Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 63−69
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.104
https://ejurnal.pdsi.or.id/index.php/bids/index

Optimizing Autoencoder-Based Feature Selection for Attack Detection in IoT Networks via Machine Learning Approaches

Eko Arip Winanto[1,*], Kurniabudi[1], Sharipuddin[2],

[1] Faculty of Computer Science , Computer Enginerring, Dinamika Bangsa University, Jambi, Indonesia
[2] Faculty of Computer Science, Informatics, Dinamika Bangsa University, Jambi, Indonesia
Email: [1,*]ekoaripwinanto@gmail.com, [2] kbudiz@yahoo.com, [3]sharifbuhaira@gmail.com
Correspondence Author Email: ekoaripwinanto@gmail.com

Abstract

The Internet of Things (IoT) presents significant security challenges as the number of connected devices continues to grow. One critical approach in developing efficient attack detection systems is the selection of relevant features to reduce model complexity without compromising accuracy. This study evaluates the effectiveness of Autoencoders as a feature reduction method for IoT network intrusion detection systems. Three machine learning algorithms are employed for comparative analysis: K-Nearest Neighbors (KNN), Naive Bayes (NB), and Support Vector Machine (SVM). The dataset is evaluated both before and after feature reduction using an Autoencoder, with performance assessed based on accuracy, precision, recall, F1-score, training time, and the number of features. Experimental results demonstrate that the Autoencoder can reduce the number of features by up to 30% without significantly degrading performance. In fact, the NB and SVM models exhibit improvements in both accuracy and training efficiency. The KNN model shows a minimal performance decline, which remains within acceptable limits. Overall, the Autoencoder proves to be an effective method for feature reduction, maintaining or even enhancing detection efficiency and performance. These findings support the use of Autoencoders as an efficient feature selection technique in IoT-based attack detection systems.

Keywords: Autoencoder; Feature Selection; IoT; KNN; Naive Bayes; SVM; Attack Detection

1. INTRODUCTION

The Internet of Things (IoT) has experienced rapid development in recent years and is now utilized across various critical sectors, including industry, healthcare, and transportation. However, the increasing number of interconnected IoT devices has also expanded the attack surface, exposing systems to greater cybersecurity risks [1], [2]. The vast amount of heterogeneous data generated by IoT devices poses a significant challenge to intrusion detection systems (IDS), which must operate efficiently and accurately under these conditions [3], [4].

Intrusion detection in IoT networks is commonly performed using machine learning algorithms, which require clean and relevant feature data to achieve optimal classification performance [5], [6]. However, not all available features contribute meaningfully to the classification process. Irrelevant or redundant features can degrade model accuracy and increase computational costs [7].

Various feature selection and dimensionality reduction techniques have been explored in recent literature to address this issue. Traditional filter-based methods, such as Information Gain (IG), Chi-Square, and Gain Ratio, assess the statistical relevance of each feature independently, offering simplicity and low computational cost. However, these methods often fail to capture feature interactions, which can be crucial in complex domains like IoT intrusion detection [8][9]. On the other hand, wrapper-based techniques, including Recursive Feature Elimination (RFE), use machine learning models to iteratively select optimal subsets of features based on classification performance. While more accurate, these approaches are computationally intensive and less scalable [10].

Principal Component Analysis (PCA), a popular unsupervised dimensionality reduction technique, projects high-dimensional data into a lower-dimensional space by identifying directions (principal components) of maximum variance. Although PCA is effective for noise reduction and compact feature representation, its linear nature may fail to capture nonlinear relationships present in real-world IoT traffic. Moreover, PCA-transformed features are typically abstract and may reduce interpretability or discard critical class-specific information in highly imbalanced datasets [11].

To overcome these limitations, recent studies have shifted towards deep learning-based approaches, particularly Autoencoders, which offer more flexible and powerful nonlinear transformations. Autoencoders are neural network architectures designed to encode input data into a compressed representation and reconstruct it back with minimal loss, thereby retaining essential information. Their ability to learn complex feature interactions in an unsupervised manner makes them especially suitable for high-dimensional and heterogeneous IoT datasets. Moreover, variations such as sparse, denoising, and variational Autoencoders can be adapted for specific challenges, including noise reduction, anomaly detection, and representation learning [12].

Autoencoders, a form of unsupervised learning, have been widely applied for dimensionality reduction and feature extraction in the context of big data [13]. With their deep neural network architecture, Autoencoders are capable of learning compressed representations of input data while preserving essential informatio [14]. For instance, study [15], [16] employed Autoencoders on the NSL-KDD dataset and reported improved detection performance in several classification scenarios. Similarly, [17], [18] integrated Deep Autoencoders with Random Forests for anomaly detection in IDS data, successfully reducing noise and enhancing accuracy. In a study [19] stacked Autoencoder approach was utilized to reduce feature dimensionality in large-scale IoT traffic datasets, resulting in increased classification speed without significant accuracy loss. Additionally, [20] explored the use of denoising Autoencoders to address noisy and

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 63−69
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.104
https://ejurnal.pdsi.or.id/index.php/bids/index

incomplete IoT sensor data, demonstrating resilience and robustness in real-world smart home environments. Another relevant work by Li and He [21] implemented a hybrid system combining Autoencoders with Support Vector Data Description (SVDD), achieving high detection rates on imbalanced IoT attack data. Meanwhile, recent research [22] evaluated sparse Autoencoders to extract compact features from time-series network traffic, significantly reducing storage and training time. Furthermore, in [23], the authors proposed a Convolutional Autoencoder architecture tailored to extract spatial-temporal features from IoT-based industrial control system (ICS) networks, showing strong detection capabilities against stealthy intrusions.

Nevertheless, most existing studies have not explicitly evaluated the effectiveness of Autoencoders in IoT contexts, where data tends to be more complex, diverse, and often imbalanced across classes . Therefore, further investigation is needed to assess the impact of Autoencoder-based feature selection on detection accuracy and overall model efficiency in IoT attack detection systems.

This study contributes by evaluating the effectiveness of Autoencoders as a feature selection and extraction technique in machine learning-based intrusion detection systems for IoT networks. Specifically, the performance of three classification algorithms KNN, NB, and SVM is compared before and after the application of Autoencoder-based feature reduction. In addition to accuracy and standard evaluation metrics precision, recall, and F1-score, this study also considers efficiency aspects such as the reduction in the number of features and model computation time.

The remainder of this paper is structured as follows: Section I presents the introduction; Section II describes the methodology; Section III discusses the results and analysis; and Section IV concludes the study and outlines directions for future research.

2. RESEARCH METHODOLOGY

2.1 Experiment Setup

Experiments were conducted to evaluate the effectiveness of Autoencoders in feature reduction and their impact on attack detection performance in IoT networks show in Figure 1. The experimental procedure consisted of three main stages:

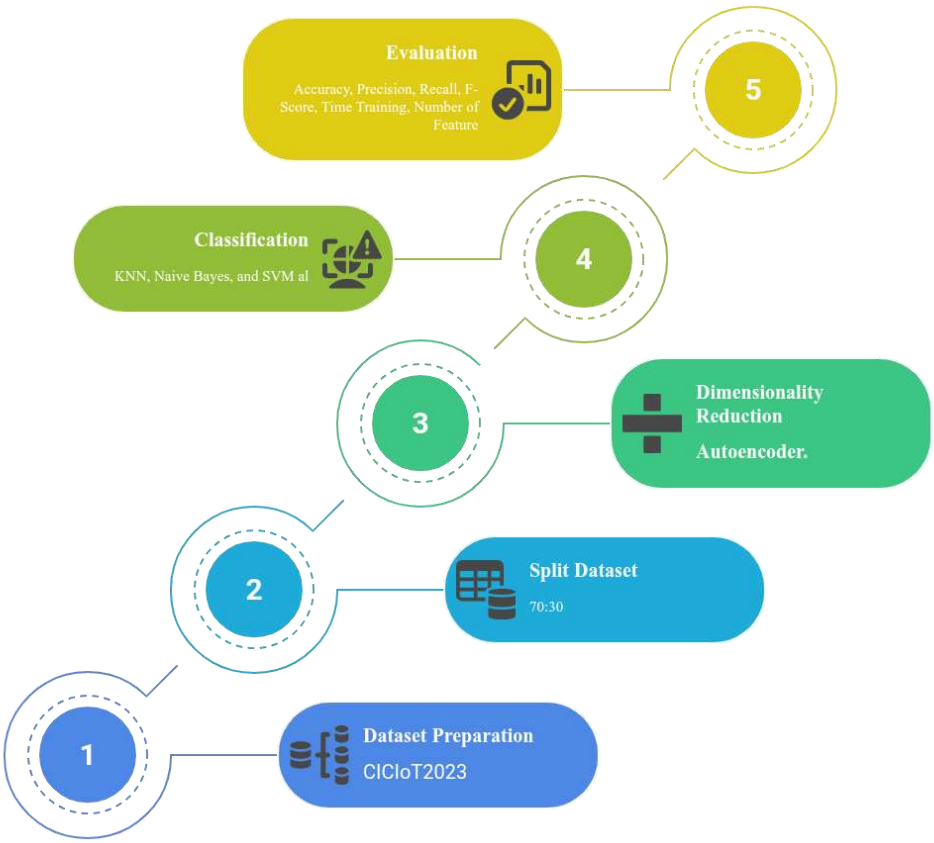

Figure 1. Experiment Setup

a) Dataset Preparation: The dataset, formatted as a CSV file, contains numerical features and attack labels. The data were cleaned, the labels were encoded using LabelEncoder, and the dataset was split into training and testing subsets in a 70:30 stratified ratio to preserve class distribution.

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 63−69
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.104
https://ejurnal.pdsi.or.id/index.php/bids/index

b) Dimensionality Reduction with Autoencoder: An Autoencoder was employed to reduce the number of features to 70% of the original set. The architecture consists of a symmetric encoder and decoder. The output from the bottleneck layer was used as a new, more compact yet informative feature representation.
c) Classification with KNN, Naive Bayes, and SVM: Three classification algorithms KNN, NB, and SVM were applied under two conditions: before and after feature reduction. The models were evaluated using accuracy, precision, recall, F1-score, training time, and number of features, in order to assess the impact of the Autoencoder on detection performance and computational efficiency.

2.2 Dataset

The dataset used in this study uses the IoT dataset from research [24] consisting of IEEE 802.11 traffic types, Zigbee-based, and Z-Wave. The purpose of this dataset is to provide a dataset for profiling, behavioral analysis, and vulnerability testing of various IoT devices. Where in the IoT dataset built by the University of New Brunswick, namely the CIC IoT Dataset 2023 which consists of IEEE 802.11, Zigbee-based and Z-Wave protocols. This dataset consists of DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai attacks.

2.3 Autoencoder for Reduction Feature

The Autoencoder is constructed as a neural network model comprising two primary components: an encoder and a decoder [8]. The encoder projects the input data into a lower-dimensional space known as the bottleneck layer, while the decoder attempts to reconstruct the original input from this compressed representation [25], [26]. In this study, the Autoencoder architecture employs an encoding dimension equal to 70% of the original number of features, serving as a trade-off between dimensionality reduction and information preservation. The model is trained in an unsupervised manner using only the feature data (X_train_scaled), with the mean_squared_error loss function and the Adam optimizer. Upon completion of training, the compressed representations from the bottleneck layer are extracted and used as the new features for the classification models.

2.4 Classification Algorithm

This study employs three commonly used machine learning algorithms for classification tasks to evaluate the performance of IoT network intrusion detection systems. The selection of these algorithms is based on their diverse operational characteristics and computational complexities, thereby providing a comprehensive perspective on the impact of feature reduction across different model types.

a) K-Nearest Neighbors (KNN):
KNN is an instance-based learning algorithm that classifies new data based on the similarity of distance to training data points [27]. In this study, Euclidean distance is used as the similarity metric. KNN is a relatively simple yet effective algorithm, particularly for low-dimensional data. Therefore, it is well-suited for observing the direct impact of feature reduction on classification performance.
b) Naive Bayes (NB):
Naive Bayes is a probabilistic classifier that assumes feature independence [28]. Despite its simplicity, it often delivers competitive results, especially on text data or datasets with clear probabilistic distributions. Since NB is sensitive to feature correlations, dimensionality reduction via Autoencoder may significantly influence its performance.
c) Support Vector Machine (SVM):
SVM is a margin-based classifier that aims to find an optimal hyperplane that separates data points from different classes in the feature space [29], [30]. In this study, a linear kernel is employed to maintain training efficiency. SVM is known for its robustness in handling high-dimensional data but typically involves higher computational costs, making it particularly relevant for evaluating the effect of dimensionality reduction on training efficiency.

Each algorithm is evaluated under two distinct scenarios to measure the impact of Autoencoder-based feature reduction:

a) Before optimization: The models are trained and tested using the full set of original features from the dataset, without any reduction.
b) After optimization: The models are trained and tested using the encoded features produced by the Autoencoder, which performs dimensionality reduction.

2.5 Evaluation

To assess the performance and efficiency of the IoT attack detection system, several evaluation metrics are [31], [32] employed as follows:

a) Accuracy: Measures the proportion of correct predictions out of the total number of instances. Higher accuracy indicates better overall model performance.
b) Precision: Indicates the proportion of correctly predicted positive instances among all predicted positives, reflecting the model's ability to avoid false positives .

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 63−69
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.104
https://ejurnal.pdsi.or.id/index.php/bids/index

c) Recall: Measures the proportion of actual positive instances that are correctly identified, reflecting the model's sensitivity in detecting true positives.
d) F1-Score: Represents the harmonic mean of precision and recall, and is used to evaluate the balance between them, especially in cases of class imbalance.
e) Training Time: Refers to the duration required for the model to learn from the training data. A shorter training time indicates a more computationally efficient model.
f) Number of Features: Denotes how many features are used during training. This metric helps evaluate the effectiveness of the feature reduction process.

3. RESULT AND DISCUSSION

This section presents the experimental results conducted to evaluate the effectiveness of Autoencoders in feature reduction for IoT data in the context of attack detection. The evaluation was performed by comparing the performance of three classification algorithma KNN, NB, and SVM under two scenarios: before and after feature reduction using an Autoencoder.

Following the Autoencoder training process, the high-dimensional features were successfully reduced to a lower-dimensional representation without losing essential data characteristics. The encoded features were observed to fall within the range of -1 to 1, which was influenced by the use of the ReLU activation function. In this experiment, the Autoencoder was constructed with two encoding layers, resulting in a final representation of 32 features, reduced from an initial total of 46 in Table 1.

Table 1. Results of Autoencoder

Feature Results from Autoencoder
3.047080039978027344e+00 8.325064182281494141e-01 2.571758747100830078e+00
1.509507656097412109e+00 0.000000000000000000e+00 0.000000000000000000e+00
3.520965576171875000e+00 4.332667827606201172e+00 5.680773854255676270e-01
1.210734605789184570e+00 0.000000000000000000e+00 4.230711936950683594e+00
9.504830837249755859e-01 2.595007181167602539e+00 2.128651142120361328e+00
1.131144762039184570e+00
1.153895378112792969e+00 8.497570753097534180e-01 1.068856835365295410e+00
8.369014859199523926e-01 0.000000000000000000e+00 1.240149736404418945e+00
6.148809790611267090e-01 3.259394168853759766e-01 1.369181871414184570e+00
1.376375317573547363e+00 4.661439061164855957e-01 0.000000000000000000e+00
5.340605974197387695e-03 5.231337547302246094e-01 0.000000000000000000e+00
0.000000000000000000e+00

Based on the evaluation results of IoT attack detection using the Autoencoder-based feature selection and machine learning methods, the next step is to calculate the performance of the detection system. This study employs four evaluation metrics: accuracy, precision, recall, and F1-score. The performance results of each classification method KNN, NB, and SVM are presented in Table 2. The table provides a detailed overview of the performance of each method across the four metrics for the corresponding test datasets.

Table 2. Results of Detection Attack

Dataset	KNN				NB				SVM			
	Acc	Pre	Rec	F-s	Acc	Pre	Rec	F-s	Acc	Pre	Rec	F-s
1	0.9313	0.9303	0.9313	0.9295	0.7138	0.7548	0.7138	0.6919	0.8105	08.225	0.8105	0.7746
2	0.9204	0.9191	0.9204	0.9180	0.7513	0.7544	0.7513	0.7243	0.8109	0.8238	0.8109	0.749
3	0.9176	0.9161	0.9176	0.9157	0.7080	0.7547	0.7080	0.6589	0.8010	0.8113	0.8010	0.7626
4	0.9232	0.9215	0.9232	0.9215	0.7381	0.7539	0.7381	0.6981	0.8032	0.8200	0.8032	0.7671
5	0.9245	0.9232	0.9245	0.9229	0.7244	0.7092	0.7244	0.6870	0.8013	0.8120	0.8013	0.7655
6	0.9202	0.9187	0.9202	0.9185	0.7239	0.7288	0.7239	0.6993	0.8003	0.8106	0.8003	0.7620
7	0.9214	0.9202	0.9214	0.9191	0.7136	0.7499	0.7136	0.6715	0.8014	0.8115	0.8014	0.7633
8	0.9243	0.9231	0.9243	0.9225	0.7431	0.7511	0.7431	0.7138	0.8035	0.8164	0.8035	0.7661
9	0.9306	0.9293	0.9306	0.9292	0.7143	0.7265	0.7143	0.6758	0.8015	0.8150	0.8015	0.7671
10	0.9302	0.9288	0.9302	0.9287	0.7035	0.7194	0.7035	0.6586	0.8001	0.8123	0.8001	0.7623

The experimental results presented in the table above demonstrate the performance of machine learning methods combined with Autoencoder-based feature selection. The results exhibit variations across accuracy, precision, recall, and F1-score metrics. These findings confirm the effectiveness of the Autoencoder in selecting relevant features, as validated through subsequent detection performance. The highest accuracy was achieved by the KNN method, reaching 93%, while the lowest accuracy was recorded by the Naive Bayes method at 70%. This variation in performance may be attributed to the diversity and complexity of the attack types within the dataset.

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 63−69
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.104
https://ejurnal.pdsi.or.id/index.php/bids/index

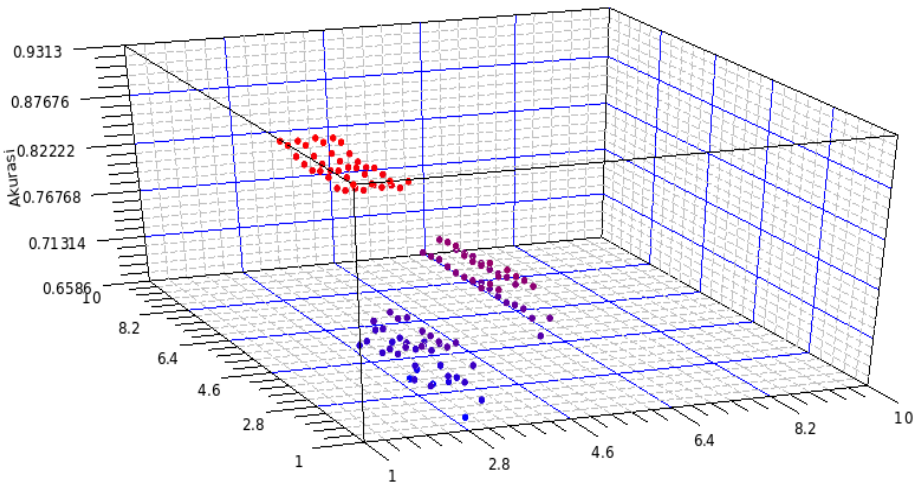

Figure 2. Detection performance test results

Figure 2 is the distribution of the performance of the attack detection system that has been carried out. In the red data distribution is the accuracy performance of the KNN method, then in purple is the result of the SVM method and blue is the performance of the NB method.

3.1 Discussion

The following section discusses the experimental results before and after applying the Autoencoder. This experiment was conducted to evaluate the effectiveness of the Autoencoder in reducing features in an IoT network attack dataset and to assess its impact on both model performance and computational efficiency. Three machine learning algorithms KNN, NB, and SVM were used for comparison. The evaluation considered several metrics, including accuracy, precision, recall, F1-score, training time, and the number of features, both before and after dimensionality reduction using the Autoencoder.

Table 3 presents a comparative analysis of the results obtained with and without the use of the Autoencoder for IoT attack detection. The Autoencoder successfully reduced the number of features from 46 to 32, achieving approximately a 30% reduction. While there was a slight decrease in accuracy for the KNN model (from 93.47% to 93.04%), its overall performance remained strong, with only minimal changes across all metrics. On the other hand, the Naive Bayes model showed an improvement in accuracy (from 71.39% to 72.04%) and F1-score (from 66.50% to 69.84%), indicating that the Autoencoder was effective in filtering out irrelevant or noisy features, thereby enhancing the model's performance. For the SVM model, both accuracy and F1-score experienced slight improvements, while the training time was significantly reduced from 575 seconds to 321 seconds demonstrating a substantial gain in computational efficiency.

Overall, the use of the Autoencoder proved to be an effective approach for feature reduction, without causing a significant drop in detection performance. In some cases, such as with the Naive Bayes and SVM models, it even led to performance improvements. These findings suggest that the Autoencoder not only simplifies the data by reducing dimensionality but also enhances feature representation, contributing to faster training and better classification outcomes. As such, the Autoencoder can be considered a promising technique for feature selection in IoT-based intrusion detection systems.

Table 3. Comparison of Results using Autoencoder

Model	Optimasion	Accuracy	Precision	Recall	F1-Score	Time Training (s)	Number of Features
KNN	Before	0.9347	0.9331	0.9347	0.9327	0.0233	46
KNN	After	0.9304	0.9285	0.9304	0.9282	0.0185	32
Naive Bayes	Before	0.7139	0.7239	0.7139	0.6650	0.2381	46
Naive Bayes	After	0.7204	0.7281	0.7204	0.6984	0.1333	32
SVM	Before	0.7909	0.7999	0.7909	0.7453	575.9240	46
SVM	After	0.7949	0.8011	0.7949	0.7517	321.2228	32

Bulletin of Informatics and Data Science
Vol. 3 No. 2, November 2024, Page 63−69
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v3i2.104
https://ejurnal.pdsi.or.id/index.php/bids/index

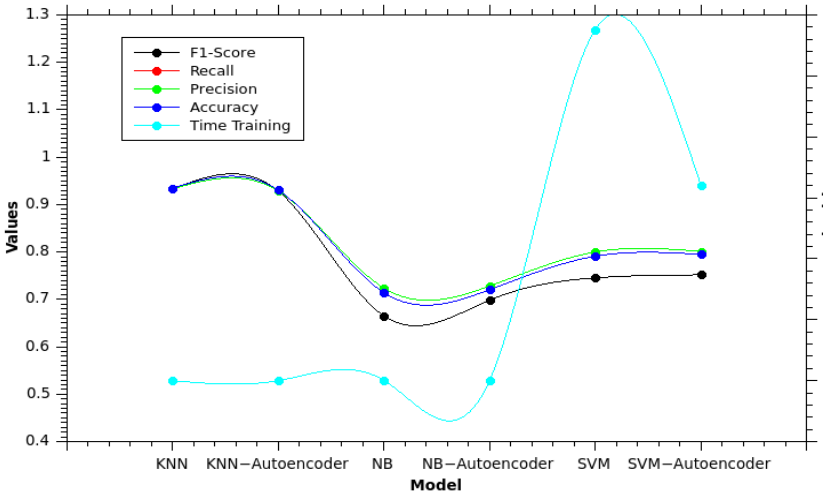

Figure 3. Comparison of detection performance

Based on Figure 3, it can be seen that the results of the performance evaluation of the three machine learning algorithms before and after the application of Autoencoder show different trends. In general, Autoencoder is able to reduce the number of features without sacrificing performance significantly. KNN shows a very small decrease in performance, but remains within the tolerance limit. In contrast, Naive Bayes experienced a significant increase in accuracy and F1-score after feature reduction, indicating that this method benefits from the data simplification process. SVM also shows improvements in performance metrics and training time efficiency, making it a responsive model to feature optimization. This graph strengthens the finding that Autoencoder not only functions as a dimensionality reduction tool but can also improve the quality of data representation, which has a positive impact on classification performance in IoT attack detection systems.

4. CONCLUSION

This study evaluates the effectiveness of Autoencoder as a feature reduction method to detect attacks on IoT networks using KNN, Naive Bayes, and SVM classification algorithms. Based on the experimental results, Autoencoder is able to reduce the number of features without losing important information needed in the classification process. The test results show that the KNN algorithm provides the best performance, with the highest accuracy reaching 93%. Meanwhile, SVM shows stable performance with an accuracy of around 80%, and Naive Bayes has the lowest accuracy of around 70%. Although there is a slight decrease in performance in some algorithms after feature reduction, the system efficiency increases significantly, especially in terms of training time and the number of features used. Overall, Autoencoder is proven to be effective in simplifying complex IoT data, as well as supporting the attack detection process efficiently and accurately. These results indicate that this approach has the potential to be applied to real attack detection systems in IoT environments with limited computing resources. Future research will use Deep Embedded Feature Selection for feature selection and propose using deep learning.

REFERENCES

[1] S. Budiyanto, L. M. Silalahi, A. R. Hakim, A. Hamid, and D. Hanafi, “Vulnerability analysis on internet of things (IoT) networks using raspberry pi and open web application security project (OWASP),” Proceedings - 2024 FORTEI-International Conference on Electrical Engineering: Empowering Innovations : Navigating The Future Of Semiconductor Industry, FORTEI-ICEE 2024, no. October, pp. 58–63, 2024, doi: 10.1109/FORTEI-ICEE64706.2024.10824490.

[2] D. D. Malicious, J. Jan, J. Feb, and J. May, 2022 index IEEE internet of things journal vol. 9, vol. 9. 2023. doi: 10.1109/jiot.2022.3232257.

[3] E. Gelenbe, B. C. Gül, and M. Nakıp, “DISFIDA: Distributed self-supervised federated intrusion detection algorithm with online learning for health internet of things and internet of vehicles,” Internet of Things (Netherlands), vol. 28, no. August, p. 101340, 2024, doi: 10.1016/j.iot.2024.101340.

[4] K. Kurniabudi, E. A. Winanto, L. Y. Astri, and S. Sharipuddin, “Ensemble method for anomaly detection on the internet of things,” IJCCS (Indonesian Journal of Computing and Cybernetics Systems), vol. 18, no. 1, p. 25, 2024, doi: 10.22146/ijccs.85834.

[5] Sharipuddin, E. A. Winanto, Z. Z. Mohtar, Kurniabudi, I. S. Wijaya, and D. Sandra, “Improvement detection system on complex network using hybrid deep belief network and selection features,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 31, no. 1, pp. 470–479, 2023, doi: 10.11591/ijeecs.v31.i1.pp470-479.

[6] K. G. Maheswari, C. Siva, and G. Nalinipriya, “Optimal cluster based feature selection for intrusion detection system in web and cloud computing environment using hybrid teacher learning optimization enables deep recurrent neural network,” Computer Communications, vol. 202, no. September 2022, pp. 145–153, 2023, doi: 10.1016/j.comcom.2023.02.003.

[7]R. SaiSindhuTheja and G. K. Shyam, "An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment," Applied Soft Computing, vol. 100, p. 106997, 2021, doi: 10.1016/j.asoc.2020.106997.

[8] I. K. Thajeel, K. Samsudin, S. J. Hashim, and F. Hashim, "Dynamic feature selection model for adaptive cross site scripting attack detection using developed multi-agent deep Q learning model," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 6, p. 101490, 2023, doi: 10.1016/j.jksuci.2023.01.012.

[9] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, "ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model," Scientific Reports, vol. 12, no. 1, pp. 1–18, 2022, doi: 10.1038/s41598-022-19366-3.

[10] S. Waskle, L. Parashar, and U. Singh, "Intrusion detection system using PCA with random forest approach," in Proceedings of the international conference on electronics and sustainable communication systems, ICESC 2020, 2020, pp. 803–808. doi: 10.1109/ICESC48915.2020.9155656

[11] Z. Liu, Y. Fang, C. Huang, and Y. Xu, "MFXSS: An effective XSS vulnerability detection method in JavaScript based on multi-feature model," *Computers & Security*, vol. 124, 103015, 2023, doi: 10.1016/j.cose.2022.103015.

[12] C. Wang, H. Zhou, Z. Hao, S. Hu, J. Li, X. Zhang, B. Jiang, and X. Chen, "Network traffic analysis over clustering-based collective anomaly detection," *Computer Networks*, vol. 205, 108760, 2022, doi: 10.1016/j.comnet.2022.108760.

[13] S. Vaishnodevi, D . V. Kumar, G. Murali, M. Azhagiri, C. A. Madhuvappan, and K. Sathishkumar, "Network traffic examination for network intrusion detection in IOV using autoencoder and decoder," in 2nd international conference on intelligent cyber physical systems and internet of things, ICoICI 2024 - proceedings, IEEE, 2024, pp. 13–18. doi: 10.1109/ICoICI62503.2024.10696347.

[14] S. Ustebay, Z. Turgut, and M. A. Aydin, "Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier," International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, IBIGDELFT 2018 - Proceedings, pp. 71–76, 2019, doi: 10.1109/IBIGDELFT.2018.8625318.

[15] J. Lee, J. G. Pak, and M. Lee, "Network intrusion detection system using feature extraction based on deep sparse autoencoder," International Conference on ICT Convergence, vol. 2020-Octob, pp. 1282–1287, 2020, doi: 10.1109/ICTC49870.2020.9289253.

[16] S. Zavrak and M. Iskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," IEEE Access, vol. 8, pp. 108346–108358, 2020, doi: 10.1109/ACCESS.2020.3001350.

[17] A. Bhardwaj, V. Mangat, and R. Vig, "Hyperband tuned deep neural network with well posed stacked sparse AutoEncoder for detection of DDoS attacks in cloud," IEEE Access, vol. 8, pp. 181916–181929, 2020, doi: 10.1109/access.2020.3028690.

[18] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment," Sensors, vol. 23, no. 13, 2023, doi: 10.3390/s23135941.

[19] Y. Meidan, M. Bohadana, and D. Breitenbacher, "N-BaIoT — network-based detection of IoT botnet attacks using deep autoencoders," IEEE Pervasive Computing, no. September, pp. 12–22, 2018.

[20] A. Sarıkaya, B. G. Kılıç, and M. Demirci, "RAIDS: Robust autoencoder-based intrusion detection system model against adversarial attacks," Computers and Security, vol. 135, no. March, p. 103483, 2023, doi: 10.1016/j.cose.2023.103483.

[21] W. Wang, S. M. Sadjadi, and N. Rishe, "Curse of feature selection: a comparison experiment of DDoS detection using classification techniques," Proceedings - 20th IEEE International Symposium on Parallel and Distributed Processing with Applications, 12th IEEE International Conference on Big Data and Cloud Computing, 12th IEEE International Conference on Sustainable Computing and Communications an, pp. 262–269, 2022, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00040.

[22] T. Wisanwanichthan and M. Thammawichai, "A double-layered hybrid approach for network intrusion detection system using combined naive bayes and SVM," IEEE Access, vol. 9, pp. 138432–138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

[23] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," Computers and Security, vol. 103, p. 102158, 2021, doi: 10.1016/j.cose.2020.102158.

[24] B. Xu, S. Chen, H. Zhang, and T. Wu, "Incremental k-NN SVM method in intrusion detection," Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, vol. 2017-Novem, pp. 712–717, 2018, doi: 10.1109/ICSESS.2017.8343013

[25] Y. Song, S. Hyun, and Y.-G. Cheong, "Analysis of autoencoders for network intrusion detection," Sensors, vol. 21, no. 13, p. 4294, 2021, doi: 10.3390/s21134294.

[26] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and M. Helal, "Intrusion detection in IoT systems using denoising autoencoder," IEEE Access, vol. 10, pp. 135308–135318, 2022, doi: 10.1109/ACCESS.2022.3222774.

[27] S. A. H. Ayubkhan, W.-S. Yap, E. Morris, and M. B. K. Rawthar, "A practical intrusion detection system based on denoising autoencoder and LightGBM classifier with improved detection performance," Journal of Ambient Intelligence and Humanized Computing, vol. 14, pp. 7427–7452, 2022, doi: 10.1007/s12652-022-03653-8.

[28] M. A. Alsoufi, M. M. Siraj, and F. A. Sim, "An anomaly intrusion detection system in IoT based on autoencoder: a review," in Advanced intelligent computing technologies and applications, Springer, 2024, pp. 331–343. doi: 10.1007/978-981-99-1299-6_27.

[29] J.-f. Cui, H. Xia, R. Zhang, B.-x. Hu, and X.-g. Cheng, "Optimization scheme for intrusion detection scheme GBDT in edge computing center," *Computer Communications*, vol. 168, pp. 136–145, 2021, doi: 10.1016/j.comcom.2020.12.007.

[30] P. Mahadevappa, R. K. Murugesan, R. Al-amri, R. Thabit, A. H. Al-Ghushami, and G. Alkawsi, "A secure edge computing model using machine learning and IDS to detect and isolate intruders," MethodsX, vol. 12, no. November 2023, p. 102597, 2024, doi: 10.1016/j.mex.2024.102597.

[31] F. Ullah, S. Ullah, G. Srivastava, and J. C. W. Lin, "IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," Digital Communications and Networks, vol. 10, no. 1, pp. 190–204, 2024, doi: 10.1016/j.dcan.2023.03.008.

[32] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar, "Anomalous communications detection in IoT networks using sparse autoencoders," 2019. [Online]. Available: https://arxiv.org/abs/1912.11831