Implementing Comparative Analysis on Feature Engineering Techniques and Multi-Model Evaluation Framework for IDS

Neha Sharma[1,*], Abhishek Kajal[1]

[1]Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, 125001, Haryana, India

Emails: nehasharma31066@gmail.com; drabhishekkajal@gmail.com

Abstract

In recent years, most of the current intrusion detection methods run for critical information infrastructure are tested for IDS datasets, but does not provide desired protection against emerging cyber- threats. Most machine and deep learning-based intrusion detection methods are inefficient on networks due to their high imbalanced or noisy IDS datasets. Therefore, in this paper, our proposed work implements a comprehensive framework, using multiple models of machine learning and deep learning by taking advantage of advanced feature engineering approaches. Our research explores the impacts of a variety of feature engineering approaches on dimensionality reduction methods used to train and test model performance with execution time taken on the CICIDS2017 dataset to reduce the time complexity and enhance performance to detect intrusion by experiment and leveraging feature engineering techniques like PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), t_SNE (t-Distributed Stochastic Neighbor Embedding), and Autoencoders. This framework also resolves the class imbalance issues by using SMOTE (Synthetic Minority Oversampling Technique), generates synthetic samples of those classes, which have a very low number of samples to balance the class for a better model performance. Our comparative analysis is performed on metrics like accuracy, training time and memory usage for machine learning models like Gradient Boosting, Logistic Regression, XGBoost and deep learning models. DL with LDA feature engineering approach achieved the highest test accuracy of 95.99% and Gradient Boosting shows strong performance by attaining a high-test accuracy of 90.8%. Illustrated DL model had higher memory usage, but LR and XG- Boost models performed computationally efficient. Further, it is observed that LDA performed better with ML and DL models in comparison to other feature engineering techniques to enhance the intrusion detection efficiency.

Keywords: PCA; LDA; t_SNE; Autoencoder; ML and DL; IDS

1. Introduction

For protecting the critical information infrastructure, intrusion detection system becomes increasingly important, particularly cyber-threats rapidly grows more while conventional detection systems demonstrate effective against known threats, they generally struggle to recognize new and complex attacks on cybersecurity. However, recent advancements of feature engineering, ML and DL are transforming IDS capabilities [1], particularly in dynamic environments like critical infrastructure and Internet of things (IoT), where resource and IDS datasets like CICIDS2017, CICIDS2018, CICIDS2019 etc., and limitations are a continuous challenge for the performances of IDS.

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

Challenges for IDS in cyber-security

In the field of cyber-security, many cyber-threats growing every day on networks. We need to detect this intrusion by applying various techniques on many IDS datasets, which are publicly available, but in the raw form or have noisy and with many other shortcomings to detect the intrusion efficiently. Few major challenges are mentioned below:

High-Dimensional Data

As datasets, become larger and more complex, lead to increase the difficulty for selecting and transforming relevant features from the raw dataset because High-dimensional data always lead to the "curse of dimensionality," and models suffer from overfitting, underfitting or poor generalization.

Data Imbalance

Class imbalance in the raw datasets more challenging, particularly in classification tasks. Techniques like Synthetic Minority Over-sampling Technique (SMOTE), random oversampling etc., [2] can be used to make the data balanced, because balancing features is the most important challenge across different classes.

Domain Expertise

Domain expertise always requires deep domain knowledge for effectiveness of feature engineering in IDS [3], which may not always be available. This limitation can delay the process of feature engineering, especially in critical information infrastructures such as finance and healthcare sectors.

Feature engineering (FE) is a crucial element in the machine learning (ML) pipeline that knowingly affects the performance of models. It includes selecting, transforming, and generating new features from original data, securing the input to ML techniques [4], which is more informative and relevant. Historically, FE has been a physical process driven by domain expertise, but recent advances in automated FE and deep learning (DL) [5] have changed the area.

Importance of Feature Engineering

Even the most complex models may function poorly if they lack the necessary features. Feature engineering connects raw data to the desired output, allowing algorithms to find underlying patterns [6], correlations, and hidden structures to train the ML and DL model effectively to enhance the performance of IDS to strengthen the critical infrastructure.

Feature Engineering Techniques

Feature Selection

The feature selection [7], is a process of extracting the more informative features from a raw dataset in order to minimize model complexity and enhance interpretability. Some important methods of feature selection are discussed as:

Filtration Methods

Filter methods according to attributes of statistical criteria like mutual information or correlation coefficients. While these strategies are fast and scalable, they may fail to capture feature interactions.

Wrapper Methods

In wrapper, RFE (recursive feature elimination) use the models of machine learning to evaluate feature's subsets and choose the one that yields the best performance. Wrapper methods is lead to better results but more computationally expensive.

Embedded Methods

During the model training process, embedded methods perform feature just like we seen in Lasso regression or decision trees algorithms. Embedded method integrated into model development and computationally efficient.

Feature Transformation

In this technique, convert data into formats that are easier for machine learning [8] models to process.

Scaling and Normalization

We have some Scaling approaches, such as Standard-scalar and Min-Max scaling to ensure that features have the same

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

range or distribution, therefore increasing model convergence in feature-sensitive algorithms, like neural networks.

In this paper, we propose a feature engineering importance that leverages the power of machine and deep learning algorithm to enhance the performance of intrusion detection system in critical information infrastructure in terms of performance metrics like recall, precision, f1-score and accuracy or computation time. Our approaches are rooted in the development of CII to detect cyber-threat, which integrate the several feature engineering techniques, we are using pre-processing in our proposed work like scaling, normalization, imputation for null values, standardization and etc., and employs sophisticated machine and deep learning techniques to enhance the detection accuracy and reduced the computation time.

The combination of various feature engineering technique allows for a more comprehensive analysis of required or more compactable features according to our objectives, and reducing the computational time for detecting the lively cyber-attacks and then apply the models of ML and DL to analysis the effects of feature engineering techniques as a result in term of performance metrics. Machine leaning models are utilized to identify patterns of attacks and anomalies that are indicative of IDS, while deep leaning algorithms handle huge and complex datasets with their capability. Further, improve the system's performance with feature engineering approaches.

We proposed feature-engineering importance to set a benchmark for intrusion detection system in critical in- formation infrastructure by offering improve the performance of ML and DL models for performance metrics. By using the strength of feature engineering, we address the limitation models of without feature engineering and provide the solution of these limitations.

This research not only contribute to the field of cyber-security but also uses in all other fields in which we use ML and DL models to enhance the system performance in every field. Feature engineering apply in every field to remove the unnecessary feature or limitations of the raw dataset. It refines the raw data by apply FE approaches and make the data clean by pre-processing techniques on the datasets. By applying FE, extract the needed or required features and also create new features according to our objectives with the use of data augmentation and also balance the unbalanced or raw dataset with the help of SMOTE, random oversampling and etc., Along all above, this paper contributes to the enhancement of datasets, our research goals to establish a more secure and reliable critical information infrastructure for IDS.

2. Background Information

To detect the intrusion on the networks, feature engineering is an important part to train the clean or accurate data with the machine learning [1] models and it plays a vital role in determining model performance to enhance the detection rate for ever-growing cyber-threats. FE includes the processes of selecting, transforming, and creating new features from raw datasets to ensuring the input for machine learning [9] algorithms is both relevant and informative [10]. Usually, this process depends on domain expertise, with researchers manually identifying significant features. Still, advancements in automated feature engineering [11] and deep learning have reshaped this domain, introducing new methodologies that improve both efficiency and effectiveness.

Research Problem

A. Research Objectives

- Implement several feature engineering approaches and their impact on performance in terms of accuracy, training time and memory usage.
- Evaluate the performance on ML and DL models with various feature-engineering techniques.
- Find out the challenges associated with feature engineering techniques and lay down a roadmap for future research works.

Significance of the Research

This research embraces significant value for multiple reasons in the context of IDS. Firstly, searching out to provide a complete understanding of several feature engineering methods and their importance across varied domains, in that way linking the gap between raw data and meaningful information's. It provides real-world use of framework in IDS for researchers to evaluate multiple models under several feature-engineering methods and addressing the effectiveness of SMOTE for balancing IDS datasets with skewed distributions, also highlights the trade-off between computational efficiency, memory overhead and accuracy in ML and DL models. These study goals to implements how these novelties can revolutionize the building process of model and improve performance to detect intrusion.

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

Moreover, addressing the challenges with FE will participate to the development of more robust ML models, ultimately enhancing outcomes and decision-making in critical applications, including cybersecurity for IDS etc.

3. Literature Review

To successfully address the ever-changing variety of cyber threats, organizations must develop robust and adaptable intrusion detection systems (IDS). Recent research has highlighted the importance of feature engineering techniques in improving IDS effectiveness by simplifying the finding of relevant patterns from network data.

This focuses are important for ensuring the safety and adaptability of interconnected systems in the face of cyber challenges with the help of literature review, which was produced findings from multiple previous work, especially on methodologies, models, and specific feature engineering techniques used, and after that Combining all these understanding allows us to gain a deeper knowledge of today's situations of IDS, also the continued need for advanced techniques to securing critical information infrastructure due to this, researchers stresses on critical areas for future research. IDS are becoming increasingly crucial in protecting critical information infrastructure, particularly as cyberattacks produce more cultured. While conventional detection systems have effective demonstrated against identified threats, but they frequently struggle to recognize new and complicated attack techniques. But, current advancements in FE, ML, and DL are transforming IDS capabilities, especially in dynamic environments like critical information infrastructure and IoT, where resource limitations are a constant challenge.

In an innovative study by Sarhan et al. (2024), the authors highlight innovative feature extraction methodologies specifically designed for IoT networks. Their work achieved an impressive accuracy of 94%, demonstrating how tailored approaches can effectively manage the unique demands of these environments. This research sets a compelling example for the importance of adaptability in IDS design. Similarly, Abdullah et al. (2021) tackled the complexities of network traffic analysis by developing context-aware algorithms that reached an accuracy of 88%. Their findings emphasize the necessity for real-time adaptability, displaying how responsive systems can significantly enhance intrusion detection capabilities.

Another critical advancement in optimizing IDS is dimensionality reduction [12]. Researchers such as Musleh et al. (2023) and Lin et al. (2023) have shown that reducing computational overhead while maintaining robust detection capabilities is vital for IoT systems. Their studies achieved accuracies of 92% and 93%, respectively, illustrating a successful balance between resource constraints and high detection rates.

Ensemble learning methods [13] further strengthen IDS effectiveness by combining various algorithms to handle complex attack patterns. Nguyen et al. (2021), Das et al. (2022) and Saha et. al (2022) explored ensemble-based feature selection and ML models, achieving accuracies of 90% and 91%. Their research underscores how hybrid systems can enhance scalability and precision in multifaceted cybersecurity scenarios.

The impact of deep learning on IDS is profound, particularly through innovations like data augmentation. Mohammad et al. (2024) and Fatani et al. (2022) have demonstrated the power of DL in detecting a wide range of cyberattacks, achieving remarkable accuracies of 95%. These studies highlight the interaction between deep learning techniques and robust feature engineering, setting new benchmarks for IDS performance.

As IoT-specific solutions evolve, they continue to address the unique challenges faced in intrusion detection. Niu et al. (2022) proposed a real-time feature generation algorithm that achieved 93% accuracy, specifically tailored to scale with IoT applications. Building on this foundation, Lin et al. (2023) supported for edge computing as a reflects to decentralize IDS, enhancing both speed and scalability by processing data closer to its source.

Despite these promising advancements, significant challenges persist. Many feature-engineering methods demonstrate high efficacy in controlled settings but often falter when applied to real-world scenarios. Zare and Mahmoudi-Nasr (2023) pointed out the complexities introduced by various attack vectors, highlighting the need for practical applicability in dynamic environments. Moreover, the detection of minority and emerging attack types remains a pressing concern; Robinson et al. (2024) argued for refining feature selection and ensemble methods to bridge these gaps.

The literature constantly stresses the need for continuous advancement in FE and methodologies to adapt IDS to growing threats. Future research must arrange real-time adaptability, scalability, and the combination of cutting-edge AI techniques to ensure that IDS remain a robust defense mechanism against an ever-changing cybersecurity landscape.

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

4. Research gap

The ongoing research on FE for IDS in critical information infrastructure and IoT networks discloses numerous significant gaps that drive farther survey. Several campaigns, such as those presented by Sarhan et. al (2024) and Zare et. al (2023) just focus on specific approaches while failing to examine their adaptation to dynamic contexts and cyber threats. Further, research works illustrated by Zare et. al (2023) and Musleh et. al (2023) demonstrate good performance, but significantly, lack real-time process capabilities required for large-scale IoT applications. Moreover, performance metric unit typically on accuracy, as experience of all works by Mohammad et. al (2024) and Saha et. al (2022), ignore previous important metrics like recall and precision. There is a lack of integration of multifaceted feature engineering, and the variety of attack scenarios that are frequently expressed reduces the practical usefulness of finding. Furthermore, the impact of feature selection demonstrated by Tibshirani et. al (1996) on IDS performance was neither fully explored, nor the generalizability of approaches across various network topologies evaluated in IDS. So, most of the existing works only focuses on single model and limited feature engineering approaches. Our study bridges the gap by applying multiple Feature Engineering techniques simultaneously and evaluating their impacts on multiple ML and DL models. Addressing these gaps could significantly enhance the robustness and effectiveness of IDS proposed by Ketepalli et. al (2023) in the growing cyber-attacks on networks.

In spite of the importance of feature engineering, many models fail to achieve optimal performance due to insufficient feature selection as presented by Natarajan et. al (2023) and Albulayhi et. al (2022). This problem is mainly noticeable in large and complex datasets of IDS, where the bulk of information can lead to challenges such as class imbalance and high dimensions. Besides these, the dependance on domain knowledge can create issues to effective feature engineering, especially in particular sectors. So, there is a persistent necessity to explore and address these challenges, while taking advantages of recent advancements in automated techniques and deep learning to enhance feature engineering practices proposed by Guo et. al (2019) and Kanter et. al (2015) for IDS.

5. Methodology

This work illustrated a method to examine the efficiency of different feature engineering approaches for intrusion detection systems designed for critical information infrastructures (CIIs).

A. Research Design

This work uses a systematic research design to discover the efficiency of FE methods to improve IDS within CII.

In this paper, we start our proposed work with loading multiple csv files into Pandas Data-Frames [9] and make use of randomly sampling 5% for each file because it helps to manage memory and computational limitation of publicly available large CICIDS2017 dataset.

After sampling, we concatenate all data-frames into single Data-Frame (df), so as it become working dataset for further analysis, that finally provides us to work with 28307 rows and 79 features.

The dataset may have some infinite values or missing values (-np.inf and np.inf). These infinite or missing values are replaced with NaN, and filled with the mean value of each column for imputation to ensure model compatibility and data integrity as missing data can distort the analysis and result in errors during train the model. By applying imputation strategies, we ensure the model get a clean dataset.

Further, in this work, we apply class filtering, and removed the classes that have fewer than 10 samples from dataset, because classes with very few samples can lead to overfitting [19]. Overfitting can cause poor generalization to the model. In addition, class filtering is important for balancing the class distribution as it ensures that each class has sufficient representation to help in preventing bias in the learning process of the models.

Next step is split our dataset into training and testing sets with the 80/20 ratio split. This allows the one portion (80) use for training our model and another portion (20) use for testing its performance to ensure the results to be unbiased.

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

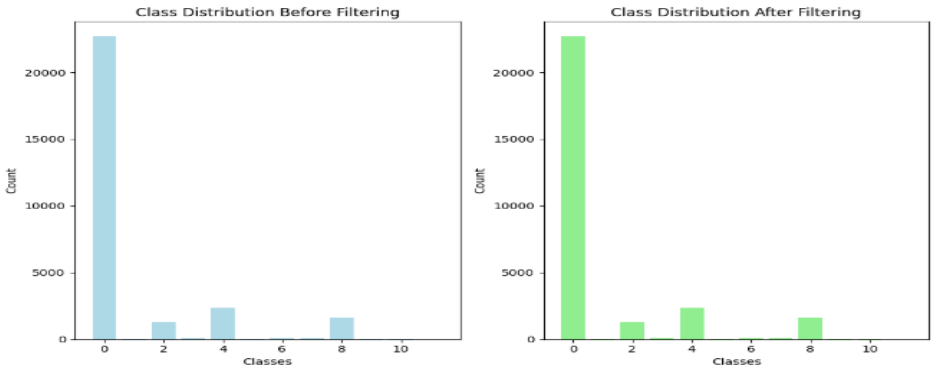

Figure 1. Class distribution before and after Filtering

Standard-Scaler is used as feature scaling approach to ensure that each feature has a mean of 0 and a standard deviation of 1. Feature scaling is important particularly for Logistic Regression like algorithms because features with different scales can cause poor model performance. Further, we implement SMOTE technique for class imbalance, followed by execution of dimensionality techniques like PCA, LDA, t_SNE and Autoencoder [25] to evaluate the performance of models.

This technique permits for the identifications of current research gaps and a considerate of how IDS can be improved to report the exclusive challenges posed by cyber threats in critical sectors such as finance, transportation and energy. The comparative nature of this research design goals to substitute a deeper knowledge of the complications involved in safeguarding CII.

B. Data Collection Methods

In this work, data is collected based on a detailed analysis of knowledgeable articles and research papers obtained from credible academic databases, industry reports, and government publications. Selected references include studies by [14] and [27] deliver a complete summary of FE approaches related to IDS deployment in CII environments. Each work is examined for its methodological consistency, and real-world suggestions, applied models confirming a robust knowledge of the recent scenery of IDS research. This qualitative data collection method allows a complex viewpoint on the challenges and progresses in securing critical information systems.

C. Data Analysis Techniques and Discussion

In this paper, we apply LDA (Linear Discriminant Analysis) for feature engineering, as LDA is a leading dimensionality reduction [13] technique that is used to project data into a lower dimensional space and maximizing the separation between classes. LDA can also help to reduce overfitting by eliminating less informative features. LDA is important for feature selection and dimensionality reduction [12] when there are many features (high-dimensional data). It helps to highlight the most important features for classification and leading to faster and potentially more accurate models. Our models are trained using LDA-transformed features, which helps in evaluating how feature engineering affects models' performance.

$$LDA: X_{lda} = XW \quad (1)$$

In Eq. (2), X denotes Input data matrix, W indicates matrix that maximizes class separability

PCA (Principal Component Analysis) is an unsupervised technique of feature engineering for dealing with high dimensional data and transforms data into a new set of variables which retaining most important variance in the data. PCA helps to enhance model performance, reducing overfitting and speeding up training by reducing the dimensionality.

$$PCA: X_{pca} = XW \quad (2)$$

In Eq. (1), X denotes Input data matrix and W indicates matrix of eigenvectors (Principal Component)

T-SNE (t-Stochastic Neighbor Embedded) used for clustering and visualization and reduces the data into

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

Two-dimensions but less suited for classification task and computationally expensive.

$$t - SNE: C = \sum_{i<j} \left[p_{ij} log \frac{p_{ij}}{q_{ij}} \right] \quad (3)$$

In Eq. (3), p_{ij} indicate probablity distribution in the high-dimensional space and q_{ij} denotes probability distribution in low-dimensional space.

Autoencoder compresses the data into lower dimensional latent space. It output is used as transformed feature set and it is a neural network model.

$$\text{Autoencoder: L} = \sum_{i=1}^{n} (X_i -) \overset{2}{\leftarrow} \quad (4)$$

In Eq. (5), X_i denotes original input, and $\hat{X}_i$ denotes reconstructed input, and N indicates number of features

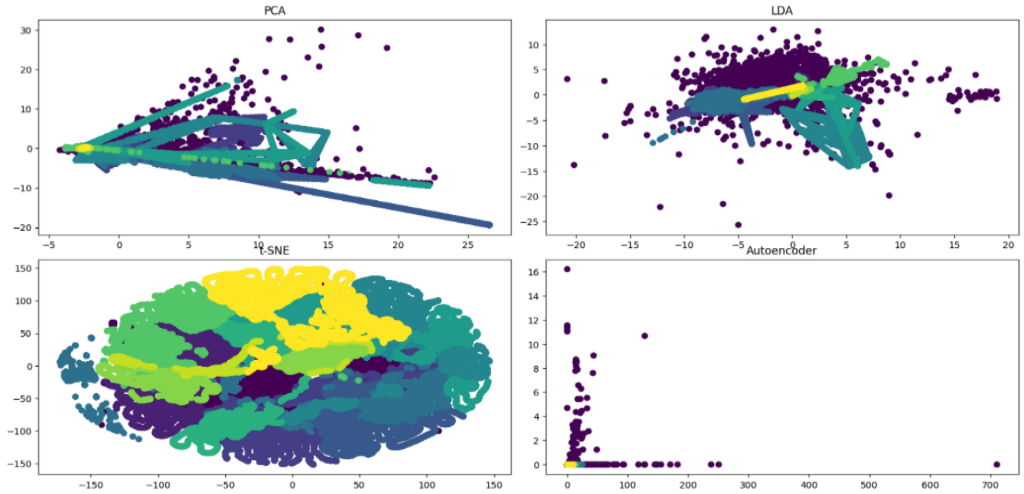

Figure 2. Impact of FE techniques on CICIDS2017 dataset

Figure 2 reflects the enhancing the understanding and visualization of high dimensional CICIDS2017 dataset by uncovering patterns and reducing dimensionality. LDA and PCA improve class separability, while t-SNE reveals nonlinear, complex relationships. Autoencoder extract related features and detect anomalies by making data easier to interpret and visualize.

D. ML Models

In our work, we use three ML models to evaluate performance metrics on each of these models. Firstly, Logistic Regression model implemented with code Logistic Regression(max_iter=1000,random_state=42), it reflects that it's a linear model which is used for both binary and multiclass classification, and its probability gives input to a certain class, max_iter=1000, this argument run to coverage for a sufficient number of iterations. This is chosen for its simple implementation process, better computational efficiency and give probability for each class that is useful for interpreting the results and preferred for classification tasks.

GradientBoostingClassifier(n_estimators=20, max_depth=3, random_state=42), it is an ensemble learning method that builds sequentially models, where each new model corrects the errors made by the previous ones and so on. It is useful for handling complex distribution of data n_estimators=20, that indicates the number of boosting trees (rounds) to build and max_depth=3, depth of each individual DT to improve generalization and avoid overfitting. It is flexible and powerful method, which works with a variety of data types that can achieve high accuracy on structured datasets like CICIDS2017, considered for our experiment. Finally, we use XGBoost (Extreme Gradient Boosting), a highly efficient implementation of gradient boosting includes optimizations such as parallelization, regularization and handling missing data. XGBClassifier(n_estimators=20, max_depth=3, learning_rate=0.1,use_label_encoder=False, eval_metric='mlogloss', random_state=42, n_jobs=-1), uses 20 number of boosting rounds, and 3 as the depth of every tree individually; learning rate controls the step size in each iteration and default label encoding behavior is disables

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

to avoid warning in newer versions. Also, evaluate metric as multi-class logarithmic loss used for classification tasks. It is used for performance to handling large datasets and its efficiency, missing valves and high dimensional data for structured classification tasks.

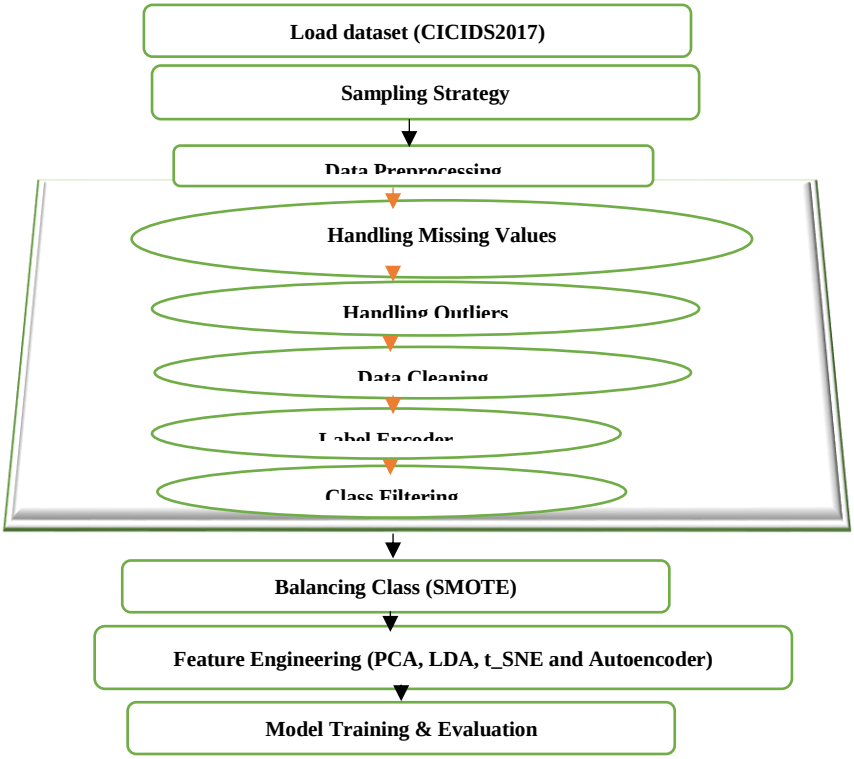

Figure 3. Our proposed work flow diagram

Figure 3. Reflects the whole process of our proposed work what and how we use techniques with gradually to understand the working easily by the future researcher.

Deep Learning Model

For learning hierarchical representations, we use feedforward neural networks to capture complex patterns and relationships in the data. A fully connected layer with 128 neurons for neural network defined using Keras. Our DL model consist of 3 dense layers and dropout (0.2) layers reflects regularization with 20% rate to prevent overfitting, also uses the ReLU (for hidden layers to add non-linearity) and softmax (for multi-class classification at output layer) activation function. After that, apply an early stopping mechanism to prevent from overfitting by halting training when the validation loss stops improving.

Adam optimizer is used to updates the weights of DL model during the training time and make the model efficient with speedup the model convergence as compared to conventional optimization methods which used for large dataset and sparse gradients like CICIDS2017 dataset, ideal for complex tasks like multi-class classification and not need the more hyper tunning for learning rate.

SMOTE (Synthetic Minority Over-sampling Technique) is used to generate new synthetic samples for underrepresented classes in the training set because it helps to prevent the model from being biased towards the majority class, which can improve performance of classification especially on imbalanced datasets. SMOTE is important for many real-world datasets which suffer from class imbalance because few classes have significantly fewer samples than others and also SMOTE helps to mitigate this by generating synthetic samples and providing the model with more balanced training data.

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

$$\boldsymbol{SMOTE}: \boldsymbol{X_{new}} = \boldsymbol{X_i} + \lambda\,(\boldsymbol{X_{ik}} - \boldsymbol{X_i}) \quad (5)$$

In Eq. (4), X_{new} denotes Synthetic sample and X_i indicates original sample and X_{ik} denotes neighboring sample and λ random value between 0 and 1

Figure 4. shows the class distribution after applying SMOTE technique to balance the attack classes of CICIDS2017 dataset which creates new synthetic samples of 18118 number of samples for each class to enhance the model performance.

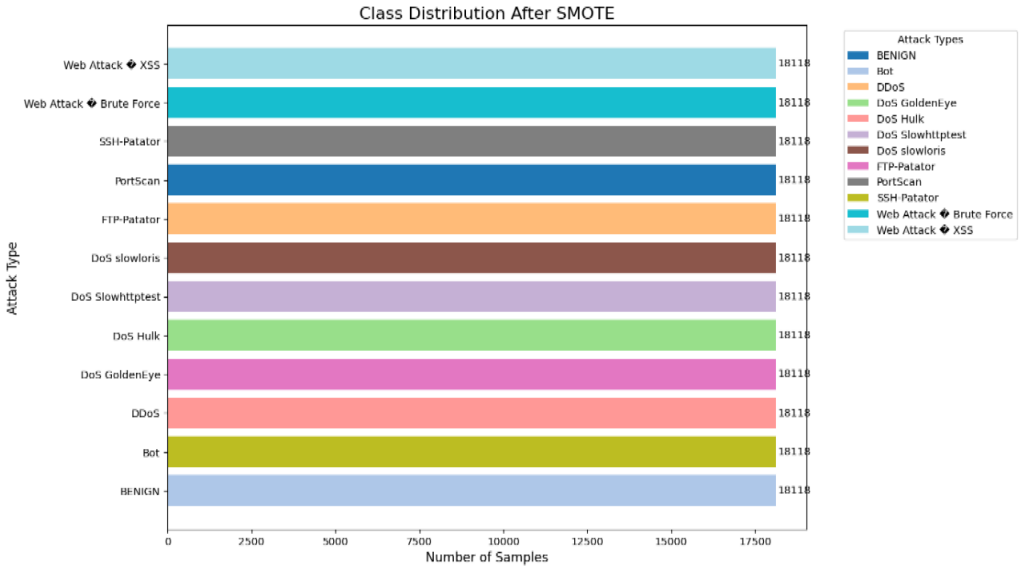

Figure 4. Class distribution after SMOTE on 5% CICIDS2017 dataset

Cross-Validation is used in our research for each model to train using Stratified K-Fold (K-3) cross-validation to ensure that each fold has the same proportion of classes, which is important to handle imbalanced datasets. Then evaluate the models in term of accuracy, training time and memory overhead. These metrics provide insights into model performance. In addition, comparative analysis of models for each feature engineering technique based on performance in the field if critical information infrastructure on IDS.

The obtained data is analyzed using qualitative synthesis and analysis to categorize, explain findings about feature engineering methodologies and their influence on IDS performance. Key subjects such as dimensionality reduction technique of FE, ML, DL and SMOTE are thoroughly evaluated to determine their relevance and usefulness in CII settings.

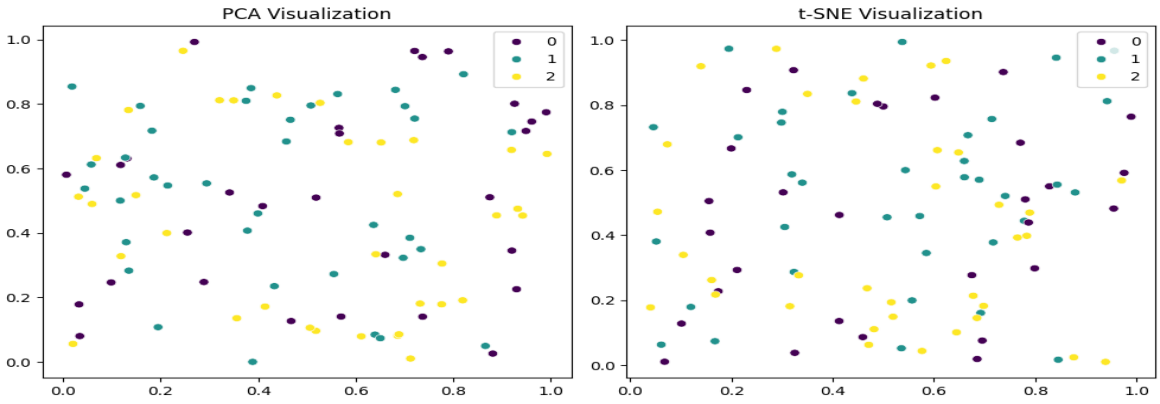

Figure 5. Data visualization with PCA and t_SNE

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

PCA emphasizes explaining the data variance by using linear transformations, so the separation between classes might not be perfect and t-SNE, being non-linear, can sometimes disclose richer clusters of similar data points, even if they are not linearly separable in the original high-dimensional space. In Figure 5, the PCA plot shows how the data looks when reduced to two dimensions in a way that tries to capture the most significant differences between points. The t-SNE plot often offers a better visual understanding of how data points cluster together, making it easier to see distinct groups or patterns, especially when classes are not linearly separable.

6. Result

Our experimental generated results are saved as a Data-Frame, which includes model performance metrics in terms of accuracy, training time and memory usage. This helps to compare models and their configurations.

Table 1 reflects the performing power of DL with LDA feature engineering approach by achieving highest test accuracy of 95.9%, and gradient boosting shows strong performance with attainment of a high-test accuracy of 90.8%.

But in term of training time, XGBoost took the least time which is 5.98 seconds when working with PCA and highlighting its efficiency to reduce the data, also deep learning with autoencoder FE approach perform with quick training time (67.71 seconds), this didn't suffer from long processing times.

In addition, in consideration of memory usage with autoencoder, LDA and PCA, all ML and DL models required similar memory usage that is 182MB approximately. Our experiment indicating the demands of consistent memory for advanced FE techniques.

Feature engineering techniques like PCA and LDA showed a strong balance between resource consumption and performance of model, particularly for gradient boosting and logistic regression, and t_SNE approach achieving lower CV (cross validation) accuracy because its representation capability did not transform well into high accuracy for most of the models. However, autoencoder technique performed very well in case of accuracy especially in DL. It required less training time compared to simpler models. Based on resource requirements, accuracy and training time providing valuable analysis for selecting the most suitable feature engineering technique and model.

In Figure 6 and 7 shows the class distribution with the attack name and also indicate the percentages of every attack in the whole and subset of CICIDS2017 dataset to know the dataset which have maximum and minimum attack samples in the dataset. We choose 5% of whole dataset to train and test our model and evaluate the performance of models and this distribution helps to balance technique apply or not for the dataset.

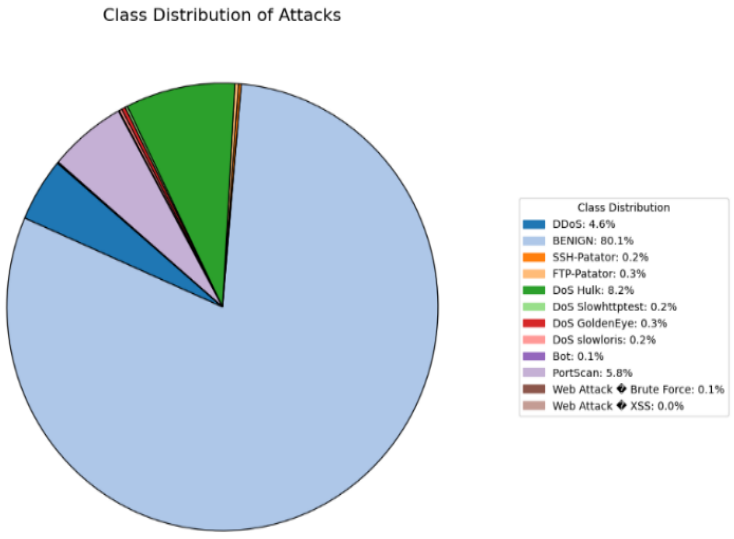

Figure 6. Class distribution of attacks for 5% CICIDS dataset

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

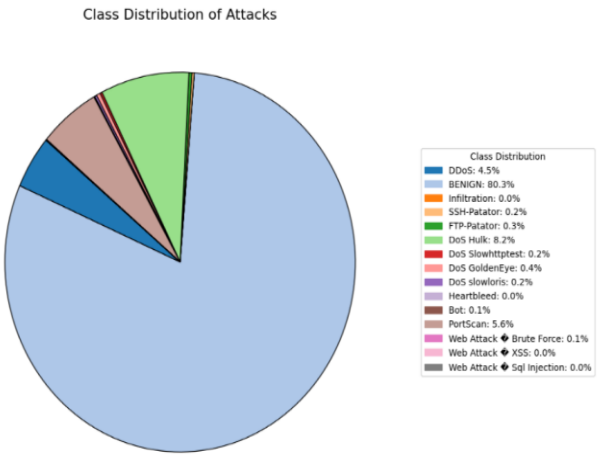

Figure 7. Class distribution of attack on full CICIDS2017 dataset

Table 1. shows the comparative analysis between performances of every model (LR, Gradient boosting, XGBossting and DL) with respect to feature engineering techniques (PCA, LDA, t_SNE and Autoencoder) in term of metrics like cross-validation accuracy, test accuracy, training time and memory usage for every model.

Table 1: Result of different FE techniques on ML and DL models

Models	**Feature Engineering Technique**	**CV Accuracy**	**Test Accuracy**	**Training Time (s)**	**Memory Usage**
Logistic Regression	PCA	0.907	0.817	211.70	115.87
	LDA	0.917	0.871	121.88	148.00
	T_SNE	0.354	0.187	116.62	182.64
	Autoencoder	0.933	0.791	255.79	182.64
Gradient Boosting	PCA	0.957	0.885	1099.11	148.00
	LDA	0.969	0.908	798.65	182.64
	T_SNE	0.781	0.281	184.15	182.64
	Autoencoder	0.961	0.875	881.25	182.64
XGBoosting	PCA	0.947	0.767	5.98	148.00
	LDA	0.963	0.862	5.14	182.64
	T_SNE	0.674	0.257	4.82	182.64
	Autoencoder	0.952	0.801	6.16	182.64
Deep Learning	PCA	0.463	0.915	69.56	182.64
	LDA	0.557	0.959	114.45	182.64
	t-SNE	0.166	0.208	111.36	182.64
	Autoencoder	0.176	0.916	67.71	182.64

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

Figure 8 reflects that LDA and PCA are best performing models with LDA, Gradient Boosting and feedforward NN achieving accuracy up to 90.8% and 95.9% respectively. On the other hand, autoencoder performed better in deep learning by achieving 91.6% accuracy, while t-SNE underperformed with very low accuracy in case of LR. Moreover, our experiment results exclusively stress on training time and memory usage trade-offs, providing a practical viewpoint.

Table 2 reflects the comparison of our proposed work result with the prominent relevant existing works in the domain; the observation concludes that the performance of proposed work improved in term of accuracy, training time and memory usage as compared to previous studies with FE techniques.

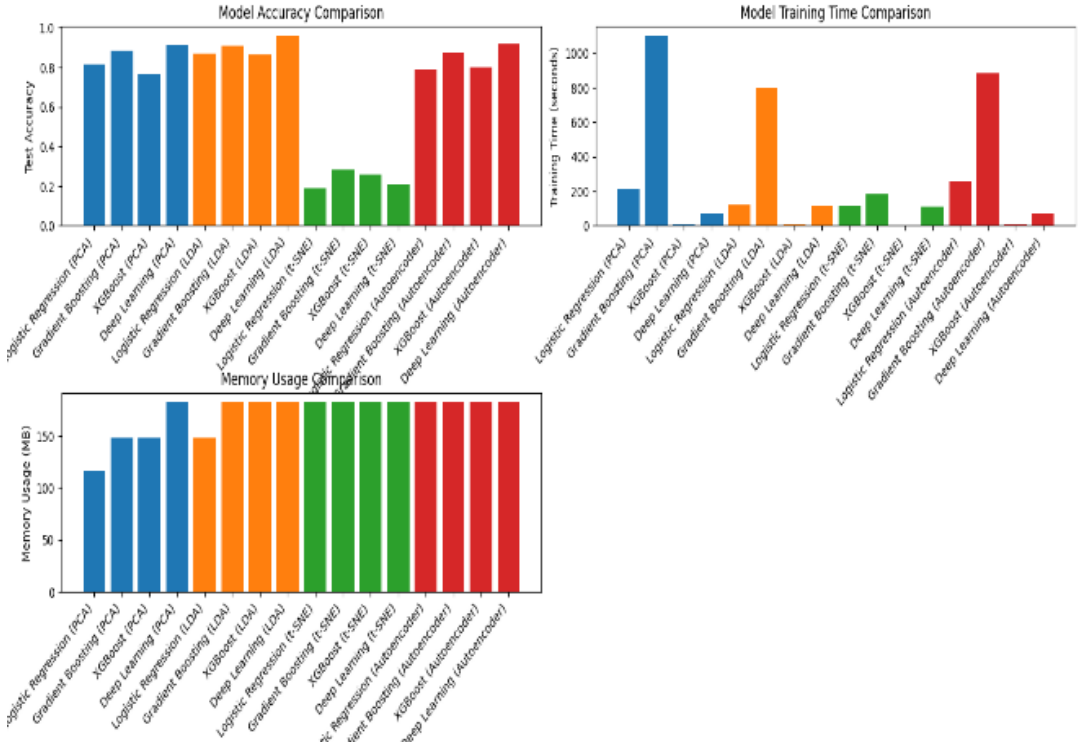

Figure 8. Comparative Analysis of different ML and DL approaches with FE techniques

Table 2: Comparative analysis with existing models

Research	FE Techniques	ML	DL	Accuracy	Time (s)	Memory
Hinton et. al (2006)	Autoencoder	No	Yes	Improved accuracy	High	Efficient
Sarhan et. al (2024)	PCA, Deep feature extraction	Yes	No	~90%	Efficient	NA
Srilatha et. al (2022)	Ensemble FS	Yes	No	~95%	Higher	Relatively High
Our Proposed Work	PCA, LDA, t_SNE and Autoencoder	Yes	Yes	LDA: 87% (LR), 90% (Gradient Boosting), 86% (XGBoost) and LDA: 95.9% (DL-FFNN)	5.14 – 1,099.11	115.87 – 182.64

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

7. Conclusion and Future Scope

In this research article, four dimensionality reduction methods LDA, PCA, t_SNE and autoencoder are evaluated for intrusion detection system for critical information infrastructures. These methods efficiently remove the irrelevant features for creating the classification model and these methods also takes necessary measures to handle the redundant features in the IDS benchmark dataset CICIDS2017. Feature Engineering remains a crucial step in the ML and DL models with significant implications for model performance and interpretability of IDS in the domain of CII.

In our proposed work, LDA and PCA consistently enhance the performance of conventional ML models, especially LDA. By capturing nonlinear feature interactions, autoencoders significantly improve DL models but t_SNE not be optimal for direct model training because it insightful for visualization.

Gradient Boosting and Logistic Regression ML model give consistent performance. However, XGBoost ML model outperformed on balanced datasets. By using all these FE techniques, we can improve our IDS in critical information infrastructures and make robust, adaptable, scalar and more secure critical infrastructures efficiently.

Despite the advancements, still challenges like high-dimensional data and the need for domain expertise persist, offering future research opportunities.

In future, researcher will use all these FE techniques on real-world datasets, streaming and high dimensional to extent the framework. In addition, apply hybrid approach to explore for example PCA + Autoencoder.

Evaluate framework efficiency with parallel processing and large datasets for scalability analysis, also explore advance class balancing technique like GANs (Generative Adversarial Networks) for minority class augmentation and SMOTE with ENN to overcome overfit.

Our proposed research gives an adaptable of complete framework for various sectors including finance, healthcare, government sectors, marketing etc., in critical information infrastructure where dimensionality reduction and class imbalance are critical. By combining ML and DL models for unified benchmarks, it provides valuable insights into choosing the FE technique and optimal model for classification tasks.

Funding Information: Not Applicable

Data Availability: This study includes only publicly available datasets referenced in the article.

Declarations:

Conflict of Interests: The authors declare that they have not any conflict of interests.

Research Involving Human and /or Animals: We did not involve any human or animals for this article.

Informed Consent: Not Applicable

References

[1] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[2] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowledge Data Eng.*, vol. 21, no. 9, pp. 1263-1284, 2009.

[3] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*, CRC Press, 2019.

[4] E. Alpaydin, *Introduction to Machine Learning*, 4th ed., MIT Press, 2020.

[5] D. Srilatha and N. Thillaiarasu, "DDoSNet: A deep learning model for detecting network attacks in cloud computing," in *Proc. 4th Int. Conf. Inventive Research Computing Applications*, ICIRCA 2022, 2022, doi: 10.1109/ICIRCA54612.2022.9985524.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157-1182, 2003.

[8] I. Guyon, J. Weston, and S. Barnhill, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 389-422, 2002.

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

[9] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825-2830, 2011.

[10] A. Kumar et al., "Feature engineering for IoT networks: A survey," *IEEE Access*, vol. 8, pp. 164107–164128, 2020.

[11] L. Kotthoff et al., "Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 826-830, 2017.

[12] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2006.

[13] S. Suhana, S. Karthic, and N. Yuvaraj, "Ensemble based dimensionality reduction for intrusion detection using random forest in wireless networks," in *Proc. 5th Int. Conf. Smart Systems Inventive Technology*, ICSSIT 2023, 2023, doi: 10.1109/ICSSIT55814.2023.10060929.

[14] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, and M. Portmann, "Feature extraction for machine learning-based intrusion detection in IoT networks," *Digital Communications and Networks*, vol. 10, no. 1, 2024, doi: 10.1016/j.dcan.2022.08.012.

[15] F. Zare and P. Mahmoudi-Nasr, "Feature engineering methods in intrusion detection system: A performance evaluation," *Int. J. Eng. Trans. B: Applications*, vol. 36, no. 7, 2023, doi: 10.5829/ije.2023.36.07a.15.

[16] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman, and R. M. Mohammad, "Intrusion detection system using feature extraction with machine learning algorithms in IoT," *J. Sensor Actuator Networks*, vol. 12, no. 2, 2023, doi: 10.3390/jsan12020029.

[17] R. Mohammad, F. Saeed, A. A. Almazroi, F. S. Alsubaei, and A. A. Almazroi, "Enhancing intrusion detection systems using a deep learning and data augmentation approach," *Systems*, vol. 12, no. 3, 2024, doi: 10.3390/systems12030079.

[18] S. Saha, A. T. Priyoti, A. Sharma, and A. Haque, "Towards an optimal feature selection method for AI-based DDoS detection system," in *Proc. IEEE Consumer Communications Networking Conf.*, CCNC 2022, 2022, doi: 10.1109/CCNC49033.2022.9700569.

[19] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Stat. Soc. B*, vol. 267, pp. 267-288, 1996.

[20] G. Ketepalli and P. Bulla, "Data preparation and pre-processing of intrusion detection datasets using machine learning," in *Proc. 6th Int. Conf. Inventive Computation Technologies*, ICICT 2023, 2023, doi: 10.1109/ICICT57646.2023.10134025.

[21] B. Natarajan, S. Bose, N. Maheswaran, G. Logeswari, and T. Anitha, "A new high-performance feature selection method for machine learning-based IoT intrusion detection," in *Proc. 12th IEEE Int. Conf. Advanced Computing*, ICoAC 2023, 2023, doi: 10.1109/ICoAC59537.2023.10249916.

[22] K. Albulayhi et al., "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Appl. Sci. (Switzerland)*, vol. 12, no. 10, 2022, doi: 10.3390/app12105015.

[23] Y. Guo et al., "Deep feature selection: Theory and application to identify enhancers and promoters," *Bioinformatics*, vol. 35, no. 21, pp. 4298-4306, 2019.

[24] J. M. Kanter and K. Veeramachaneni, "Deep feature synthesis: Towards automating data science endeavors," in *Proc. 2015 IEEE Int. Conf. Data Science Advanced Analytics*, 2015, pp. 1-10.

[25] A. Ng, "Sparse autoencoder," CS294A Lecture Notes, 2011.

[26] Y. Niu et al., "Application of a new feature generation algorithm in intrusion detection system," *Wireless Commun. Mobile Comput.*, 2022, doi: 10.1155/2022/3794579.

[27] E. Roponena and I. Polaka, "Classifier selection for an ensemble of network traffic analysis machine learning models," in *Proc. 2022 63rd Int. Sci. Conf. Information Technology Management Science Riga Technical University*, ITMS 2022, 2022, doi: 10.1109/ITMS56974.2022.9937116.

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025

[28] S. Das et al., "Network intrusion detection and comparative analysis using ensemble machine learning and feature selection," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, 2022, doi: 10.1109/TNSM.2021.3138457.

[29] A. Fatani et al., "Advanced feature extraction and selection approach using deep learning and aquila optimizer for IoT intrusion detection system," *Sensors*, vol. 22, no. 1, 2022, doi: 10.3390/s22010140.

[30] H. Lin, Q. Xue, J. Feng, and D. Bai, "Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine," *Digital Commun. Networks*, vol. 9, no. 1, 2023, doi: 10.1016/j.dcan.2022.09.021.

[31] A. M. Abdullah et al., "Feature engineering algorithms for traffic dataset," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 4, 2021, doi: 10.14569/IJACSA.2021.0120435.

[32] R. R. Rejimol Robinson, K. P. Anagha Madhav, and C. Thomas, "Improved minority attack detection in intrusion detection system using efficient feature selection algorithms," *Expert Systems*, vol. 41, no. 7, 2024, doi: 10.1111/exsy.13546.

[33] P. C. Nguyen, Q. T. Nguyen, and K. H. Le, "An ensemble feature selection algorithm for machine learning-based intrusion detection system," in *Proc. 2021 8th NAFOSTED Conf. Information Computer Science*, NICS 2021, 2021, doi: 10.1109/NICS54270.2021.9701577.

[34] R. Mohammad et al., "Enhancing intrusion detection systems using a deep learning and data augmentation approach," *Systems*, vol. 12, no. 3, 2024.

DOI: https://doi.org/10.54216/JCIM.160105
Received: November 14, 2024 Revised: January 16, 2025 Accepted: February 14, 2025