cipher, and GZIP Compression. Base64 Encoding is used to convert files into a standardized text format, making them more flexible for encryption processing. The Rail Fence Cipher provides additional security through a simple yet effective transposition technique to obfuscate the original message content. Finally, GZIP compression reduces data size while also making unauthorized reading more difficult due to the compression process. By integrating these three methods, this approach is expected to enhance data security and improve efficiency in digital data transmission.

2. METHODS

This research employs the Design and Development (D&D) research method. This method consists of several key stages: problem identification, solution design, solution development, and final product evaluation [5][6], You can see **Figure 1**.

In the initial stage, an analysis is conducted to identify the needs for digital data security and assess the weaknesses of existing methods. Subsequently, the design of the combination of Base64 Encoding, Rail Fence Cipher, and GZIP Compression is systematically structured. The development stage involves implementing this design into an application or prototype that can be tested directly.

Finally, the evaluation process is carried out through a series of tests that assess security effectiveness, file size efficiency, and data processing time. The evaluation results will be used to measure the success of the proposed solution in enhancing digital data security and provide recommendations for future improvements.

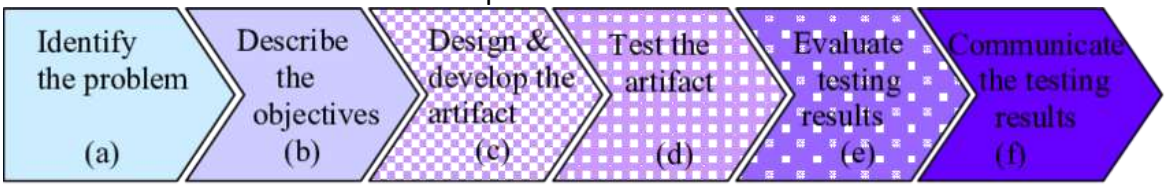


Figure 1. Design and Development (D&D) [7]

2.1. Data Security

Data security is a set of measures, practices, and technologies used to protect digital information from threats such as theft, damage, or unauthorized access [8]. This includes the

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

use of encryption to maintain confidentiality, user authentication to ensure that only authorized individuals can access information, and the implementation of firewalls and antivirus software to protect against cyber threats such as malware or ransomware.

The primary goal of data security is to preserve data integrity, confidentiality, and availability, thereby reducing the risks of information leaks, misuse of personal data, and protecting user privacy. Due to the rapid advancement of information technology and increasingly complex threats, the implementation of strict data security policies and educating users on safe data management practices has become crucial.

2.2. Cryptography

Cryptography is the science and technique of securing information by transforming original data (plaintext) into an unreadable format (ciphertext) through the encryption process, ensuring that only authorized parties with the secret key can revert it to its original form via decryption [9][10].

This technique is used to protect confidentiality, integrity, and authentication, and ensure that data is not accessed or altered by unauthorized entities during transmission or storage, can you see in **Figure 2**. Modern cryptography utilizes complex algorithms, such as symmetric encryption (which uses a single key for both encryption and decryption) and asymmetric encryption (which employs a pair of public and private keys) to provide a high level of security against cyber threats.

With the advancement of digital technology, cryptography has become a vital component of data security in various aspects of life, including financial transactions, electronic communications, and user privacy protection in the digital world.

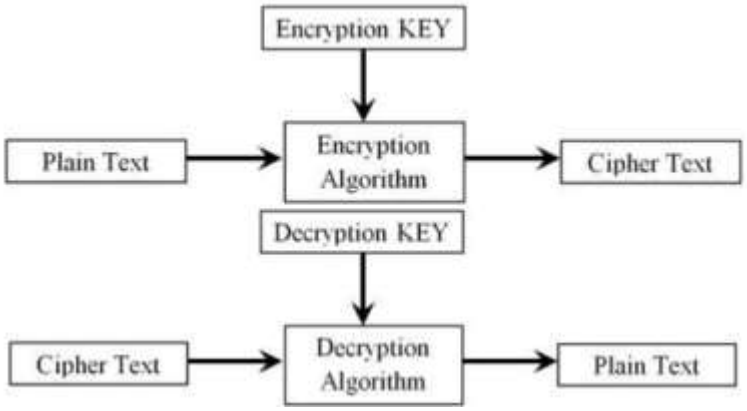


Figure 2. Encryption and Decryption Mechanism

2.3. Data Compression

Data compression is a technique or method used to reduce the size of files or digital information, making storage more efficient and data transmission faster [11]. This process works by eliminating redundancy or unnecessary information, so the remaining data becomes more compact without significantly reducing quality.

There are two main types of data compression:

1. Lossless compression, which allows the original data to be fully restored, as seen in formats like ZIP or PNG.

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

2. Lossy compression, which sacrifices some non-essential information to achieve a higher compression rate, is used in formats like JPEG or MP3.

Data compression is essential in various digital applications, particularly for image, audio, and video storage, as well as internet data transmission, to enhance efficiency, speed, and reduce bandwidth usage.

2.4. Base64

Base64 is an encoding scheme used to convert binary data (such as image, audio, or document files) into a text format consisting of 64 ASCII characters, making it easier to transmit or store through text-based protocols such as email or HTTP [12], as shown in **Figure 3**. Base64 works by dividing data into small blocks (each 3 bytes or 24 bits) and then converting each block into 4 ASCII characters by referring to the Base64 character table [13].

The Base64 encoding steps are as follows:

1. Take the binary data and divide it into 3-byte (24-bit) blocks.
2. Convert each 3-byte block into 4 groups of 6 bits (since 24 bits = 4 groups × 6 bits).
3. Convert each 6-bit group into a corresponding character from the Base64 table, which consists of uppercase A-Z, lowercase a-z, digits 0-9, and the symbols + and /.
4. If the number of bytes is not a multiple of 3, padding characters ('=') are added at the end to complete the block.

The reverse process is called Base64 decoding, which converts Base64 text back into its original binary form by following the opposite steps. Base64 is widely used in internet data transmission, particularly for embedding images in HTML or sending email attachments. However, in this study, the author identifies a weakness in Base64 encoding—the availability of numerous online encoders and decoders, which creates a security vulnerability when someone obtains the encoded data.

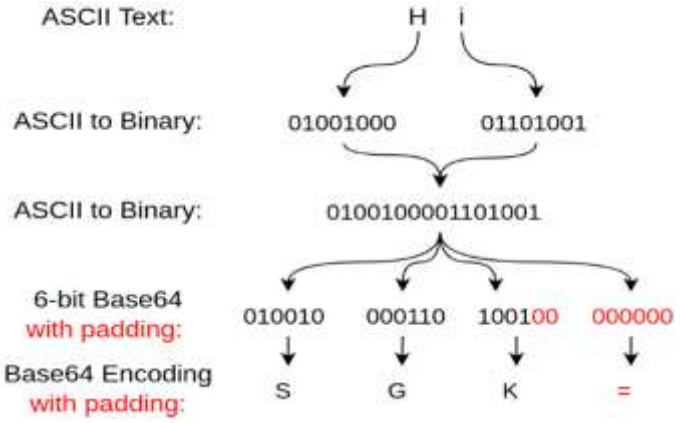


Figure 3. Base64 process [14]

2.5. Rail Fence Cipher

The Rail Fence Cipher is a classical cryptographic method categorized as a transposition cipher, which encrypts a message by altering the position or arrangement of its characters rather than replacing them [15]. This method works by writing the text in a zig-zag pattern across a specified number of rows and then reading it horizontally to produce the ciphertext, in **Figure 4**. Rail Fence Cipher Steps:

a. Encryption:

1. Determine the number of rows (rails) to be used, for example, 3 rails.

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

2. Write the message diagonally in a zig-zag pattern, moving downwards and then upwards, repeating until all characters are placed.
3. After completing the zig-zag pattern, read the text horizontally, row by row, from top to bottom to obtain the ciphertext.
4. Example encryption with the message "KEAMANANDATA" and 3 rails, and then the ciphertext obtained: KMNTEAAADAAA:

```
Rail 1: K . . . M . . . N . . . T .
Rail 2: . E . A . A . A . D . A . A
Rail 3: . . A . . . N . . . A . . .
```

Figure 4. Encryption of a Message through the Rail Fence Cipher Technique

b. Decryption:

1. Determine the same number of rows (rails) used during encryption.
2. Mark the zig-zag positions where the ciphertext characters will be placed.
3. Fill the ciphertext into the marked positions horizontally.
4. Read the message diagonally in a zig-zag pattern to reconstruct the original plaintext.

With this technique, Base64-encoded information is further secured by scrambling the character order. However, this method is relatively simple and not secure enough for critical communications without additional encryption techniques.

2.6 Gzip (GNU zip)

Gzip (GNU zip) is a digital data compression method that uses the lossless DEFLATE algorithm to reduce file size, making data more efficient for storage and transmission [16]. This format is widely used on web servers to accelerate data transfer over the HTTP protocol. Steps in the Gzip Compression Process:

a. Compression (Encoding) Steps:

1. Identify Input Data:
    a) The original data or file (text, images, or other files) to be compressed.
2. Apply the DEFLATE Algorithm:
    b) Analyze the data to detect redundancy (repeated patterns).
    c) Use the LZ77 method (pattern matching) and Huffman coding to significantly reduce file size.
3. Store in Gzip Format:
    a) The compressed data is saved with an added header and checksum for validation during extraction.
4. Generate the Output File:
    1. The resulting file has the extension .gz.

b. Decompression (Decoding) Steps:

1. Retrieve the .gz file.
2. Extract header information to validate data integrity.
3. Use the DEFLATE algorithm to restore the data to its original form without loss of quality (lossless).
4. Recover the original file exactly as it was before compression.

2.7 Pearson Correlation

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

The Pearson correlation, or Pearson Correlation Coefficient, is a statistical measure used to assess the strength and direction of a linear relationship between two numerical variables [17]. However, in cryptography, the best value is one where there is no correlation between the plaintext and the ciphertext. This is because if there is no correlation, it can be said that there is a random transformation from plaintext to ciphertext, which implies enhanced security. Equation 1 presents the Pearson correlation formula, which can be seen in equation 1. Pearson Correlation Formula:

$$r = \frac{\sum (x_i - \underline{x})(y_i - \underline{y})}{\sqrt{\sum (x_i - \underline{x})^2 . \sum (y_i - \underline{y})^2}} \quad (1)$$

Wherex_i and y_i: $are\ the\ i-th\ \ data\ points\ of\ variable\ X\ and\ Y$.
Then $\underline{x}$ and $\underline{y}$: $represent\ the\ mean\ of\ X\ and\ Y$.
Next, r: $is\ the\ koefisien\ korelasi\ pearson\ correlation\ coefficient$.
From the Pearson correlation value, the interpretation is as follows:
if ***r*** = 1, there is a perfect positive relationship (as X increases, Y also increases linearly).
if ***r*** = -1, there is a perfect negative relationship (as X increases, Y decreases linearly). And
if ***r*** = 0, here is no linear relationship between X and Y.

Table 1 provides a comprehensive interpretation of Pearson correlation coefficients, outlining the commonly accepted thresholds used to classify the strength and direction of relationships between variables. This framework allows researchers to evaluate the degree of linear association observed in the data, facilitating more informed statistical inferences and discussions within the context of empirical analysis.

Table 1. General Interpretation of Pearson Correlation Values

Value of *r*	Interpretation
0.90 – 1.00	Very strong (positive) correlation
0.70 – 0.89	Strong (positive) correlation
0.50 – 0.69	Moderate (positive) correlation
0.30 – 0.49	Weak (positive) correlation
0.00 – 0.29	Very weak or no correlation
Negative values	Same interpretation, but in the negative direction

In addition to statistical evaluation, recent advancements in cryptographic methods have demonstrated the effectiveness of combining multiple security layers. For example, Zala and Khan [18] proposed an integrated approach using Base64 encoding and Fernet encryption to protect textual data more robustly. Similarly, Speidel and Manoharan [19] explored individualized puzzle-based learning to enhance conceptual understanding of security mechanisms. From a cybersecurity infrastructure standpoint, Finck and Dohrmann [20] revealed that even more than a decade after the Stuxnet incident, critical systems such as

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

Siemens S7 remain vulnerable, emphasizing the urgent need for layered and adaptive data protection strategies.

3. RESULTS AND DISCUSSION

a. Pearson Correlation Results

Table 2 presents the results of the Pearson correlation analysis based on 10 trials involving various file extensions, including XLSX, DOCX, JPG, PDF, and PPTX, with the Rail Fence cipher applied using a key size of 3. The analysis compares the Base64-encoded plaintext with the ciphertext generated by the Rail Fence cipher algorithm. The findings consistently demonstrate that the average Pearson correlation coefficient is approximately 0.00 or slightly negative (-0.00), indicating a very weak or essentially no correlation between the two datasets. In the field of cryptography, such an outcome is highly favorable, as it signifies a minimal statistical association between the plaintext and the ciphertext. This absence of correlation suggests that the encryption process effectively transforms the plaintext into a randomized sequence that is statistically independent of its original structure, thereby enhancing the unpredictability and security of the ciphertext. The randomness indicated by this lack of correlation contributes to the cipher's resistance to statistical and cryptanalytic attacks, ultimately strengthening the confidentiality and integrity of the protected data.

Table 2. Pearson Correlation Analysis Result

No	File	Rail (key)	Base64	Rail Fence Cipher	Pearson Correlation
1.	excel_one.xlsx	3	UesDB… AAAA	UBBAI… cAA4A	0.0054762160302451
2.	excel_two.xlsx	3	UesDB… AAA==	UBBAI… oI2A=	-0.0015123902837401
3.	jpg_one.jpg	3	/9j/4…B/9k=	/4SRA… d8Vfk	-0.01659774571885
4.	jpg_two.jpg	3	/9j/4… f/9k=	/4RZS… Diomk	-0.0029870957094411
5.	msword_one.docx	3	UesDB… AAA==	UBBAI… wMSA=	0.0023166717367943
6.	msword_two. docx	3	UesDB… AAAAA	UBBAI… IDAMA	0.013524173737449
7.	pdf_one.pdf	3	JVBER… PRgo=	JRLJz… YkoVo	0.00057255410312938
8.	pdf_two.pdf	3	JVBER… FT0Y=	JRLCt… EI0VY	-0.0031939823667947
9.	ppt_one.pptx	3	UesDB… AAAA	UBBAI… EAApA	0.00018997790720693

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

10.	ppt_two.pptx	3	UesDB… WAAAA	UBBAI… MDABA	0.00050871635587214

In the subsequent file compression experiments summarized in **Table 3**, ten trials were conducted using files with various extensions, including XLSX, JPG, DOCX, PDF, and PPTX. The results consistently show that, on average, the file sizes were reduced by approximately 1 KB after applying the Gzip compression algorithm. This reduction demonstrates the capability of Gzip to achieve a modest degree of compression, which can contribute positively to storage efficiency and optimize resource utilization, particularly when processing multiple files or managing limited storage environments.

However, it is noteworthy that in the third trial, no file size reduction was observed following the compression process. This outcome can be attributed to the characteristics of the Gzip algorithm, which may yield minimal or no compression gains when applied to files that are already small or contain limited redundancy. In such cases, the compression overhead may equal or exceed the potential savings, resulting in unchanged file sizes. Despite these limitations, the overall compression performance remains beneficial, especially for larger datasets where redundancy and data patterns offer greater opportunities for compression.

Table 3. The Result of Gzip Compression

No	File Name	Before Extention	Before Compress	After Compress	After Extention
1.	excel_one	.xlsx	82 KB	78 KB	.gz
2.	excel_two	.xlsx	42 KB	41 KB	.gz
3.	jpg_one	.jpg	7 KB	7 KB	.gz
4.	jpg_two	.jpg	183 KB	177 KB	.gz
5.	msword_one	.docx	36 KB	34 KB	.gz
6.	msword_two	.docx	27 KB	24 KB	.gz
7.	pdf_one	.pdf	1.608 KB	1.495 KB	.gz
8.	pdf_two	.pdf	129 KB	121 KB	.gz
9.	ppt_one	.pptx	5.707 KB	5.630 KB	.gz
10.	ppt_two	.pptx	5.559 KB	5.468 KB	.gz

4. CONCLUSION

In conclusion, this study successfully demonstrates the implementation and performance analysis of a hybrid digital data security approach that integrates Base64 encoding, the Rail Fence cipher algorithm, and Gzip compression. The combination of these three techniques offers a multi-layered security framework that enhances data confidentiality, integrity, and storage efficiency. The Pearson correlation analysis reveals that the ciphertext generated by the Rail Fence cipher exhibits an average correlation coefficient of approximately 0.00 or slightly negative when compared to the Base64-encoded plaintext, indicating a statistically insignificant relationship between the two datasets. This absence of correlation reflects the effectiveness of the encryption process in producing a randomized ciphertext that resists statistical and cryptanalytic attacks. Furthermore, the Gzip compression trials show an

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

average file size reduction of approximately 1 KB across multiple file types, including XLSX, JPG, DOCX, PDF, and PPTX. Although some trials demonstrated no change in file size due to the limitations of Gzip when compressing already small or less redundant files, the overall compression performance remains beneficial for optimizing storage resources, particularly for larger datasets.

For future research, it is recommended to explore the integration of more advanced or modern encryption algorithms alongside Base64 and Rail Fence to further enhance the security robustness of the system. Additionally, examining the performance of alternative compression algorithms such as BZIP2, LZMA, or Zstandard may provide better compression efficiency, especially for diverse file types and larger datasets. Future studies may also consider evaluating the proposed method under real-world scenarios involving network transmission, cloud storage, and resistance against various cryptographic attacks, to validate its practical applicability and scalability in modern data security infrastructures.

5. ACKNOWLEDGMENT

The authors would like to express their deepest gratitude to the Computer Engineering Study Program at Universitas Pendidikan Indonesia (UPI) Cibiru Campus for the continuous academic guidance, research facilities, and valuable support throughout the completion of this study. Special appreciation is also extended to Universitas Widyatama and Universitas Informatika dan Bisnis Indonesia (UNIBI) for their collaborative contributions, insightful discussions, and institutional support that greatly enhanced the quality and scope of this research. The constructive feedback and academic resources provided by all parties involved have played a significant role in the successful execution and completion of this research project.

6. AUTHORS' NOTE

All authors contributed equally to the development and completion of this research. The first author is affiliated with the Computer Engineering Study Program, Universitas Pendidikan Indonesia (UPI) Cibiru Campus. The second author is affiliated with Universitas Widyatama, and the third author is affiliated with Universitas Informatika dan Bisnis Indonesia (UNIBI). The authors declare no conflict of interest related to this study. Correspondence concerning this article should be addressed to dedenpradeka@upi.edu

7. REFERENCES

[1] Pradeka, D., Adiwilaga, A., Agustini, D. A. R., Suheryadi, A., & Nuriman, R. (2023). Design and Build an Assessment Platform by Inserting Moodle-Based Cryptographic Methods. Jurnal Nasional Teknologi dan Sistem Informasi, 9(3), 264-270.

[2] Kurniawan, M. S., Putra, I. G. A. S., Maheswara, I. M. A., Labamaking, R. Y. M. N., Listartha, I. M. E., & Saskara, G. A. J. (2023). Analisis Efektivitas Dan Efisiensi Metode Encoding Dan Decoding Algoritma Base64. *Jurnal Informatika Dan Tekonologi Komputer (JITEK)*, *3*(1), 24-34.

[3] Rachmawati, D., Budiman, M. A., & Yusuf, A. (2020, May). Combination of Rail Fence Cipher Algorithm and Least Significant Bit Technique to Secure The Image File. In *IOP Conference Series: Materials Science and Engineering* (Vol. 851, No. 1, p. 012069). IOP Publishing.

[4] Sulistyo, F. P. (2018). Aplikasi Server Web Dengan Kompresi GZIP. *Jurnal Ilmu Teknik dan Komputer*, *2*(1), 2621-1491.

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

[5] Ellis, T. J., & Levy, Y. (2010, June). A guide for novice researchers: Design and development research methods. In *Proceedings of Informing Science & IT Education Conference (InSITE)* (Vol. 10, No. 10, pp. 107-117). Italy, Cassino.

[6] Cotton, W. (2008). Supporting the Use of Learning Objects in The K-12 Environment: A Design-Based Research Project.

[7] Ellis, T. J., & Levy, Y. (2010, June). A Guide for Novice Researchers: Design and Development Research Methods. In *Proceedings of Informing Science & IT Education Conference (InSITE)* (Vol. 10, No. 10, pp. 107-117). Italy, Cassino.

[8] Barmawi, A. M., & Pradeka, D. (2017). Information Hiding Based on Histogram and Pixel Pattern. *Journal of Cyber Security and Mobility*, 397-426.

[9] Pradeka, D., Adiwilaga, A., Agustini, D. A. R., Suheryadi, A., & Nuriman, R. (2023). Design and Build an Assessment Platform by Inserting Moodle-Based Cryptographic Methods. *Jurnal Nasional Teknologi dan Sistem Informasi*, *9*(3), 264-270.

[10] Pradeka, D. (2019). Implementasi Aplikasi Kriptografi Berbasis Android menggunakan Metode Substitusi dan Permutasi. *In Search (Informatic, Science, Entrepreneur, Applied Art, Research, Humanism)*, *18*(1), 161-168.

[11] Aruan, M. C., & Rahayu, W. (2023). Analisis Performa Algoritma Kompresi Data dalam Penyimpanan dan Transfer Data. *LANCAH: Jurnal Inovasi dan Tren*, *1*(2), 228-232.

[12] Wen, S., & Dang, W. (2018, June). Research on base64 encoding algorithm and PHP implementation. In *2018 26th International Conference on Geoinformatics* (pp. 1-5). IEEE.

[13] Mesran, M., Abdullah, D., Hartama, D., Roslina, R., Asri, A., Rahim, R., & Ahmar, A. S. (2018, June). Combination Base64 and Hashing Variable Length for Securing Data. In *Journal of Physics: Conference Series* (Vol. 1028, p. 012056). IOP Publishing.

[14] Red Hat. (n.d.). *Base64 encoding*. Red Hat. Retrieved January 15, 2025, from https://www.redhat.com/en/blog/base64-encoding

[15] Godara, S., Kundu, S., & Kaler, R. (2018). An improved Algorithmic Implementation of Rail Fence Cipher. International Journal of Future Generation Communication and Networking, 11(2), 23-32.

[16] Gailly, J. L., & Adler, M. (1992). Gnu gzip. *GNU Operating System*, 8-18.

[17] Beddolo, C. A., & Budiman, H. (2024). Kriptografi Hill Cipher Menggunakan Matriks Fibonacci. *Jurnal Matematika Komputasi dan Statistika*, *4*(3), 790-800.

[18] Zala, D. K., & Khan, M. A. (2024). *Compressive Method for Securing Text Data Using Base64 Encoding and Fernet Cryptography*. 2024 2nd International Conference on Cyber Security and Digital Forensics. IEEE.https://ieeexplore.ieee.org/document/10625646

[19] Speidel, U., & Manoharan, S. (2022). Strengthening Puzzle-based Learning with Individualization. IEEE Global Engineering Education Conference (EDUCON). https://ieeexplore.ieee.org/document/9820231

[20] Finck, C., & Dohrmann, T. (2023). A Decade After Stuxnet: How Siemens S7 is Still an Attacker's Heaven. Blackhat Conference Europe. https://colinfinck.de/files/EU-23-Finck-A-Decade-After-Stuxnet-How-Siemens-S7-is-Still-an-Attackers-Heaven-wp.pdf

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149