Available online at: https://journal.upgripnk.ac.id/index.php/saintek
Jurnal Pendidikan Informatika dan Sains
15(1), 2026, 16-30
E-ISSN: 2407-1536
P-ISSN: 2089-2802

CYBER INTRUSION DETECTION MODEL USING DEEP LEARNING BASED ON AUGMENTED IMAGE-BASED FEATURE CONSTRUCTION

Mauludil Asri M. Cane*, Kusrini Kusrini, Melwin Syafrizal
[1,2,3]PJJ Magister Teknik Informatika, Ilmu Komputer, AMIKOM Yogyakarta, Jl. Ring Road Utara, Yogyakarta, DI Yogyakarta, Indonesia
*email: mauludil.cane@gmail.com

Received: 2026-02-02 Accepted: 2026-04-15 Published: 2026-04-29

Abstract
Network intrusion detection remains a critical challenge in cybersecurity, particularly due to the increasing volume and complexity of network traffic. To address this issue, this study develops a deep learning framework that transforms tabular NSL-KDD data into image representations using the Lightweight Multi-feature Image Generator for Tabular Data (LM-IGTD). In addition, Homogeneous Noise Generation (HoNG) is applied to enrich feature diversity prior to processing. The transformed data are then classified using a Convolutional Neural Network (CNN) under a binary classification scheme to distinguish between normal and attack activities. Experimental results on the KDDTest+ dataset show that the proposed approach achieves an accuracy of 81.81%, an F1-score of 81.51%, and a ROC-AUC of 95.11%. The results indicate that LM-IGTD significantly contributes to improving the model's ability to distinguish between classes, particularly in terms of ROC-AUC, while HoNG enhances classification performance in terms of accuracy and F1-score. However, a trade-off is observed between improved classification accuracy and the model's probability ranking capability. Overall, these findings highlight that LM-IGTD provides an effective feature representation strategy, while HoNG offers a complementary contribution depending on the evaluation metric prioritized.
Keywords: *Network Intrusion Detection, CNN, LM-IGTD, HoNG, NSL-KDD.*

How to cite (in APA style): Cane, M. A. M., Kusrini, K., & Syafrizal, M. (2026). Cyber intrusion detection model using deep learning based on augmented image-based feature construction. *Jurnal Pendidikan Informatika Dan Sains, 15*(1), 16–30. https://doi.org/10.31571/saintek.v15i1.10499

DOI: 10.31571/saintek.v15i1.10499

INTRODUCTION

In an era where information systems are critical to operational processes, their integrity and accessibility also create vulnerabilities to cyber threats. According to the 2024 annual report from the Indonesian National Cyber and Crypto Agency (BSSN), the total number of anomalous traffic incidents reached 330,527,636. The highest anomaly traffic occurred in December with 112,085,045 incidents, while the lowest was recorded in May with 12,273,078 incidents. These conditions highlight the essential role of Intrusion Detection Systems (IDS) in identifying suspicious activities. Early detection is crucial to prevent potential breaches that may lead to significant losses in data and digital infrastructure (Zou et al., 2020). However, the performance of current IDS approaches is often constrained by the way data is represented. Many existing models still process network traffic features independently in tabular form (Phulre et al., 2024). In contrast, modern cyber attacks exhibit complex patterns that can only be effectively detected when inter-feature relationships are properly captured.

The inability to model these interactions often results in poor detection of novel attack variants (Aslam et al., 2024).

With the advancement of deep learning, Convolutional Neural Networks (CNNs) are no longer limited to image processing tasks but have also been applied to non-visual data through representation transformation. CNNs are particularly effective in extracting spatially localized patterns, making them suitable for modeling inter-feature relationships when tabular data is transformed into image representations (Lara-Abelenda et al., 2025). To enable this capability, an appropriate transformation method is required to convert tabular data into images without losing critical information. The Lightweight Multi-feature Image Generator for Tabular Data (LM-IGTD) addresses this challenge by arranging features based on statistical correlation, thereby placing highly related features in spatial proximity (Lara-Abelenda et al., 2025).

Previous studies have explored tabular-to-image approaches in various domains. For instance, a study on cardiovascular disease prediction introduced LM-IGTD to map tabular features into a two-dimensional space based on their statistical correlations. This arrangement allows CNN models to more effectively extract patterns of inter-feature relationships, leading to improved classification performance. However, most existing approaches directly convert the original feature set without dimensional enhancement. In datasets such as NSL-KDD, which contain only 41 features, direct conversion results in low-resolution images (e.g., 6×6 or 7×7 pixels), limiting the ability of CNNs to capture complex spatial patterns. On the other hand, recent studies leveraging LM-IGTD often rely heavily on deep transfer learning architectures such as ResNet or VGG to compensate for this limitation (Lara-Abelenda et al., 2025). These approaches require substantial computational resources and are less suitable for real-time intrusion detection scenarios that demand efficient inference.

In contrast, this study proposes a novel approach to constructing the feature space in image form to address both dimensional limitations and computational cost. This strategy is inspired by successful applications in the medical domain, particularly in cardiovascular disease prediction, and is adapted to the cybersecurity domain. The key novelty lies in the integration of Homogeneous Noise Generation (HoNG) with LM-IGTD within a lightweight CNN framework. In this context, HoNG functions not merely as a data augmentation technique but as a mechanism to expand the original feature space from 41 to 123 features. This expansion is critical for generating higher-resolution image representations (12×12 pixels), which provide a more suitable spatial structure for convolutional operations compared to direct low-resolution conversions.

Unlike prior studies that treat augmentation and transformation as separate processes, this framework integrates both mechanisms into a unified pipeline to enhance feature representation. With a lightweight architecture, the proposed model is capable of capturing complex inter-feature correlations without requiring excessive computational resources. The effectiveness of the approach is evaluated using standard performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC.

METHOD

Research Design

By adopting a machine learning workflow based on the NSL-KDD dataset, this study detects network traffic anomalies through a series of systematic stages. The core process involves transforming tabular data into image representations using the LM-IGTD technique, which are subsequently used as input for the CNN model. CNN is a deep learning architecture designed to recognize patterns in grid-like data structures, such as images or feature matrices (Alsaedi et al., 2024). Overall, the research framework consists of data acquisition, preprocessing, feature transformation, and model training and evaluation, as illustrated in Figure 1. The final objective of this system design is to perform binary classification to distinguish network activities into “normal” or “attack” categories.

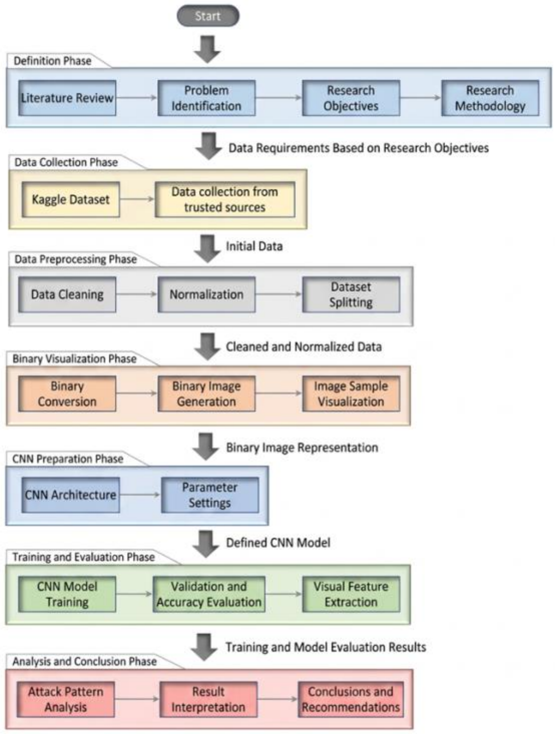

Figure 1. Research Workflow

Dataset Description

This study utilizes the NSL-KDD dataset obtained from the Canadian Institute for Cybersecurity (CIC), University of New Brunswick. The dataset is an improved version of the KDD'99 dataset, which has been refined by removing redundant and duplicate records (Tavallaee et al., 2009). It consists of a training set (KDDTrain+) containing 125,973 samples and a testing set (KDDTest+) comprising 22,544 samples. Out of the total 43 available features, the difficulty attribute is excluded, resulting in 41 input features and one target variable represented by the class label (Malhotra et al., 2021).

Data Analysis Technique

The development of the CNN model, from label initialization to performance evaluation, is carried out in a structured manner as follows:

Data Preprocessing

This stage begins with loading the training and testing datasets, which are integrated with the fieldnames file for attribute labeling purposes (Zou et al., 2020). Subsequently, the classification scheme is simplified into a binary classification by grouping all types of attacks into the "attack" class, while normal traffic is labeled as "normal" (Thana-Aksaneekorn et al., 2024), as presented in Table 1 (Saputra et al., 2025). The difficulty feature is removed as it is not relevant to the classification process.

Table 1. Label Grouping into Five Classes in the NSL-KDD Dataset

DoS	Probe	R2L	U2R	Normal
'apache2', 'back', 'land', 'neptune', 'mailbomb', 'pod', 'processtable',	'ipsweep', 'mscan', 'nmap',	'ftp_write', 'guess_passwd', 'httptunnel', 'imap', 'multihop', 'named', 'phf', 'sendmail', 'snmpgetattack', 'snmpguess',	'buffer_overflow', 'loadmodule', 'perl', 'ps', 'rootkit', 'sqlattack',	'normal'

DoS	Probe	R2L	U2R	Normal
'smurf', 'teardrop', 'udpstorm', 'worm'	'portsweep', 'saint', 'satan'	'spy', 'warezclient', 'warezmaster', 'xlock', 'xsnoop'	'xterm'	

Feature Identification and Encoding

The feature transformation process is performed based on data types. Label Encoding is applied to convert categorical features, including flag, protocol_type, and service, into numerical representations. Meanwhile, numerical features are normalized by scaling the data into the range of [0, 1] using the MinMaxScaler method (Zhao et al., 2024).

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

Equation (1) produces a scaled value, x_{scale}, representing the normalized feature value with a uniform scale. Here, X denotes the original feature value, while min(X) and max(X) represent the minimum and maximum values of the feature, respectively. The normalization process is applied separately to the training and testing datasets to preserve the integrity of the test data and to prevent data leakage (Ambarwari et al., 2020).

Feature Augmentation Using HoNG

This study applies a feature augmentation method using Homogeneous Noise Generation (HoNG) to improve the model's generalization capability by enriching data variability (Bayu Sasongko & Amrullah, 2023). The augmentation parameter is set to k=2, meaning that two synthetic features are generated for each original feature, with a noise standard deviation of 0.01. Gaussian noise is injected into numerical attributes, where the noise is sampled from a Gaussian distribution with a mean of zero and a standard deviation of σ. Mathematically, this process can be expressed as shown in Equation (2).

$$X_{augmented} = X + \epsilon \quad (2)$$

Here, X represents the original feature value, $X_{augmented}$ denotes the augmented feature value, and ϵ is the noise sampled from a normal distribution $N(0, \sigma^2)$. The addition of small-scale noise aims to introduce variability in the data without altering the fundamental characteristics of the feature distribution.

For categorical features, the application of swapping noise generates variation through a random shuffling process within the same feature column. This process involves randomly exchanging category values across samples without modifying the frequency distribution of each category. As a result, the overall category distribution remains consistent with the original dataset, while the combinations of feature values become more diverse (Karahan et al., 2025; Liu et al., 2022).

The augmentation procedure is applied only to the training dataset to generate additional feature variations, whereas the testing dataset remains unchanged to avoid evaluation bias. As summarized in Table 2, the HoNG algorithm generates synthetic features through two primary mechanisms: Gaussian noise injection for numerical features and swapping operations for categorical features. Each original feature produces a number of synthetic features according to the augmentation parameter k.

Table 2. HoNG Algorithm for Feature Augmentation in the NSL-KDD Dataset

Homogeneous Noise Generation

Input
X : Input data (matrix)
F_{names}: Set of feature names
F_{num}, $F_{categorical}$: Sets of numerical and categorical features
k : Augmentation factor (number of new columns generated per feature)
σ : Noise scale
rs : Random State
Do
Set random seed ← rs
Convert X into a DataFrame X_{df}
$noisy_{features}$ ← initialize an empty list
For each feature col in F_{names} do
    For i = 1 to k do
        If col is F_{num} then
            noise ← $N(0, \sigma^2)$
            col_{new} ← $X_{df}[col]$ + noise
        Else If col is $F_{categorical}$ then
            col_{new} ← Shuffle($X_{df}[col]$)
        End If
        $noisy_{features}$ ← Add col_{new}
    End for
End for
$noisy_{df}$ ← concat $noisy_{features}$
$X_{augmented}$ ← Combine X_{df} and $noisy_{df}$
Output
$X_{augmented}$: Final matrix containing original features and all augmented features

With the augmentation parameter set to k=2, each original feature generates two additional synthetic features, resulting in a significant increase in the total number of features from 41 to 123 before entering the LM-IGTD image transformation stage. For numerical features, small-scale noise is applied to ensure that the generated synthetic values remain consistent with the underlying characteristics of the original data distribution. For categorical features, swapping noise is applied by randomly rearranging values within the same feature column without altering the frequency distribution of each category. Through this approach, the overall statistical distribution of the data is preserved, while feature variability is enhanced to enrich the spatial representation used by the model.

Image-Based Feature Representation Using LM-IGTD

Before the image transformation process is performed, the number of features in the dataset changes as a result of the HoNG augmentation stage. A summary of feature dimensionality changes at each stage of data processing is presented in Table 3.

After the HoNG augmentation process generates higher-dimensional features, as summarized in Table 3, the next step is to transform these features into a two-dimensional image representation using the Lightweight Multi-feature Image Generator for Tabular Data (LM-IGTD). The transformation process begins by computing the Pearson correlation matrix among features. Subsequently, features are arranged based on the mean absolute correlation values, ensuring that features with strong statistical relationships are positioned in close spatial proximity within the image representation.

Table 3. Changes in Feature Dimensions at Each Stage of Data Processing

Stage	Operation	Data Dimension
Initial Dataset	NSL - KDD	41 Features
After HoNG Augmentation	k =2	123 Features
Image Size Determination	Equation (3)	S = 12
LM-IGTD Transformation	Feature mapping to image matrix	12 x 12
CNN Input	Grayscale image	1 x 12 x 12

The image dimension is determined dynamically based on the total number of available features. The feature vector is then reshaped into a square matrix of size $S \times S$ pixels, with zero-padding applied when the number of features does not form a perfect square. To determine the image side length S that can accommodate all features N, the ceiling function of the square root of N is used (Golubev et al., 2022), as mathematically defined in Equation (3).

$$S = \lceil \sqrt{N} \rceil \tag{3}$$

CNN Data Preparation

In the CNN-based modeling stage, the transformed image representations are first adjusted to ensure compatibility with the CNN architecture input format. This adjustment is performed by adding a channel dimension to the image data, converting the data shape from (N, H, W) to $(N, 1, H, W)$, where the value 1 represents a single grayscale channel (Yu et al., 2020). Subsequently, all images and their corresponding labels are converted into tensor format using the PyTorch framework. To facilitate efficient and structured training, the image dataset is managed using a custom class, NSLKDDImageDataset, which is integrated with the DataLoader mechanism. A batch size of 64 is employed to improve computational efficiency during the training process.

Model Architecture Design and Configuration

The proposed CNN architecture consists of two sequential feature extraction blocks followed by a classification stage. Each convolutional block employs a 3×3 kernel with a padding size of 1, followed by Batch Normalization, a ReLU activation function, and a MaxPooling operation to reduce spatial dimensions. The output from the convolutional layers is then flattened and passed through a fully connected layer with 128 neurons, along with a dropout rate of 0.5 to mitigate overfitting (Gorle & Guttavelli, 2025). The final layer utilizes a Sigmoid activation function to produce a probability score for binary classification. The complete structure of the implemented model architecture is summarized in Table 4.

Table 4. Proposed CNN Architecture Design

Layer	Kernel Configuration	Activation Function	Output Dimension
Input	Grayscale Image	-	1 x 12 x 12
Conv Block 1	Conv2D (32 filters, 3x3, padding=1) + Maxpooling(2x2)	ReLU	32 x 6 x 6
Conv Block 2	Conv2D (64 filters, 3x3, Padding=1) + Maxpooling(2x2)	ReLU	64 x 3 x 3
Fully Conected	Dense(128) + Dropout(0.5)	ReLU	128
Output	Dense (1)	Sigmoid	1

The model is trained using the Adam optimization algorithm with a learning rate of 0.001 and the Binary Cross-Entropy (BCE) loss function. The training process is conducted over 30 epochs, with continuous monitoring of training loss, validation loss, and accuracy at each iteration to quantitatively evaluate model performance (J. Zhu, 2024).

Model Performance Evaluation

To validate the reliability of the trained model, testing is conducted using a separate test dataset (KDDTest+). Model performance is quantitatively evaluated using a set of standard evaluation metrics, including Accuracy, Precision, Recall, F1-score, and ROC-AUC (Saputra et al., 2025). In addition to numerical metrics, prediction results are also analyzed visually using a Confusion Matrix and ROC Curve to provide a comprehensive view of the distribution of classification errors. This comprehensive evaluation aims to assess the model's generalization capability in distinguishing between "normal" and "attack" network activities.

RESULTS AND DISCUSSION

Feature Transformation and Image Representation Results

As illustrated in Figure 2, the initial stage of this study focuses on converting the NSL-KDD dataset from its tabular form, as shown in Table 5, which presents the first five records out of 125,973 samples. The dataset is subsequently transformed into an image representation to serve as input for the CNN model. The initial characteristics of the data are further analyzed through descriptive statistics presented in Table 6, providing insights into feature distributions, value variability, and class proportions within the dataset. This information is essential for understanding the baseline condition of the data prior to the normalization and augmentation processes. The transformation process is built upon two key mechanisms: the application of Homogeneous Noise Generation (HoNG) to enrich feature diversity, and the use of the Lightweight Multi-feature Image Generator for Tabular Data (LM-IGTD) to construct spatial feature representations (Lara-Abelenda et al., 2025).

Table 5. NSL-KDD Training Data with Input Features and Target Variable

duration	protocol_type	service	flag	…	dst_host_srv_rerror_rate	class
(1)	(2)	(3)	(4)	…	(41)	(42)
0	tcp	ftp_data	SF	…	0,00	normal
0	udp	other	SF	…	0,00	normal
0	tcp	private	S0	…	0,00	neptune
0	tcp	http	SF	…	0,01	normal
0	tcp	http	SF	…	0,00	normal

Table 6. Descriptive Statistics of the Dataset

#	duration	Protocol_type	Service	Flag	…	dst_host_srv_rerror_rate	class
(1)	(2)	(3)	(4)	(5)	…	(42)	(43)
Count	125973	125973	125973	125973	…	125973	125973
Unique	NaN	3	70	11	…	NaN	2
top	NaN	Tcp	http	SF	…	NaN	Normal
Freq	NaN	102689	40338	74945	…	NaN	67343
mean	287,14465	NaN	NaN	NaN	…	0,120240	NaN
Std	2604,51531	NaN	NaN	NaN	…	0,319459	NaN
Min	0	NaN	NaN	NaN	…	0	NaN
25%	0	NaN	NaN	NaN	…	0	NaN
50%	0	NaN	NaN	NaN	…	0	NaN
75%	0	NaN	NaN	NaN	…	0	NaN
Max	42908	NaN	NaN	NaN	…	1	NaN

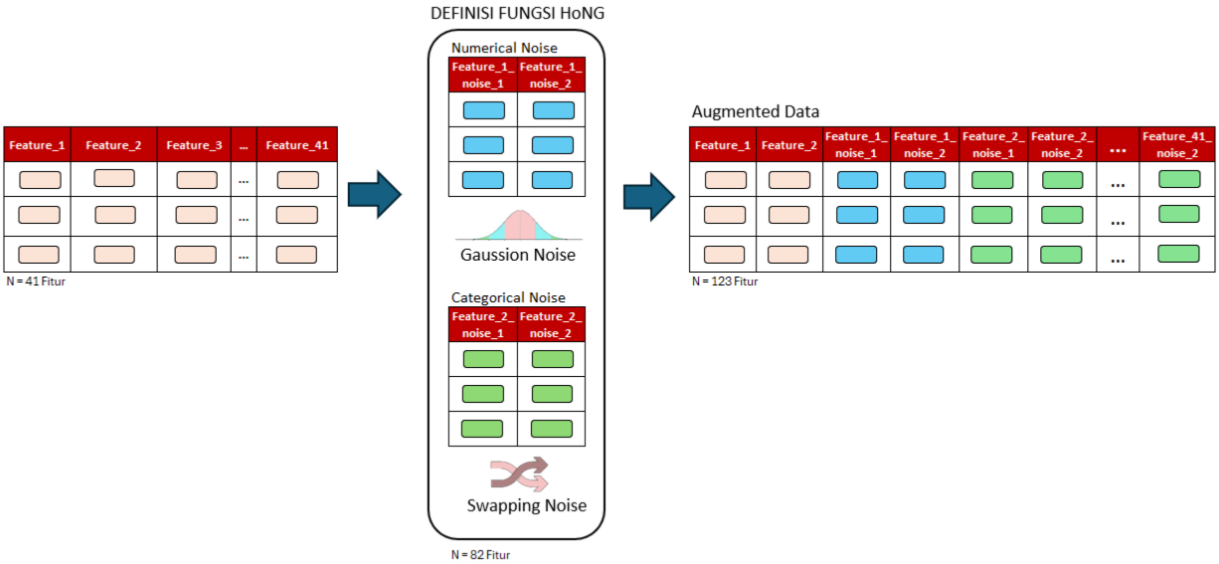

Figure 2. Feature Transformation Process into Image Representation

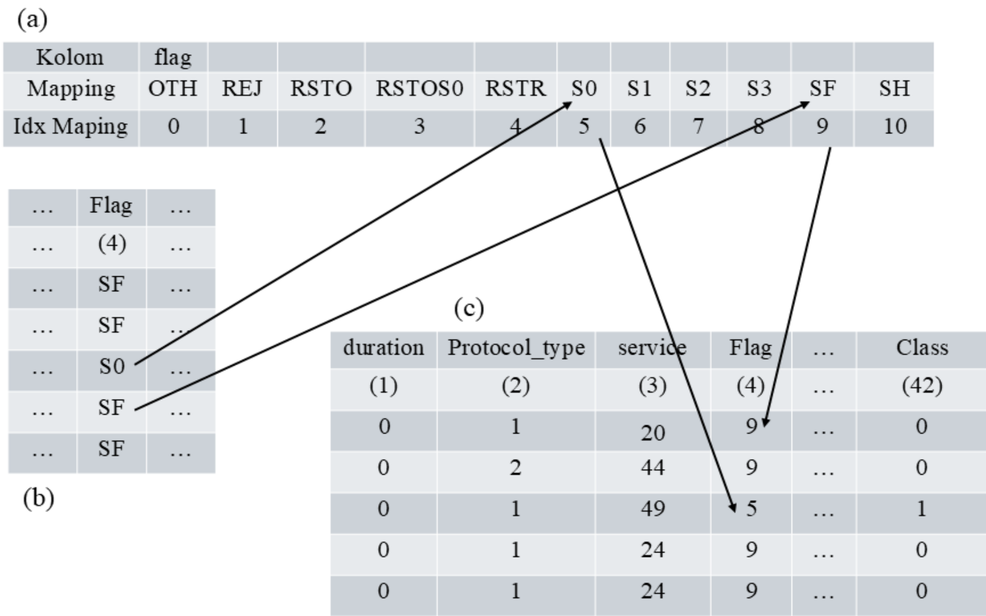

Figure 3. Data Augmentation and Normalization Process with Label Encoding

The augmented features, as illustrated in Figure 3, are subsequently reorganized based on their correlation levels. A Pearson correlation matrix is computed on the training data to determine the feature ordering, where highly correlated features are positioned in close spatial proximity (Alenizy & Berri, 2025). Given a total of 123 features, the resulting image representation is formed as a square matrix of 12×12 pixels, calculated using the ceiling function of the square root of the number of features. A visual example of a single NSL-KDD data sample after the LM-IGTD transformation is presented in Figure 4. The image representation is generated from the processed dataset, which is then ready to be used as input for the CNN model, as shown in Table 7.

Table 7. Augmented Dataset Generated Using the HoNG Algorithm (as Illustrated in Figure 2 and Table 2)

duration	protocol_type	service	…	Duration_noise_1	Duration_noise_2	Protocol_type_noise_1	Protocol_type_noise_2	service_noise_1	service_noise_2	...
0	1	20	…	0.00497	-0.0102	2	1	12	24	...
0	2	44	…	-0.0014	-0.01	1	1	24	24	...
0	1	49	…	0.00648	0.00348	1	1	47	20	...
0	1	24	…	0.01523	0.0053	1	1	49	24	...
0	1	24	…	-0.0023	0.00191	1	1	49	49	...

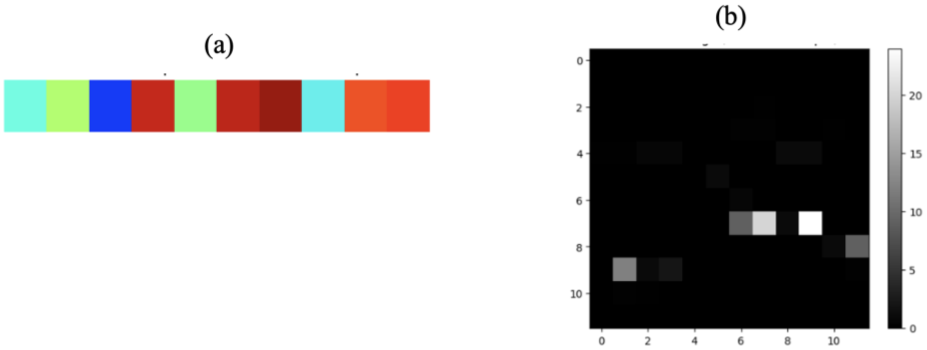

Figure 4. (a) Multicolor feature map (1×10) of highly correlated features; (b) LM-IGTD-based image representation

This transformation demonstrates that LM-IGTD is capable of converting tabular data into a grid-based image representation without losing its structural information. The arrangement of features based on Pearson correlation ensures that features with strong statistical relationships are positioned in adjacent pixels. This design aligns with the fundamental principle of CNNs, which exploit spatial locality to extract meaningful patterns (Yu et al., 2020). Previous studies have shown that grouping correlated features facilitates convolutional kernels in capturing complex attack patterns, which are often embedded within inter-feature interactions that are not readily observable in raw tabular data.

CNN Model Training Performance

The CNN model is trained using the transformed training dataset over 30 epochs, employing the Adam optimization algorithm with a learning rate of 0.001 and an appropriate loss function. The progression of accuracy and loss throughout the training process is illustrated in Figure 5.

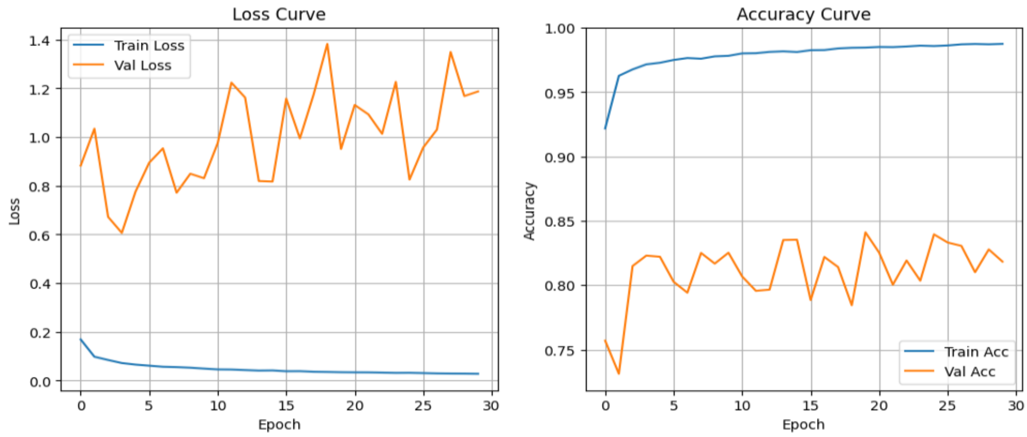

Figure 5. Training and Validation Performance Curves: (a) Loss Convergence; (b) Accuracy Improvement Over Time

Performance Evaluation

The final evaluation is conducted using the KDDTest+ dataset, which consists of unseen data. A comprehensive assessment is performed using standard evaluation metrics, including Accuracy, Precision, Recall, F1-score, and ROC-AUC, along with Confusion Matrix analysis (Y. Zhu et al., 2021). The Confusion Matrix analysis illustrates the distribution of model predictions against the actual classes ("Normal" and "Attack"), as presented in Figure 6.

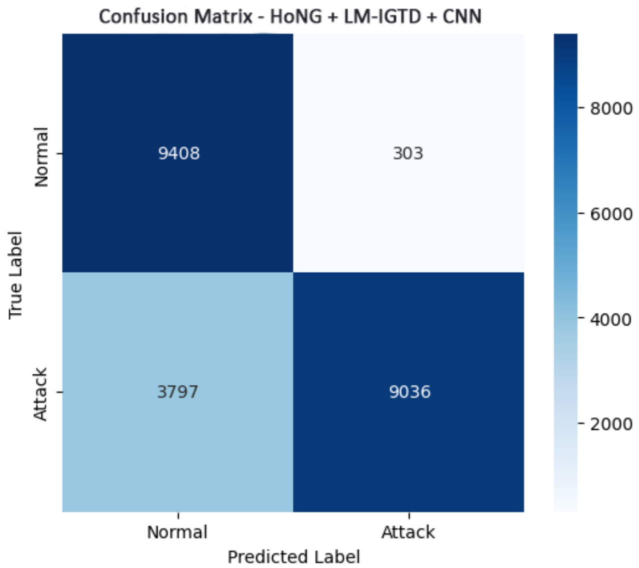

Figure 6. Confusion Matrix of Model Predictions

To evaluate the effectiveness of the proposed method, a comparative analysis is conducted using two baseline scenarios. The first baseline employs a purely tabular-based model, while the second utilizes a CNN model based on LM-IGTD without the inclusion of HoNG augmentation, as presented in Table 8. The results indicate that model performance is influenced by the combination of methods applied. Specifically, LM-IGTD contributes significantly to improvements in ROC-AUC, whereas HoNG plays a key role in enhancing accuracy and F1-score.

Table 8. Model Performance Comparison

Model	Accuracy	F1-Score	ROC-AUC
Baseline (Tabular)	0,8007	0,7928	0,9264
CNN + LM-IGTD	0,8056	0,7992	0,9592
HoNG + LM-IGTD + CNN	0,8181	0,8151	0,9511

The experimental results indicate that the LM-IGTD transformation significantly improves model performance, particularly in terms of ROC-AUC, which increases to 0.9592. This finding suggests that CNN effectively leverages spatial correlations among features when tabular data are represented as images.

On the other hand, the incorporation of HoNG enhances classification performance, achieving an accuracy of 0.8181 and an F1-score of 0.8151. However, a slight decrease in ROC-AUC to 0.9511 is observed, indicating that noise-based feature augmentation may introduce variability that affects the model's ability to rank probabilities accurately. Overall, these results highlight a trade-off between classification accuracy and probability ranking capability, where HoNG improves prediction accuracy but slightly reduces the consistency of probabilistic estimations.

Impact Analysis of LM-IGTD

The experimental results demonstrate that the LM-IGTD transformation provides a significant performance improvement compared to the tabular baseline, particularly in terms of ROC-AUC, which increases to 0.9592. This finding indicates that the CNN model is more effective in capturing inter-feature relationships when tabular data are transformed into image representations. By arranging features based on their correlation, LM-IGTD enables the formation of spatial patterns that can be more efficiently exploited by convolutional operations. This spatial structuring enhances the model's ability to learn complex feature interactions that are not readily captured in raw tabular form.

Impact Analysis of HoNG

On the other hand, the application of HoNG demonstrates an improvement in classification performance, achieving an accuracy of 0.8181 and an F1-score of 0.8151. This indicates that feature augmentation enhances the model's ability to distinguish between classes more effectively. However, a decrease in ROC-AUC to 0.9511 is observed, suggesting that the introduction of noise-based features may increase variability in the predicted probability distribution. This effect implies that while HoNG improves classification accuracy, it may slightly reduce the stability of probability estimation.

Trade-off Analysis of Model Performance

The results reveal a trade-off between classification accuracy and the model's ability to rank probabilities. The model employing LM-IGTD without HoNG achieves the best performance in terms of ROC-AUC, while the model incorporating HoNG demonstrates superior performance in accuracy and F1-score. This finding indicates that noise-based augmentation does not necessarily improve all performance aspects simultaneously, highlighting the need to balance classification precision and probabilistic ranking depending on the intended application.

Discussion

The transformation of tabular data into image representations using LM-IGTD significantly enhances the model's ability to capture inter-feature relationships, particularly as reflected in the improved ROC-AUC performance. This approach addresses the inherent limitations of conventional tabular representations, which typically treat features independently and fail to capture complex non-linear interactions (Phulre et al., 2024; Aslam et al., 2024). By organizing features based on statistical correlations, LM-IGTD constructs a spatial structure that enables CNNs to effectively exploit local connectivity, thereby facilitating the detection of latent attack patterns embedded within feature interactions (Yu et al., 2020; Zhu et al., 2021). These findings are consistent with prior studies demonstrating that tabular-to-image transformations improve deep learning performance by enhancing feature representation quality (Golubev et al., 2022; Wu et al., 2024).

In contrast, the application of HoNG improves classification performance in terms of accuracy and F1-score, indicating that enriched feature variability enhances the model's generalization capability (Bayu Sasongko & Amrullah, 2023; Karahan et al., 2025). However, the observed decline in ROC-AUC suggests that noise-based augmentation introduces variability into the predicted probability distributions, thereby affecting the model's ranking consistency (Liu et al., 2022). This phenomenon reflects the bias–variance trade-off, where increased model flexibility in discrete classification does not necessarily translate into stable probabilistic estimation (Wu et al., 2024). Consequently, the integration of LM-IGTD and HoNG yields distinct performance characteristics depending on the evaluation metric considered.

From a practical perspective, this study demonstrates that effective feature representation design can reduce reliance on complex deep learning architectures, allowing lightweight models to achieve competitive performance (Lara-Abelenda et al., 2025; Gómez-Martínez et al., 2024). The combination of LM-IGTD and HoNG is more suitable for IDS applications prioritizing classification accuracy, whereas LM-IGTD without augmentation is preferable for probability-based systems such as risk scoring (Saputra et al., 2025). Nevertheless, the use of the NSL-KDD dataset as a benchmark remains a limitation, as it does not fully reflect the complexity of modern network traffic. Therefore, further evaluation using more representative datasets and real-time scenarios is necessary to comprehensively assess the model's generalization capability (Tavallaee et al., 2009; Wu et al., 2024).

Research Implications

These findings suggest that the selection of methods should be aligned with the specific objectives of the intrusion detection system. When the primary goal is to achieve strong global class separation, LM-IGTD without augmentation provides optimal performance. In contrast, when the priority is to enhance classification accuracy, the combination of HoNG and LM-IGTD serves as a more effective alternative.

CONCLUSION

This study evaluates the effectiveness of transforming NSL-KDD tabular data into image representations using the LM-IGTD method, as well as the impact of feature augmentation through HoNG in a CNN-based intrusion detection system. The experimental results demonstrate that LM-IGTD significantly improves class separability, achieving a ROC-AUC of 0.9592, which is higher than the tabular baseline of 0.9264. This indicates that image-based representations enable CNN models to more effectively capture spatial correlations among features.

The incorporation of HoNG further enhances classification performance, achieving an accuracy of 0.8181 and an F1-score of 0.8151. However, a decrease in ROC-AUC to 0.9511 is observed, suggesting that noise-based augmentation may affect the stability of probabilistic estimation. These findings reveal a trade-off between classification accuracy and probability ranking capability. Accordingly, method selection should be aligned with system objectives: LM-IGTD without HoNG is more suitable for global class separation, whereas the combination of HoNG and LM-IGTD is more effective for improving classification accuracy. Overall, this study confirms that the tabular-to-image approach based on LM-IGTD is an effective strategy for enhancing CNN-based intrusion detection systems, with HoNG serving as a contextual augmentation technique depending on the targeted performance metric.

Future Work

Future research may focus on developing more adaptive augmentation methods that account for underlying data distributions, thereby improving model performance without compromising the stability of probabilistic estimation. In addition, further exploration of more effective feature arrangement strategies in the tabular-to-image transformation process could enhance the quality of spatial representations. The development of more advanced deep learning architectures, such as attention-based models or hybrid CNN–Transformer frameworks, also represents a promising direction for maximizing the potential of image representations derived from tabular data. Moreover, evaluation on more diverse intrusion detection datasets and real-time scenarios is necessary to assess the model's generalization capability more comprehensively. Finally, integrating explainable artificial intelligence (XAI) approaches would improve model interpretability, enabling more transparent and reliable decision-making in practical cybersecurity applications.

REFERENCES

Badan Siber dan Sandi Negara. (2024). *Publikasi laporan tahunan monitoring keamanan siber*. https://www.bsn.go.id/monitoring-keamanan-siber

Alenizy, H. A., & Berri, J. (2025). Transforming tabular data into images via enhanced spatial relationships for CNN processing. *Scientific Reports, 15*(1). https://doi.org/10.1038/s41598-025-01568-0

Alsaedi, M., Ghaleb, F. A., Saeed, F., Ahmad, J., & Alasli, M. (2024). Multi-modal features representation-based convolutional neural network model for malicious website detection. *IEEE Access, 12*, 7271–7284. https://doi.org/10.1109/ACCESS.2023.3348071

Ambarwari, A., Adrian, Q. J., & Herdiyeni, Y. (2020). Analisis pengaruh data scaling terhadap performa algoritme machine learning untuk identifikasi tanaman. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 1*(3), 117–122.

Aslam, M., Tufail, A., Apong, R., Silva, L., & Taqi Raza, M. (2024). Scrutinizing security in industrial control systems: An architectural vulnerabilities and communication network perspective. *IEEE Access*, 1. https://doi.org/10.1109/ACCESS.2024.3394848

Bayu Sasongko, T., & Amrullah, A. (2023). Analisis efek augmentasi dataset dan fine tune pada algoritma pre-trained convolutional neural network (CNN). *Jurnal Teknologi Informasi dan Ilmu Komputer, 10*(4), 763–768. https://doi.org/10.25126/jtiik.2023106583

Golubev, S., Novikova, E., & Fedorchenko, E. (2022). Image-based approach to intrusion detection in cyber-physical objects. *Information, 13*(12). https://doi.org/10.3390/info13120553

Gómez-Martínez, V., Lara-Abelenda, F. J., Peiro-Corbacho, P., Chushig-Muzo, D., Granja, C., & Soguero-Ruiz, C. (2024). LM-IGTD: A 2D image generator for low-dimensional and mixed-type tabular data to leverage the potential of convolutional neural networks. *arXiv*. https://arxiv.org/abs/2406.14566

Gorle, R., & Guttavelli, A. (2025). Enhanced image tampering detection using error level analysis and a CNN. *Engineering, Technology and Applied Science Research, 15*(1), 19683–19689. https://doi.org/10.48084/etasr.9593

Karahan, O., Ataşlar-Ayyıldız, B., & Ayyıldız, P. (2025). Network intrusion detection system using a hybrid deep learning model with swarm intelligence-based hyperparameter optimization. *The Journal of Supercomputing, 81*(15), 1346. https://doi.org/10.1007/s11227-025-07802-w

Lara-Abelenda, F. J., Chushig-Muzo, D., Peiro-Corbacho, P., Gómez-Martínez, V., Wägner, A. M., Granja, C., & Soguero-Ruiz, C. (2025). Transfer learning for a tabular-to-image approach: A case study for cardiovascular disease prediction. *Journal of Biomedical Informatics, 165*, 104821. https://doi.org/10.1016/j.jbi.2025.104821

Liu, X., Zhu, S., Yang, F., & Liang, S. (2022). Research on unsupervised anomaly data detection method based on improved automatic encoder and Gaussian mixture model. *Journal of Cloud Computing, 11*(1), 58. https://doi.org/10.1186/s13677-022-00328-z

Malhotra, L., Bhushan, B., & Singh, R. (2021). Artificial intelligence and deep learning-based solutions to enhance cyber security. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3833311

Phulre, A. K., Jain, S., & Jain, G. (2024). Evaluating security enhancement through machine learning approaches for anomaly-based intrusion detection systems. In *2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)* (pp. 1–5). https://doi.org/10.1109/SCEECS61402.2024.10482161

Saputra, F. H., Ilham, I., Rizal, M., Wisda, W., Wanita, F., Mursalim, M., & Fadillah, A. (2025). Enhancing intrusion detection using random forest and SMOTE on the NSL-KDD dataset. *Journal of System and Computer Engineering (JSCE), 6*(3), 240–247. https://doi.org/10.61628/jsce.v6i3.2056

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD Cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1–6). https://doi.org/10.1109/CISDA.2009.5356528

Wu, Y., Zou, B., & Cao, Y. (2024). Current status, challenges, and future trends of deep learning-based intrusion detection models. *Journal of Imaging, 10*(10). https://doi.org/10.3390/jimaging10100254

Yu, C., Han, R., Song, M., Liu, C., & Chang, C. I. (2020). A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial-spectral fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13*, 2485–2501. https://doi.org/10.1109/JSTARS.2020.2983224

Zhao, X., Leng, X., Wang, L., & Wang, N. (2024). Research on fine-tuning optimization strategies for large language models in tabular data processing. *Biomimetics, 9*(11). https://doi.org/10.3390/biomimetics9110708

Zhu, J. (2024). Research on software vulnerability detection methods based on deep learning. *Journal of Computing and Electronic Information Management, 14*(3), 2413–1660.

Zhu, Y., Brettin, T., Xia, F., Partin, A., Shukla, M., Yoo, H., Evrard, Y. A., Doroshow, J. H., & Stevens, R. L. (2021). Converting tabular data into images for deep learning with convolutional neural networks. *Scientific Reports, 11*(1), 11325. https://doi.org/10.1038/s41598-021-90923-y

Zou, D., Wang, S., Xu, S., Li, Z., & Jin, H. (2020). VulDeePecker: A deep learning-based system for multiclass vulnerability detection. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/TDSC.2019.2942930