The Study of Feature Engineering in Machine Learning and Deep Learning for Network Intrusion Detection Systems

Steven Ning
Silicon Valley Cybersecurity Institute
Saratoga High School, CA, USA

Khanh Nguyen
Software Engineering
San Jose State University, CA, USA

Sohini Bagchi, Younghee Park*
Computer Engineering
San Jose State University, CA, USA

2024 Silicon Valley Cybersecurity Conference (SVCC) | 979-8-3503-8314-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/SVCC61185.2024.10637359

***Abstract*—With the rise of sophisticated cyberattacks, the efficiency of intrusion detection systems becomes paramount. Machine learning (ML) and deep learning (DL) models used for intrusion detection often encounter datasets with irrelevant or redundant features, leading to low performance. To address this challenge, feature engineering techniques are important in extracting the most informative features, enabling faster and more accurate detection of malicious patterns. This paper investigates a comparative analysis of four feature engineering methods using a historical archival dataset: entropy, mutual information, chi-squared statistics, and ANOVA. By evaluating and comparing the effectiveness of these methods under different conditions of ML/DL models, this study aims to provide insights into their respective strengths and weaknesses, guiding the selection of the most suitable approach to improve the efficiency of network intrusion detection systems.**

***Index Terms*—Network Intrusion Detection System, NSL-KDD Datasets, Feature Engineering, Machine Learning Models, Deep Learning Models**

I. Introduction

The application of machine learning (ML) and deep learning (DL) techniques in the development of Network Intrusion Detection Systems (NIDS) has been increasingly recognized for its potential to accurately identify and respond to cyber threats. These techniques offer advanced capabilities in pattern recognition and anomaly detection, leveraging vast amounts of network data to identify malicious activities. As cyber-attacks become more sophisticated, leveraging the strengths of ML and DL in NIDS is crucial for enhancing cybersecurity measures across digital infrastructures [1].

A wealth of research has been dedicated to exploring the effectiveness of various ML and DL models in the realm of NIDS. Studies have compared the performance of traditional machine learning algorithms, such as Random Forest and Support Vector Machines, against more complex deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), across multiple datasets. The comprehensive analysis has been instrumental in understanding the capabilities and limitations of different approaches in detecting network intrusions, guiding the cyber-security community toward more robust and adaptive detection systems [2].

This paper aims to contribute to the evolving field of ML/DL-based NIDS by exploring the impact of feature engineering on the performance of selected machine learning and deep learning models. Recognizing the critical role of feature selection and optimization in enhancing model accuracy and efficiency, our study provides a comprehensive analysis of different feature engineering methods, including entropy, mutual information, chi-square statistics, and ANOVA. By evaluating these methods across both the ML and the DL models, this paper seeks to provide valuable insight into optimizing NIDS for better threat detection and system performance. Our contribution lies not only in the comparative analysis of feature engineering methods, but also in demonstrating the practical implications of these methods on the efficacy of NIDS in real-world scenarios.

The paper is organized into sections that cover related work, methodology, experimental results, and conclusions. The related work section discusses previous research, while the methodology section outlines the intrusion detection system's development. The experimental results present the performance of the model, and the conclusion suggests future research directions.

II. Related Works

Many research communities have utilized machine learning and deep learning techniques to develop network intrusion detection systems, as they can detect malicious activities by profiling attack behavior with large datasets.

Dhanabal and Shantharajah conducted a study on the NSL-KDD dataset using classification algorithms to evaluate their effectiveness for intrusion detection, as understanding the dataset's characteristics is crucial for developing robust intrusion detection systems [3]–[5]. Verma and Ranga employed k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) on the CIDDS-001 dataset for Network Intrusion Detection Systems (NIDS). Their methodology involved taking advantage of the distance-based nature of k-NN and the decision boundary optimization of SVM to enhance the system's ability to identify and mitigate network threats [6]. Liu and Lang's survey comprehensively analyzed machine learning methodologies applied in NIDS, including Decision Trees, Naive Bayes, and SVMs. They highlighted the adaptability of these techniques to various intrusion detection

Authorized licensed use limited to: Steven Ning. Downloaded on September 10,2026 at 08:08:49 UTC from IEEE Xplore. Restrictions apply.

scenarios, emphasizing their individual strengths in handling various types of network threats [7]. Alzahrani and Alenazi designed a machine learning-based NIDS specifically tailored for Software Defined Networks (SDNs). Their methodology involved incorporating Decision Trees and Ensemble methods to adapt to the dynamic nature of SDNs, demonstrating the effectiveness of machine learning in securing evolving network architectures [8]. Lee et al. developed a machine learning-based NIDS utilizing classification algorithms such as Random Forest and Support Vector Machines. They focused on feature selection and classifier training to create a robust intrusion detection system capable of profiling and identifying malicious network activities [9]. Sommer and Paxson explored the use of machine learning beyond closed systems for network intrusion detection, employing unsupervised learning techniques such as clustering. Sommer and Paxson involved anomaly detection and behavior profiling, showcasing the applicability of machine learning in real-world intrusion scenarios [10]. Farnaaz and Jabbar implemented a Random Forest model for NIDS, utilizing ensemble learning to enhance intrusion detection capabilities. They involved constructing an ensemble of decision trees, emphasizing the diversity and collective decision-making power of the Random Forest model [11].

Despite various effective ML-based NIDS, deep learning-based NIDS have been popularly proposed to improve detection accuracy while decreasing false alarms. Ashiku and Dagli introduced a deep learning-based NIDS employing Convolutional Neural Networks (CNNs). They would be training CNNs on network traffic data to automatically learn hierarchical features, showcasing the potential of deep learning in capturing complex patterns and anomalies [12]. Shone et al. proposed a deep learning approach for NIDS, utilizing Long Short-Term Memory (LSTM) networks. This involved sequence modeling to capture temporal dependencies in network traffic, highlighting the effectiveness of deep learning in discerning intricate intrusion patterns over time [13]. Thapa et al. conducted a comparative analysis of machine learning and deep learning models for NIDS, focusing on Deep Neural Networks (DNNs) and their ability to automatically extract hierarchical features. They evaluated the performance of DNNs compared to traditional machine learning methods, emphasizing the inherent feature learning capabilities of deep architectures [14]. Vinayakumar et al. implemented a deep learning approach for intelligent intrusion detection systems, employing a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). This included the use of CNNs for spatial feature extraction and RNNs for capturing temporal dependencies, demonstrating the synergy of different deep learning architectures [15]. Gamage and Samarabandu conducted a survey and comparison of various deep learning methods in network intrusion detection, emphasizing the strengths of deep architectures such as Autoencoders and Generative Adversarial Networks. They were evaluating these models in terms of accuracy, robustness, and their ability to handle diverse intrusion scenarios [16]. Qazi et al. presented an intelligent and efficient NIDS utilizing a Hybrid Deep Learning Model combining CNNs and Long Short-Term Memory (LSTM) networks. They showcased the evolving landscape of intrusion detection with advanced neural network architectures, highlighting the importance of combining spatial and temporal learning for improved accuracy and efficiency [17].

III. Our Approach

The development of the network intrusion detection system includes multiple steps in network data analysis: data collection, data cleaning and feature engineering, followed by parameter tuning on a specific ML/DL algorithm, significantly influencing the performance of network intrusion detection systems. This paper investigates the impact of feature engineering based on different methods: entropy, mutual information, chi-square statistics, and ANOVA. The different feature engineering methods are evaluated with the most popular machine-learning and deep-learning algorithms: Random Forest, Support Vector Machine(SVM), Recurrent Neural Network (RNN), and Deep Learning with Multi-Layer Perceptron (MLP). With the four algorithms, the various feature engineering methods automate the extraction of the top ten and twenty features based on statistical tests.

With the two populuar ML and DL algorithms, this paper investigates the impact of different automatic feature engineering methods: entropy, mutual information, chi-square statistics, and ANOVA. Here is the summary of each feature engineering method.

Mutual Information: Mutual information measures the amount of information shared between two variables [12]. Mutual information was used to automate the extraction of the top features. This statistical test identified the features that offered the most information about the target variable, enhancing the discriminatory power of the selected features for intrusion detection in the NSL-KDD dataset.

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \left(\frac{p(x,y)}{p_X(x) p_Y(y)} \right)$$

Entropy: Entropy is a fundamental concept in information theory that measures uncertainty and disorder within a system [11]. In this study, entropy was computed for each of the 41 features in the NSL-KDD dataset. The features were then sorted according to their entropy difference, where the top features with the highest entropy differences were selected. This method helps identify features that significantly contribute to distinguishing between different classes of network traffic.

Chi-Square: Chi-square statistics assess the independence between categorical variables by comparing observed and expected frequencies of different categories within a contingency table. Chi-square statistics were integrated with the SelectKBest method to identify the best categorical features most relevant for intrusion detection in the NSL-KDD dataset. This

Authorized licensed use limited to: Steven Ning. Downloaded on September 10,2026 at 08:08:49 UTC from IEEE Xplore. Restrictions apply.

statistical test ensured the selection of features significantly associated with the target variable.

$\chi_c^2 = \sum \left(\frac{(O_i - E_i)^2}{E_i} \right)$

ANOVA: ANOVA analyzes the variation between group means and within-group variability. It assesses whether there are statistically significant differences between the means of multiple groups. ANOVA was applied to identify the greatest continuous features with significant differences in mean between different classes. This method is particularly useful for continuous features, helping select those that contribute significantly to class distinctions in the NSL-KDD dataset.

We use four different models to evaluate the various feature engineering methods: Random Forest (RF), Support Vector Machine(SVM), Recurrent Neural Network (RNN), and Deep Learning with Multi-Layer Perceptron (MLP). First, the RF model is an ensemble learning algorithm that combines multiple decision trees to make predictions. Second, SVM is a supervised learning algorithm that aims to find a decision boundary that maximizes the margin between different classes. The algorithm identifies a subset of training samples, known as support vectors, that are closest to the decision boundary. In our implementation, utilizing the Radial Basis Function (RBF) kernel implicitly facilitates this process, enabling the SVM to discern intricate patterns within the data space. Third, RNN designed for sequential data, was investigated due to its capacity to capture temporal dependencies in network traffic. The model architecture was configured with multiple hidden layers to effectively capture intricate temporal patterns in network data. Lastly, deep learning with MLP, a feedforward neural network, is applied to capture intricate patterns in network data. It consists of multiple hidden layers, allowing the model to learn hierarchical representations of features. With different hyperparameter settings in each model, this paper investigates the impact of each feature engineering method.

IV. Evaluation

This section presents the experimental results with different insights regarding the impact of feature engineering, depending on different models and the number of feature sets.

A. Dataset

The NSL-KDD dataset [5] was used for this study. Using the well-known NSL-KDD dataset containing 197,222 traces and 41 features, experiments were performed with 10 selected features and 20 features, and the data were split following an 80:20 ratio for training and testing. The evaluation goal was to observe the impact of feature engineering, compare results between models, and assess performance with varying hyperparameter values.

B. Performance Evaluation

The feature engineering process in our study focused on optimizing the model performance for intrusion detection using the NSL-KDD dataset through subsets of 10 and 20 features. Four distinct methods—entropy, mutual information, chi-square statistics, and ANOVA—were employed to determine the most informative features. First, in the entropy-based feature engineering method, the top 10 and 20 features with the greatest differences in entropy were selected. Second, mutual information with statistical tests automatically identified the top 10 and 20 feature sets that offered the most information about the target variable, enhancing the discriminatory power of the selected features for intrusion detection in the NSL-KDD dataset. Third, chi-square statistics were integrated with the SelectKBest method to identify the top 10 and 20 categorical features most relevant for intrusion detection in the NSL-KDD dataset. This statistical test ensured the selection of features significantly associated with the target variable. Finally, ANOVA was also applied to identify the top 10 and 20 continuous features with significant differences in means between different classes. This method is particularly useful for continuous features, helping to select those that contribute significantly to class distinctions in the NSL-KDD dataset.

Impact of Feature Engineering: Tables 1-4, Figure 1, and Figure 2 demonstrated that the feature engineering methods showed different impacts on the performance of machine learning and deep learning models. The selection of feature engineering methods in the ML model is more important than the one in the DL model. In other words, the ML models with different feature engineering methods demonstrated different performance results, however, the feature engineering methods showed similar performance regardless of different deep learning models. The difference in performance results was very minimal when we used 20 feature sets for the DL models. For example, depending on different feature engineering methods, SVM showed 94%- 98.77% accuracy with 10 feature sets and 90%- 96.61% accuracy with 20 feature sets. However, RNN presented a similar accuracy (93.41% - 95.59% with 10 feature sets and 97.12% - 97.82% with 20 feature sets) regardless of different feature engineering methods. Furthermore, the feature engineering methods did not impact the performance (less than 1% differences) in the DL models with 20 feature sets. This suggests that DL models inherently possess a higher capacity to automatically extract relevant features from raw data, diminishing the significance of explicit feature engineering compared to traditional ML approaches. Therefore, when using machine learning methods, we select the right feature engineering method with desirable feature sets.

Number of feature sets: Figure 1 and Figure 2 demonstrate that the number of feature sets is the most important factor to improve the performance of ML/DL-based NIDS. In particular, when 20 feature sets were used through different feature engineering methods, the performance showed almost 98% for RNN and 99% for MLP regardless of the feature engineering methods. Compared to the performance with 10 feature sets, the performance with 20 feature sets increased 3-4% accuracy. However, the SVM method was sensitively dependent on the number of feature sets and each feature engineering method. Therefore, we consider the desirable number of feature sets in machine learning models, however, we can include more

Authorized licensed use limited to: Steven Ning. Downloaded on September 10,2026 at 08:08:49 UTC from IEEE Xplore. Restrictions apply.

valuable feature sets in DL models based on the outcomes of feature engineering methods.

Hyperparameter Tuning: The feature engineering methods and the ML/DL models included many parameters that we needed to set up for these experiments. In particular, depending on each model, it is important to select the right values for each hyperparameter. The RF model can specify the maximum depth of each decision tree and SVM has γ to influence the shape of the decision boundary for finding the right balance between underfitting and overfitting. We set up the maximum depth with 10 and $\gamma = 0.1$ with the cost (C = 100) in our experiments. In addition, RNN and MLP have more than ten hyperparameters including the number of hidden neurons, the number of layers, activation functions, dropout, and learning rates. In these experiments, for the RNN, we used a configuration of 50 LSTM units with 'relu' activation and a 'sigmoid' output activation, trained using the 'adam' optimizer on a binary_crossentropy loss. The MLP was configured with one hidden layer of 100 neurons, using 'relu' as the activation function, and was trained for a maximum of 500 iterations.

Tuning hyperparameters plays a crucial role in the performance of an ML/DL model. The optimal values for these hyperparameters depend on the characteristics of the datasets. To improve the performance, hyperparameter tuning is an essential task to find the best combination of hyperparameter values for a given dataset.

TABLE I: Results from 10 features and 20 features using Entropy

	Model	10 feat.				20 feat.			
		Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
ML	RF	99.7	99.87	99.53	99.72	99.66	99.87	99.41	99.66
	SVM	98.71	99.74	97.5	98.7	90.43	1	79.6	90.2
DL	MLP	98.27	99.74	97.5	98.7	99.13	98.37	99.45	99.12
	RNN	95.59	96.68	96.02	96.35	97.82	97.63	97.71	97.67

TABLE II: Results from 10 features and 20 features using Mutual Information

	Model	10 feat.				20 feat.			
		Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
ML	RF	99.58	99.7	99.41	99.58	99.6	99.87	99.28	99.6
	SVM	98.77	99.74	97.63	98.76	90.43	1	79.6	90.19
DL	MLP	98.31	99.74	97.63	98.31	99.11	98.86	99.11	99.1
	RNN	93.41	90.28	96.32	93.2	97.64	98.7	96.23	97.45

TABLE III: Results from 10 features and 20 features using Chi Squared

	Model	10 feat.				20 feat.			
		Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
ML	RF	95.51	97.24	95.26	96.49	98.91	99.57	98.1	98.9
	SVM	94.44	93.96	94.2	94.42	96.61	97.53	95.18	96.59
DL	MLP	95.3	94	94.2	95.3	98.41	97.53	95.18	98.41
	RNN	94.48	96.11	91.96	93.99	97.28	98.1	96.06	97.1

TABLE IV: Results from 10 features and 20 features using ANOVA

	Model	10 feat.				20 feat.			
		Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
ML	RF	97.64	97.41	95.6	96.73	98.91	99.61	98.05	98.9
	SVM	94.52	94.31	94	94.5	96.63	97.85	95.09	96.61
DL	MLP	95.47	94.31	94	95.45	98.81	97.65	95.1	98.81
	RNN	94.76	95.53	93.19	94.34	97.12	97.97	95.85	96.9

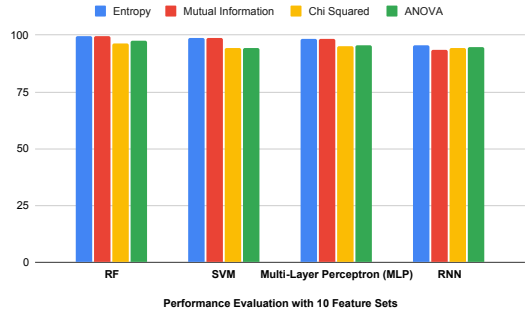


Fig. 1: Performance Evaluation with 10 features

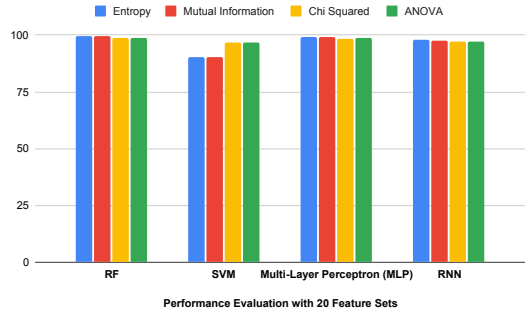


Fig. 2: Performance Evaluation with 20 features

V. Conclusion

This study's in-depth analysis and evaluations of two prevalent machine learning algorithms, Random Forest and Support Vector Machine, utilizing the NSL-KDD dataset, have underscored the pivotal role of hyperparameter tuning in optimizing model performance across various metrics. The findings reiterate that selecting appropriate hyperparameters is crucial for the effective training of machine learning models, directly influencing their accuracy, precision, recall, and F1 scores. Furthermore, our exploration extends beyond these traditional machine learning approaches to include an examination of deep learning techniques, specifically Recurrent Neural Networks (RNNs) and Multi-Layer Perceptrons (MLPs). These deep learning models demonstrate a significant capacity for handling complex data structures and dependencies, offering a complementary perspective on network intrusion detection.

In conclusion, the collective insights from this study illuminate the multifaceted approach required to enhance the

Authorized licensed use limited to: Steven Ning. Downloaded on September 10,2026 at 08:08:49 UTC from IEEE Xplore. Restrictions apply.

performance of network intrusion detection systems. The careful selection of hyperparameters, alongside judicious feature engineering and the strategic application of both machine and deep learning models, forms the cornerstone of developing robust, efficient, and accurate intrusion detection systems. As cyber threats evolve in complexity and sophistication, the findings from this research contribute to the ongoing efforts to fortify digital infrastructures against malicious activities. Future work in this area may explore the integration of ensemble learning methods, the application of advanced deep learning architectures, and the investigation of real-time detection capabilities, aiming to continuously adapt and respond to the dynamic landscape of cybersecurity threats.

ACKNOWLEDGMENT

This research was supported in supported by the National Science Foundation under Grant Number 2244597. Dr. Park is a corresponding author.

REFERENCES

[1] Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerging Tel Tech.2021;32:e4150.https://doi.org/10.1002/ett.4150

[2] Liu H, Lang B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences. 2019; 9(20):4396. https://doi.org/10.3390/app9204396

[3] Dhanabal, L. and S. P. Shantharajah. “A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms.” (2015).

[4] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, Jaideep Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection”

[5] University of New Brunswick, Canada. NSL-KDD dataset.

[6] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorban “A Detailed Analysis of the KDD CUP 99 Data Set”, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)

[7] S. Revathi, Dr. A. Malathi, “A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection”, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 2 Issue 12, December - 2013

[8] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, ”A Deep Learning Approach to Network Intrusion Detection,” in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[9] B. Dong and X. Wang, “Comparison deep learning method to traditional methods using for network intrusion detection,” in Proc. 8th IEEE Int. Conf. Commun. Softw. Netw., Beijing, China, Jun. 2016, pp. 581

[10] Muhuri, Pramita Sree, Prosenjit Chatterjee, Xiaohong Yuan, Kaushik Roy, and Albert Esterline. 2020. ”Using a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to Classify Network Attacks” Information 11, no. 5: 243. https://doi.org/10.3390/info11050243

[11] K. Yoshida, ”Entropy based intrusion detection,” 2003 IEEE Pacific Rim Conference on Communications Computers and Signal Processing (PACRIM 2003) (Cat. No.03CH37490), Victoria, BC, Canada, 2003, pp. 840-843 vol.2, doi: 10.1109/PACRIM.2003.1235912.

[12] S. Mohammadi, V. Desai and H. Karimipour, ”Multivariate Mutual Information-based Feature Selection for Cyber Intrusion Detection,” 2018 IEEE Electrical Power and Energy Conference (EPEC), Toronto, ON, Canada, 2018, pp. 1-6, doi: 10.1109/EPEC.2018.8598326.

[13] Z. Rustam, N. P. A. A. Ariantari; Comparison between support vector machine and fuzzy Kernel C-Means as classifiers for intrusion detection system using chi-square feature selection. AIP Conf. Proc. 22 October 2018; 2023 (1): 020214. https://doi.org/10.1063/1.5064211

[14] H. Tribak, B. L. Delgado-Márquez, P. Rojas, O. Valenzuela, H. Pomares and I. Rojas, ”Statistical analysis of different artificial intelligent techniques applied to Intrusion Detection System,” 2012 International Conference on Multimedia Computing and Systems, Tangiers, Morocco, 2012, pp. 434-440, doi: 10.1109/ICMCS.2012.6320275.

[15] S. Moghanian, F. B. Saravi, G. Javidi and E. O. Sheybani, ”GOAMLP: Network Intrusion Detection With Multilayer Perceptron and Grasshopper Optimization Algorithm,” in IEEE Access, vol. 8, pp. 215202-215213, 2020, doi: 10.1109/ACCESS.2020.3040740.

[16] Sunanda Gamage, Jagath Samarabandu, Deep learning methods in network intrusion detection: A survey and an objective comparison, Journal of Network and Computer Applications, Volume 169, 2020, 102767, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2020.102767.

[17] Emad-ul-Haq Qazi, Muhammad Imran, Noman Haider, Muhammad Shoaib, Imran Razzak, An intelligent and efficient network intrusion detection system using deep learning, Computers and Electrical Engineering, Volume 99, 2022, 107764, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2022.107764.

Authorized licensed use limited to: Steven Ning. Downloaded on September 10,2026 at 08:08:49 UTC from IEEE Xplore. Restrictions apply.