PENINSULA PUBLISHING LLC

SHIFRA

Vol. **(2025)**, 2025, **pp**. 126–136
ISSN: 3078-3186

Research Article

An Intelligent Intrusion Detection Framework Using Deep Learning and Unsupervised Feature Selection for Industry 4.0

Vugar Abdullayev [1,2,*], Nazila Ragimova [1],

[1] *Department of Computer Engineering, Azerbaijan State Oil and Industry University, Baku, Azerbaijan*

[2] *Department of Information Technology and Systems, Azerbaijan University of Architecture and Construction, Azerbaijan*

ARTICLE INFO

Article History

Received 10 Apr 2025

Revised: 2 Jun 2025
Accepted 1 Jul 2025
Published 15 Jul 2025

Keywords

Industry 4.0,

Intrusion Detection System (IDS),

Unsupervised Learning,

Cybersecurity,

Anomaly Detection.

ABSTRACT

Manufacturing has been revolutionized by the integration of advanced technologies such as the Internet of Things, cyber-physical systems, and cloud computing, but it is also exposed to a variety of cyber-attacks because of these technologies. A system for detecting unsupervised intrusions is proposed as an answer to these growing security challenges, which we call the Unsupervised Intrusion Detection System for Industry 4.0. Using the isolation forest method, the framework identifies anomalous network traffic anomalies through random forest-based feature selection. Based on deep learning and unsupervised anomaly detection, the proposed system is more accurate, computationally efficient, and reduces false positives than traditional intrusion detection systems. With the proposed model, interconnected industrial systems will be protected from evolving cyber threats, demonstrating enhanced security for Industry 4.0 systems.

1. INTRODUCTION

Cloud computing and cyber-physical systems are bringing unprecedented efficiency and innovation to global industries as the industry 4.0 revolution continues to reshape global industries. As a result of this technological transformation, however, there are also new vulnerabilities, particularly in the form of cyber threats targeting interconnected industrial systems. The protection of these infrastructures against potential attacks relies heavily on intrusion detection systems (IDS)[1]. A solution to this challenge can be found in intelligent intrusion detection frameworks that use deep learning and unsupervised feature selection techniques to identify intrusions. A powerful neural network is employed in conjunction with intelligent feature selection to enhance the accuracy and reliability of anomaly detection. A framework of this type is developed and implemented, demonstrating its potential to protect Industry 4.0 systems from cyber threats that are constantly evolving. Manufacturing 4.0 is about making factories more flexible, customizable, efficient, affordable, safer, and more responsible. Digitalization and connectivity are at the core of the fourth industrial revolution, which is bringing about a paradigm shift in many fields. An autonomous network connected in real-time that interacts with each other [2], [3]. A data-driven approach can help Industry 4.0 systems operate more efficiently and make better decisions. The generation and secure accumulation of personal information are also important issues in the privacy and transparency arenas [4].

The machines in these systems heavily utilize cyber-physical systems (CPS) and the Internet of Things (IoT), making them more vulnerable to cyber-attacks[5]. A highly volatile network environment, despite highly efficient and secure manufacturing machinery, still leaves organizations highly susceptible to intrusions and privacy breaches. An example of an intrusion would be a denial-of-service attack, unauthorized access, identity theft, a buffer overflow, and others. As a result, the line between authentication levels is blurred since attacks could come from any component of the overall system [6]. Integrating I4.0 into legacy systems can also compromise their security by exposing sensitive information. There is a reason for this since several complex technologies have been integrated. The more access consumers and users have to business and control systems, the more challenging security becomes in a smart industry [7], [8]. Predictions are more accurate, and low overfitting is associated with them. Compute costs are reduced, and additional analytics are supported

*Corresponding author email: abdulvugar@mail.ru
DOI: https://doi.org/10.70470/SHIFRA/2025/009

by RF-selected relevant features. Compute costs are reduced, and additional analytics are supported by RF-selected relevant features. An unsupervised anomaly detection approach, also referred to as intrusion detection (ID), is applied to the dataset after the optimal features have been extracted. A method of detecting intrusions may be classified as either supervised, unsupervised, or semi-supervised. No label information is included in the collected data in an application context. Additionally, most ID models are incapable of detecting undefined attacks and must be trained periodically. This section provides a detailed explanation of the proposed unsupervised intrusion detection system for Industry 4.0. Using feature optimization, it detects intrusions in a generalized manner. A training algorithm exploits optimal features in the data and trains the IF to detect anomalies in the selected features. An efficient approach is proposed for handling unbalanced datasets through data sampling. Aside from that, it utilizes data features that are optimized to detect intrusions.

2. RELATED WORK

Cyber security challenges arising from Industry 4.0's rapid advancements have posed significant challenges in securing interconnected systems. These challenges have prompted researchers to develop intelligent intrusion detection systems (IDS). To detect various attacks, hybrid deep learning algorithms are employed, including autoencoders, long-short-term memory networks (LSTMs), and convolutional neural networks (CNNs) for robust feature extraction and classification [9], [10]. Additionally, feature learning techniques have been used to reduce computational costs and dimensionality, resulting in high accuracy in identifying attacks from datasets like CICIDS-2018. Furthermore, frameworks leveraging swarm-based deep learning classifiers have also demonstrated effectiveness in multi-cloud IoT environments by employing innovative methods such as opposition-based learning for feature selection and 2D-array-based CNNs for classification. IDS effectiveness and efficiency can be enhanced in complex industrial settings by combining deep learning with feature selection techniques [11]. It detects anomalies, routes traffic, and recognizes suspicious activities [12], ensuring security and recognizing anomalous behaviour as it captures and decodes packets. Because it can track both visible and invisible threats (zero-day threats), it uses standard data to create patterns. It treats any deviation from them as an intrusion [13] for testing and verifying the system [14] utilized Modbus/TCP message networking streams to optimize One-Class Support Vector Machine (OCSVM) performance through Particle Swarm Optimization (PSO). According to [13], network traces from offline SCADA data were used to develop this IDS/ADS. In order to construct an IDS, the authors used K-NN classifiers based on Modbus/TCP protocol settings [15]. However, they were designed for configurations with strong FPR, so they don't work well in all situations. According to [16], multiple attacks can be effectively identified using a broad range of OCSVM frameworks for enhanced intrusion detection systems (IDS). However, this computer was very computationally intensive when operating, with a high false-warning rate. The Authors in [17] proposed a Modbus/TCP protocol-infiltrated assault detection system based on SCADA mechanisms and an SVM algorithm[18].

The learning firewall in [19] writes conservative preventive rules based on tagged samples to prevent false positives. As opposed to traditional classifiers, which only consider accuracy to be the primary decision criterion, our family of classifiers focuses on zero false positives as the primary decision criterion. To achieve this goal, the authors propose a generic iterative technique based on naive modifications to current classifiers, such as SVM. CART is used to create a firewall for a power grid monitoring system using the proposed classifier. A test of the technique was carried out on the KDD CUP'99 dataset to see how well it worked. We are able to demonstrate the effectiveness of our strategy based on the outcomes. Researchers have investigated IDSs that use subsurface networks to detect irregularities on a host or a network [20]. There are one or two hidden layers in an ANN with a shallow network, whereas there are several hidden layers in an ANN with a deep network, resulting in different hidden state architectures. Academics and industrial researchers use machine-learning techniques such as deep learning because they teach detailed computational mechanisms that mimic human behaviour[18], [21]. IDS architectures face a significant challenge when analyzing the subsequent large amounts of data generated by the rapid conversion of system signals into massive datasets [22]. Detecting DoS attacks is based on rule-based approaches proposed in [23] using domain expert knowledge. DoS attacks were identified using a rule-based classification algorithm, which was then verified by a domain expert based on the rules from the rule base. Spatial removal, also known as feature selection, can be used to convert databases from an elevated spatial domain to a lesser spatial domain that is more representative of the problem domain. The introduction of feature collection allows for the elimination of unconnected variables without lowering the value of data for the detection model. Most datasets contain several attributes, but there are few examples, according to [24], where feature selection is needed or used. It has been shown by [25] that adversaries can exploit the specific characteristics of devices, infrastructure, communications, and services using an attack taxonomy that takes into account each layer.

Furthermore, nine real-world cybersecurity incidents related to IoT devices used in commercial, industrial, and consumer sectors illustrate vulnerabilities, exploitation techniques, attacks, impacts, and mitigation strategies. This taxonomy provides a systematic method for categorizing attacks according to the layer impacted and their impact. At the same time, the presented examples highlight the fundamental security vulnerabilities of IoT systems and their potential attack implications. Using smart rule-based identification schemes, the authors are detecting Denial of Service (DoS) attacks on cloud servers [22]. Using scoring and rating algorithms, the best features of a cloud service and assault identities were

selected. The selected features were grouped based on quality expertise to find assaults. Their proposed model provides greater protection and lower false alarm rates. Because of the complexity of attacks, confusion risks were not addressed.

3. PROPOSED METHODOLOGY

The paper presents a system for detecting IoT intrusions using denoising autoencoders (DAEs). Rather than relying on rule-based or signature-based approaches to detect anomalies in IoT network data, our methodology utilizes unsupervised learning with DAEs. Its sophisticated feature engineering approach makes our solution uniquely suited for IoT contexts, improving the model's ability to adjust to network conditions. The dataset's resilience is enhanced by dividing it into equal training and testing sets, reducing bias and ensuring generalizability. For deep learning models to be efficiently trained, the data preparation methods must be rigorously validated to ensure reliability and precision. A systematic approach to IoT security based on DAEs establishes a standard for comprehensive, data-driven approaches to intrusion detection. In this study, a well-structured and flexible framework for tackling the intricacies of IoT network security is provided with deep learning.

3. 1 Pre-Processing

Detecting intrusions in IoT systems using denoising autoencoders (DAE) is a crucial area of cybersecurity research in light of the constantly changing threats in networked environments. Several key characteristics distinguish this methodology from previous approaches, including the use of state-of-the-art pre-processing methods with creative DAE applications. Data preparation is at the core of our methodology, ensuring that the dataset is optimized for subsequent machine learning operations. The dataset is standardized by using the scikit-learn StandardScaler function. To ensure that machine learning algorithms perform well, feature scales must be uniform. The purpose of this step is to reduce the biases associated with different scales, as well as enhance the model's interpretability and generalizability across a range of IoT datasets. The key component of our methodology is how we systematically encode category labels using Label Encoder, which makes it easier to incorporate categorical data into the model. Our algorithm becomes better at identifying trends and abnormalities linked to incursion behaviours when categorical data is converted into numerical representations. A pre-processing approach is essential to simplifying data preparation, enabling more effective modelling and analysis without sacrificing the accuracy of data from IoT devices.

DAEs are used in our approach to detect intrusions, which is different from conventional approaches that frequently employ supervised learning [26], [27]. The development of resilient representations enables the identification of underlying patterns in noisy inputs using neural networks called DAEs. The DAE can detect deviations indicative of malicious activity in IoT networks by being trained to minimize reconstruction errors. Additionally, this unsupervised learning technique can detect new threats without requiring large labelled datasets, which are often costly and difficult to acquire in IoT settings.

3. 2 Model Architecture

Using denoising autoencoders, this study proposes a new method for detecting intrusions in IoT systems. As opposed to traditional approaches that merely identify anomalies or use simple models, the DAE architecture in this project is specifically designed for the unique properties of IoT data streams. Using the DAE's learning capability, the technique identifies potential intrusions and typical behaviours. Data from Internet of Things networks is often noisy and originates from a variety of unreliable sources, which reduces their complexity. To implement this methodology, it is essential to have a strategic network design for encoders and decoders. Conventional autoencoders are capable of noise reduction and compression, but this adaptation focuses on decoding to reconstruct data unique to the Internet of Things precisely. Encoders use dense layers with ReLU activations, and decoders use sigmoid activations to add dimensions [28]. Therefore, the model can minimize reconstruction errors while capturing intricate IoT patterns. A structured approach increases the accuracy of anomaly detection and improves feature extraction for subsequent classification tasks based on the encoder's learned representations. Adding a derived encoder model for categorization to the DAE expands its utility beyond reconstruction. With this method, intrusion events can be classified faster and more accurately by retaining decoded representations that contain valuable IoT data properties. There are two distinct applications of autoencoder architecture in this study, making it different from others in its field. In addition to improving reconstruction accuracy, the algorithm leverages learned representations for performing sophisticated analytical tasks crucial to IoT security. By applying Denoising Autoencoder technology to IoT contexts, intrusion detection systems should be more precise and adaptable than they are today.

3. 3 Model Training

DAE is trained to detect intrusion in IoT systems using our unique approach that combines intensive pretraining and fine-tuning stages. First, the DAE goes through pretraining to become capable of reassembling clean data from noisy inputs and extracting significant characteristics required for anomaly detection. During this stage, reconstruction error is reduced, allowing the model to weed out unimportant data and focus on patterns that suggest intrusion attempts. Our fine-tuning strategy involves adding more layers after pretraining and fixing the encoder's parameters after that. As a result, the encoded

representations produced by the DAE can be used for reliable intrusion detection. Classifiers are optimized to identify intrusion risks using techniques such as binary cross-entropy loss functions and Adam optimizations [29], [30]. IoT settings must be protected against cyberattacks by this method. The model must be thoroughly evaluated across different datasets in order to be used as a reliable tool in the real world. Thus, the model is suitable for various IoT applications due to its strong cybersecurity defence and flexibility in changing attack techniques. Using DAE-based solutions, we demonstrate how security can be enhanced in networked IoT environments, how infrastructure and sensitive data can be protected from cyberattacks, and how detection accuracy can be increased. A detailed explanation of the training model is provided, with a focus on practical efficacy and scalability.

1. Pretraining: Denoising autoencoders can be pre-trained to acquire a condensed representation of input characteristics using training data. During pretraining, the autoencoder is trained to reduce reconstruction errors in order to minimize input and output data discrepancies. As a result of this technique, the model is able to gather significant qualities by removing noise and irrelevant data.
2. Fine-Tuning for Classification: Autoencoder models are created by fixing encoder layers and adding more dense layers to create classifiers after the pretraining phase. The fixed encoder layers are used as inputs to train the classifier with the encoded representations produced by the encoders. In IoT systems, intrusion risks can be detected using binary cross-entropy loss functions and Adam optimization algorithms.
3. Model Evaluation: Using the testing data, the trained classifier was able to predict intrusion assaults correctly. This is indicated by the test accuracy metric, which indicates the proportion of cases accurately classified from the total number of samples. Using existing methods, we have developed a versatile and reliable model that achieves excellent performance on a variety of datasets associated with incursions. With this technique, we can successfully identify intrusions, thereby increasing dependability and flexibility in protecting our network against cyberattacks.
4. Development of a Novel Approach: To detect unauthorized access to IoT devices, we propose an approach based on denoising autoencoding models. Using unsupervised learning algorithms, this innovative technique accurately detects intrusion attacks and extracts useful features from network traffic.
5. Effective Detection of Intrusion Attacks: Denoising Autoencoders for Internet of Things intrusion detection are presented in this paper. Intensive testing and assessment of the model demonstrate its ability to distinguish benign and hostile network events, resulting in high accuracy and low loss values.

3. 4 Unsupervised Learning

As a result of the Denoising Autoencoder model, salient characteristics from the input data are captured and enhanced to facilitate supervised classification of the data. In IoT systems, the model aims to reduce potential security threats by effectively identifying and categorizing representations and information. We utilize a deep learning model called denoising autoencoder for our approach. Using a denoising autoencoder, data that is damaged or noisy can be removed by learning how to reconstruct the original data. The training aims to minimize disparities between original and reconstructed data. It is possible to build deep networks by stacking denoising autoencoders. Architecture can accept a number of other types of data, including text, audio, and images. It can incorporate personalized noise, such as Gaussian noise or salt-and-pepper noise.

1. The traditional auto-encoder minimizes errors.

$$L\left(x, g\left(f(x)\right)\right) \quad (1)$$

- In $L2$ the norm of difference is calculated as the mean squared error of $g(f(x))$ relative to X.
  A DAE minimizes the amount of energy consumed.

$$L\left(x, g\left(f(x)\right)\right) \quad (2)$$

- Noise corrupts the original copy of x, and l is the corrupted copy of L
- The autoencoder must undo this corruption rather than copy it.

Encoding is reversed when decoding, which reverts the encoded data back to its original form in the data space. As part of the training process, the autoencoder is provided with both clean and noisy examples. This experiment is aimed at training an encoder-decoder function to convert noisy inputs into clear ones. Below is a diagram illustrating the phases of training and testing for the nl DAE. nl DAE parameters can be optimized using equation 3 as follows for all $i \in \{1,\cdots, M\}$: $\theta * nl$:

$$\theta_{nl}^{*}, \theta_{nl}^{\prime *} = argmin_{\theta,\theta'} \frac{1}{M} \sum_{i=1}^{M} L^{ni}, g^{\theta'}\left(f_{\theta}(y^{i})\right) \quad (3)$$

It is only by replacing XI by I that (1) differs from (2). Accordingly, $\hat{x}(j)nl$ denotes the jth regenerated data based on nl DAE, which can be expressed as follows for all $j \in \{1, \ldots, L\}$: $\hat{x}j$ in equation 4.

$$\hat{x}_{nl}^{j} = y^{j} - g_{\theta'} * nl\left(f_{\theta * n} l(y^{j})\right) \quad (4)$$

3. 5 LSTM-DAE Model

With the LSTMM-DAE model, abnormal signals are extracted, classified, and recognized using LSTMs and autoencoders. To maximize the effectiveness of pipeline leakage data analysis, we exploit a relationship between the AEs and the time series using unsupervised learning with linear support vector machines. A diagram of the LSTM-DAE model framework, which includes encoders and decoders, can be found in Figure 4. There is no need to tag signals with a time sequence in order to learn their characteristics. Three layers are present in the encoder: an input layer, a fully connected layer (which retains most spatial dimension information), and two layers of LSTMs. There are two LSTM layers in the encoder: an input layer and a fully connected layer (which is capable of retaining most spatial information). The time series length is maintained throughout the process by encoding spatial dimensions into 32-dimensional features. LSTM layers (128 nodes) are used in conjunction with the fully connected layers (128 nodes) to extract digital signal features from input data. Instead of encoding 32-dimensional features, the decoder reconstructs their spatial dimensions. Softmax transforms the hidden layer conduction features into the classifier's outputs by using the LSTM-DAE. There is a connection between the softmax layer and the decoder's output.

$$P_m = Softmax(a_m) = \frac{\exp\,(a_m)}{\sum_{j=1}^{M} \exp\,(a_j)} \tag{5}$$

Each of the m categories in $m = 1,2,\dots,M$, p_m has a probability of classification and a_m represents the neurons that need to be activated.

Cross entropy loss, therefore, serves as the objective function:

$$L(\theta) = \sum_{m=1}^{M} y_m \log\,(P_m) \tag{6}$$

A tag's expected value is represented by y_m in this example.

This study is conducted based on the LSTM-DAE model. LSTM-DAE classification models can be trained using pipeline data by using characteristic parameters as input. The data sets used in the analysis are as follows:

$$X = [X_1, X_2, \dots, X_n] \tag{7}$$

A complete data set is fed to the LSTM-DAE; a single feature parameter data set is fed to it, and a number of effective features are fed to it. Normalization results in the following formula:

$$y = \frac{(y_{max} - y_{min})(x - x_{min})}{(x_{max} - x_{min})} + y_{min} \tag{8}$$

A vector is represented by y after normalization, x before normalization, max after normalization, min after normalization, and the same for x before normalization.

It is possible to obtain hyperparametric optimal solutions in particle swarm optimization. The particles update themselves iteratively by tracking two extremes.

3. 6 Dataset Description

As a result of a collaboration between the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE) in 2018, this dataset has been created. Seventy-five network data features were extracted using their newly developed tool, CICFlowMeter-V3. Out of 16,232,943 flows, 14,484,708 (83.17%) were benign flow types, and 2,748,235 (16.93%) were attack types. There are five features in its flow identifier list: the Dst IP, the Flow ID, the Src IP, the Src Port, the Dst Port, and the Time Stamp. The attacks included bot attacks, brute force attacks, DoS attacks, DDoS attacks, infiltration attacks, and web attacks.

This database depicts a wide variety of assaults, including typical traffic on IoT networks. Data from the Bot-IoT database includes keylogging, server and OS scanning, DDoS attacks, and data exfiltration. Bot-IoT was created as a new dataset to allow comparisons with older datasets.

3. 7 Measures for evaluating

This study utilizes a variety of evaluation metrics to evaluate the proposed model's performance.

- Accuracy: Calculated using equation (9), it represents the average correctly identified IoT traffic.

$$Accuracy(Acc) = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{9}$$

- Precision: Equation (10) defines positively identified samples as essentially positive samples.

$$Precision(PRC) = \frac{TP}{TP + FP} \times 100 \tag{10}$$

- Recall: This measure measures the proportion of real positive instances detected as positive based on an equation (11).

$$Recall(RCL) = \frac{TP}{TP + FN} \times 100 \quad (11)$$

- F1-measure: In equation (12), it reaches its highest value at 1 and its lowest value at 0 and is calculated as a weighted average of the RCL and PRC.

$$F1 - Score(F1) = 2 * \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

4. RESULT AND DISCUSSION

Based on various configurations of hidden layers, Figure 1 shows the trend in loss function for various models. It appears that adding hidden layers to the models does not make them more capable of reconstructing input data. At the beginning of the epoch, the single hidden layer model with the smallest loss value outperforms the other two autoencoders. According to [31], increasing hidden layers results in a greater loss error in autoencoders.

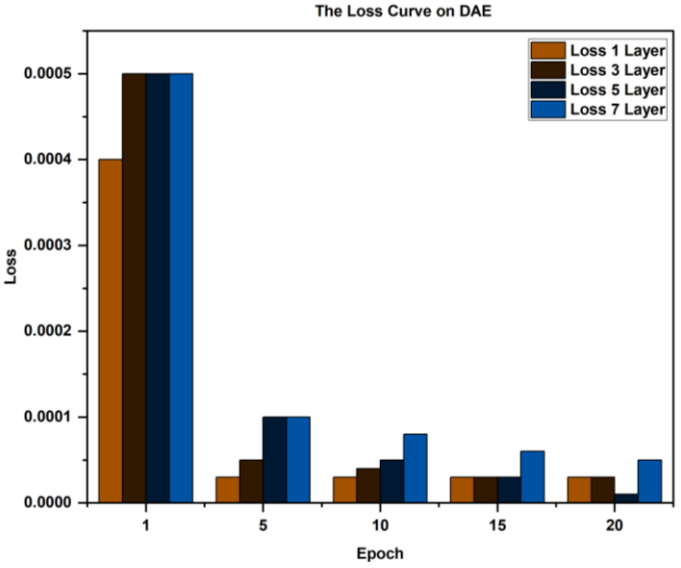


Fig. 1. An example of a loss curve in a deep autoencoder (DAE).

The loss value decreases as epochs are added, and it is close to zero as epochs are added. DAE and LSTM-DAE models present loss values after 20 epochs, whereas DCAE presents them after 10 epochs. There is no difference in loss value between different configurations of the LSTM-DAE model, except in scenarios where three hidden layers are involved. Network traffic datasets exhibit a time-series pattern, which can explain the effectiveness of the LSTM-DAE model [32]. Because of this, autoencoders incorporating LSTM layers, whose internal memory can utilize input sequences, perform better at reconstruction.

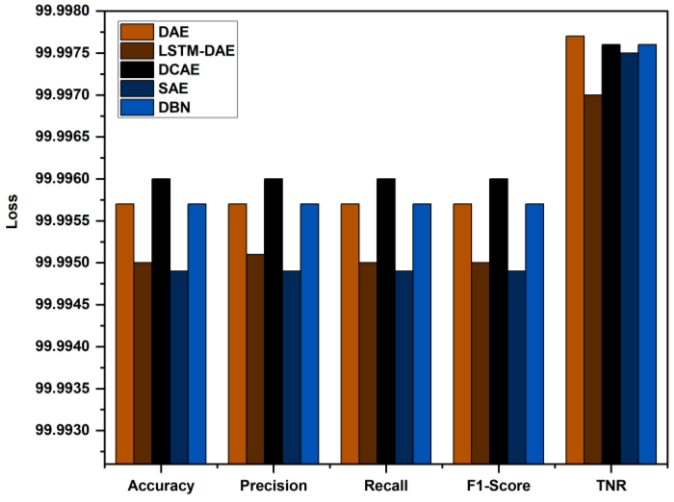


Fig. 2. Mean squared error loss.

According to Figure 3, all models have minimal false positives (FP). False positive rates (FPRs) are lowest with the DAE model, while they are highest with the LSTM-DAE model, although they are negligible. When the BoT-IoT dataset was fine-tuned, the transfer learning model's results were relatively consistent. A DAE model's performance is not significantly affected by the number of hidden layers except during transfer.

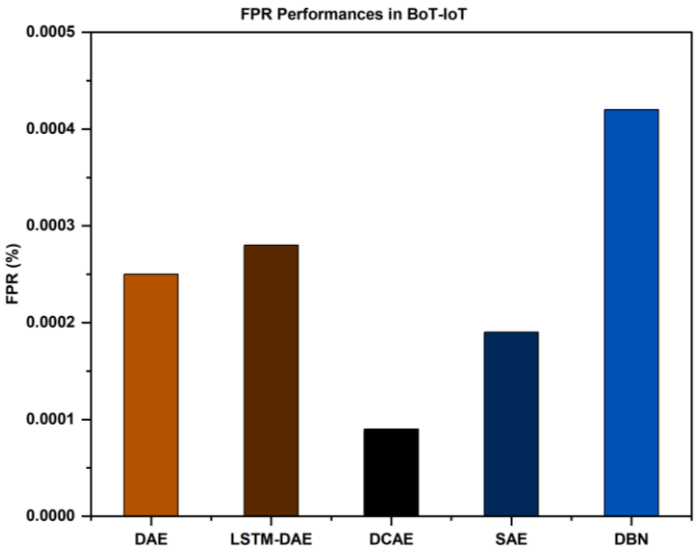
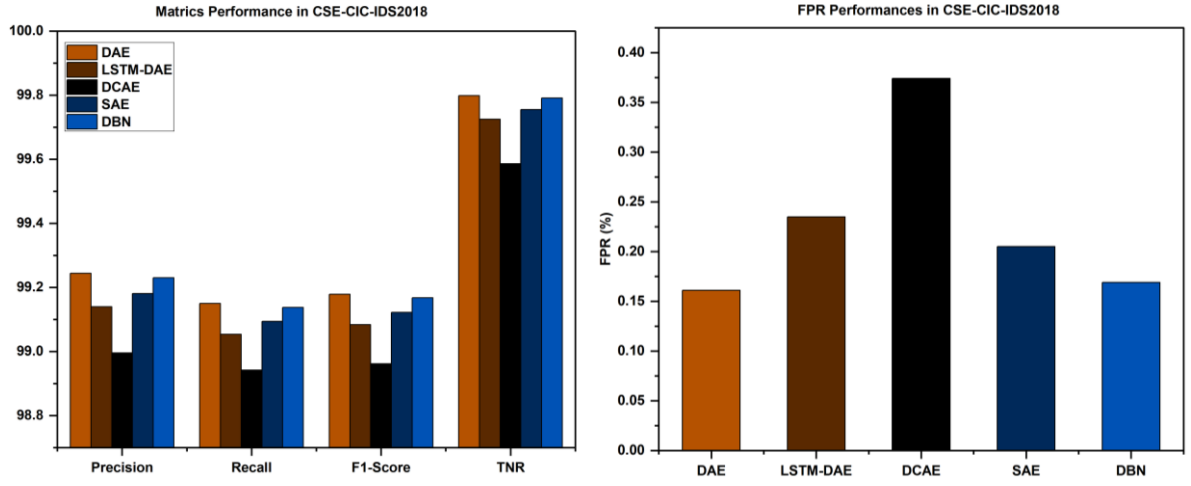


Fig. 3. Metrics comparison between different deep learning model variations on the BoT-IoT testing set.

The CSE-CIC-IDS2018 dataset is significantly improved by the three-layer model as compared with models based on HAE, HBN, SAE, or DCAE features, as shown in Figure 4(a, b). Fig. 4a illustrates how the DAE model excels over all other models, with an accuracy rate of 99.264%, a recall rate of 99.15%, an F1-score of 99.179%, and a specificity rate of 99.799%. Unlike Figure 4a, Figure 4b shows the DCAE model's poor performance, which is further accentuated by its FPR value. CSE-CIC-IDS2018 shows an over 99% success rate with transfer learning.

It appears that deep learning models can transfer learning across a wide variety of datasets to detect network intrusions based on these findings.



(a) (b)

Fig. 4. An analysis of the metrics for each variation of the deep learning model on the CSE-CIC-IDS2018 testing set.

As shown in Figure 5, each model takes different amounts of time to load using both datasets. Deep learning models are affected by their architecture, with LSTM-DAE models requiring slightly longer loading times than others. LSTM modules are more complex than alternative structures as a result of this outcome.

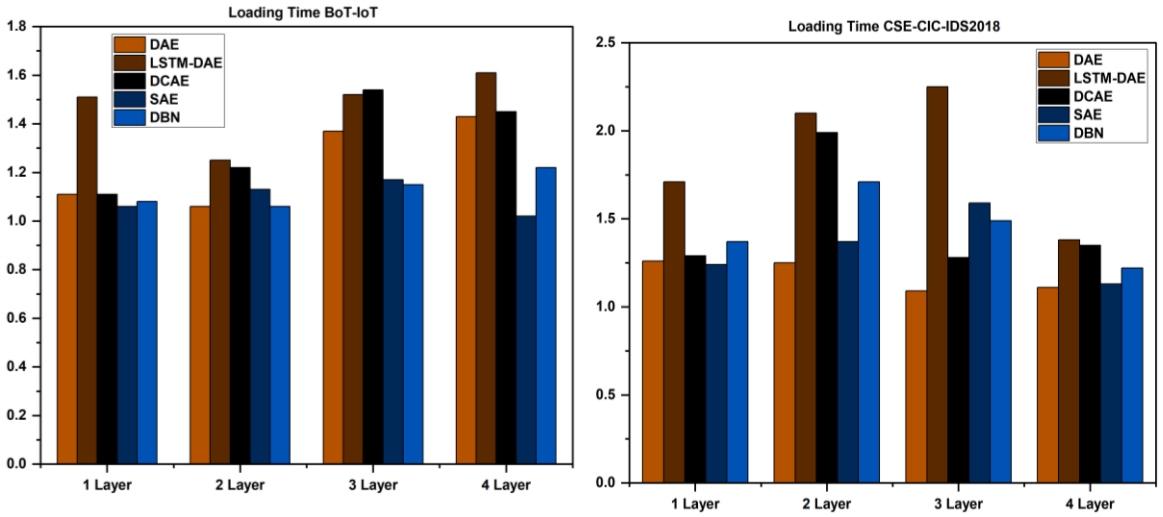


Fig. 5. Detecting attacks using both testing datasets and comparing loading times for each model.

According to Figure 6, the DCAE design has the best performance since each type of attack is identified by two convolutional layers hidden in each other. Performance falls below 90% in the theft data category, where the model shows strong detection capabilities.

A data exfiltration attack has a precision of 75%, whereas a theft-keylogging attack has a recall of 85.714 %. As a result of the limited number of theft records in this category, which comprise fewer than 100 entries out of a total of over 3.7 million, detection and precision are reduced in this category.

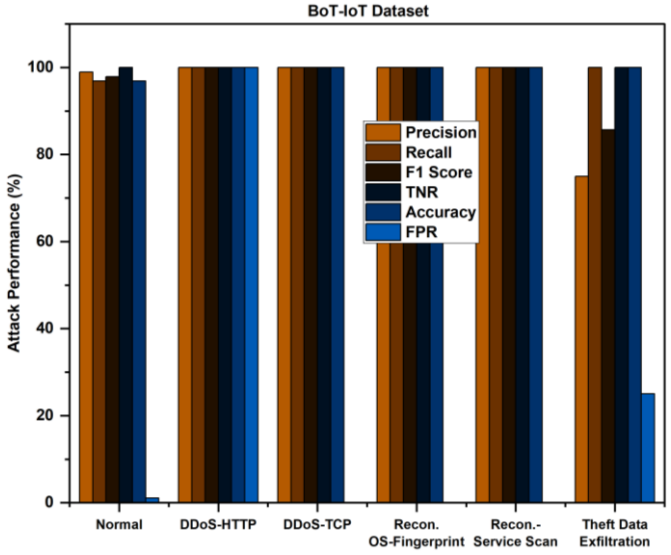


Fig. 6. A performance analysis of each class's ability to recognize attacks on the Bot-IoT dataset.

A breakdown of CSE-CIC-IDS2018 classification results is shown in Figure 7. The recall rate (or detection rate) for SQL injection attacks is 43.75 per cent, with a false detection rate of 18.98 per cent. As a result of the minimal data packet proportion of 0.004%, poor performance was observed. As opposed to the BoT-IoT dataset, the CSE-CIC-IDS2018 dataset is not up-sampled during fine-tuning. Thus, the web attack category suffers from significant class imbalances, making learning more difficult; rates are only 43.75% in this class.

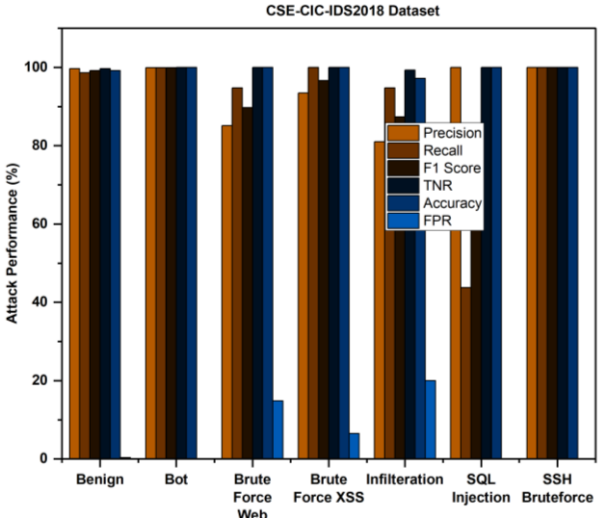


Fig. 7. CSE-CIC-IDS2018 performance on attack recognition.

Using the 5% BoT-IoT data set (57), figure 8 compares previous attack detection methods, with bolded results indicating the best results. IDSs employing deep learning for all performance indicators are outperformed by the proposed feature extraction technique. Furthermore, feature extraction and hyperparameter tuning produce deep learning models that round, are accurate, precise, and detect 100% of errors. This translates into no false positives (FPRs) for these deeper learning models. Based on these results, a proposed approach to intrusion detection systems reduces FPR values more efficiently than previous approaches.

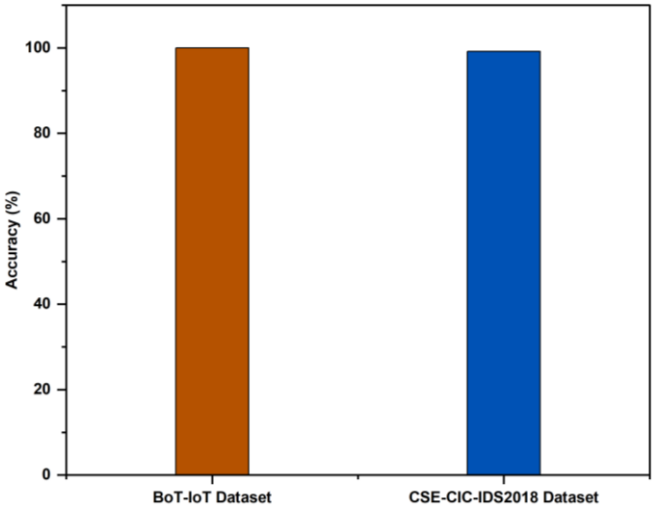


Fig. 8. Data comparison with previous BoT-IoT research.

5. CONCLUSION

Using unsupervised feature selection and deep learning techniques, this research presents an approach to intrusion detection in Industry 4.0 environments. Through the application of an unsupervised anomaly detection model that efficiently identifies malicious activity, the proposed framework significantly improves intrusion detection capabilities. This system provides increased accuracy, reduced false positives, and lower computational overhead compared to existing intrusion detection methods. As cyber threats become more complex in Industry 4.0, a scalable and adaptive approach to safeguarding critical infrastructure is essential. Developing the system's applicability to diverse industrial environments and integrating it with real-time threat response mechanisms could be a future goal.

Funding:

The authors declare that no financial aid or sponsorship was received from any external agencies or institutions for this study. All research activities were independently carried out.

Conflicts of Interest:

The authors declare no conflicts of interest.

Acknowledgment:

The authors are sincerely grateful to their institutions for their invaluable guidance and technical support.

References

[1] P. Rani and R. Sharma, "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities," *Comput. Electr. Eng.*, vol. 105, p. 108543, 2023.
[2] J. Lee, H.-A. Kao, and S. Yang, "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014, doi: 10.1016/j.procir.2014.02.001.
[3] M. Janmaijaya, A. K. Shukla, P. K. Muhuri, and A. Abraham, "Industry 4.0: Latent Dirichlet Allocation and clustering based theme identification of bibliography," *Eng. Appl. Artif. Intell.*, vol. 103, p. 104280, Aug. 2021, doi: 10.1016/j.engappai.2021.104280.
[4] M. M. H. Onik, C.-S. Kim, and J. Yang, "Personal Data Privacy Challenges of the Fourth Industrial Revolution," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, PyeongChang, Korea (South), Feb. 2019, pp. 635–638, doi: 10.23919/ICACT.2019.8701932.
[5] N. Hussain, P. Rani, H. Chouhan, and U. S. Gaur, "Cyber security and privacy of connected and automated vehicles (CAVs)-based federated learning: challenges, opportunities, and open issues," in *Federated Learning for IoT Applications*, pp. 169–183, 2022.
[6] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. MilCIS*, IEEE, 2015, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7348942/
[7] J. Yan, Y. Meng, L. Lu, and L. Li, "Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance," *IEEE Access*, vol. 5, pp. 23484–23491, 2017, doi: 10.1109/ACCESS.2017.2765544.
[8] P. Rani and R. Sharma, "Intelligent Transportation System Performance Analysis of Indoor and Outdoor Internet of Vehicle (IoV) Applications Towards 5G," *Tsinghua Sci. Technol.*, vol. 29, no. 6, pp. 1785–1795, Dec. 2024, doi: 10.26599/TST.2023.9010119.
[9] B. Susilo, A. Muis, and R. F. Sari, "Intelligent Intrusion Detection System Against Various Attacks Based on a Hybrid Deep Learning Algorithm," *Sensors*, vol. 25, no. 2, p. 580, Jan. 2025, doi: 10.3390/s25020580.
[10] P. Rani and R. Sharma, "IMFOCA-IOV: Intelligent Moth Flame Optimization based Clustering Algorithm for Internet of Vehicle," in *Proc. 14th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, IEEE, 2023, pp. 1–6.
[11] S. M. T. Nizamudeen, "Intelligent intrusion detection framework for multi-clouds – IoT environment using swarm-based deep learning classifier," *J. Cloud Comput.*, vol. 12, no. 1, p. 134, Sep. 2023, doi: 10.1186/s13677-023-00509-4.
[12] N. Moustafa, J. Hu, and J. Slay, "A holistic review of Network Anomaly Detection Systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019, doi: 10.1016/j.jnca.2018.12.006.
[13] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Machine Learning Models for Secure Data Analytics: A taxonomy and threat model," *Comput. Commun.*, vol. 153, pp. 406–440, Mar. 2020, doi: 10.1016/j.comcom.2020.02.008.
[14] W. Shang, P. Zeng, M. Wan, L. Li, and P. An, "Intrusion detection algorithm based on OCSVM in industrial control system," *Secur. Commun. Netw.*, vol. 9, no. 10, pp. 1040–1049, Jul. 2016, doi: 10.1002/sec.1398.
[15] P. Silva and M. Schukat, "On the use of k-nn in intrusion detection for industrial control systems," in *Proc. IT&T 13th Int. Conf. Inf. Technol. Telecommun.*, Dublin, Ireland, 2014, pp. 103–106. [Online]. Available: https://www.academia.edu/download/49066311/Pedro_Silva_MScSDD.pdf
[16] B. Stewart *et al.*, "A novel intrusion detection mechanism for SCADA systems which automatically adapts to network topology changes," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 4, no. 10, 2017. [Online]. Available: https://www.researchgate.net/profile/Leandros-Maglaras/publication/312320528
[17] W. Shang, J. Cui, M. Wan, P. An, and P. Zeng, "Modbus Communication Behavior Modeling and SVM Intrusion Detection Method," in *Proc. 6th Int. Conf. Commun. Netw. Secur.*, Singapore, Nov. 2016, pp. 80–85, doi: 10.1145/3017971.3017978.
[18] P. Rani, K. Ur Rehman, S. P. Yadav, and L. Hussein, "Deep Learning and AI in Behavioral Analysis for Revolutionizing Mental Healthcare," in *Demystifying the Role of NLP in Mental Health*, A. Mishra *et al.*, Eds., IGI Global, 2025, pp. 263–282, doi: 10.4018/979-8-3693-4203-9.ch014.
[19] M. Sayad Haghighi, F. Farivar, and A. Jolfaei, "A Machine Learning-based Approach to Build Zero False-Positive IPSs for Industrial IoT and CPS with a Case Study on Power Grids Security," *IEEE Trans. Ind. Appl.*, pp. 1–1, 2020, doi: 10.1109/TIA.2020.3011397.
[20] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *Int. J. Secur. Appl.*, vol. 9, no. 5, pp. 205–216, 2015.
[21] P. Rani, S. P. Yadav, P. N. Singh, and M. Almusawi, "Real-World Case Studies: Transforming Mental Healthcare With Natural Language Processing," in *Demystifying the Role of NLP in Mental Health*, A. Mishra *et al.*, Eds., IGI Global, 2025, pp. 303–324, doi: 10.4018/979-8-3693-4203-9.ch016.
[22] F. Zafar *et al.*, "A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends," *Comput. Secur.*, vol. 65, pp. 29–49, Mar. 2017, doi: 10.1016/j.cose.2016.10.006.
[23] R. Rajendran, S. V. N. S. Kumar, Y. Palanichamy, and K. Arputharaj, "Detection of DoS attacks in cloud networks using intelligent rule based classification system," *Cluster Comput.*, vol. 22, no. S1, pp. 423–434, Jan. 2019, doi: 10.1007/s10586-018-2181-4.
[24] V. Snášel, J. Nowaková, F. Xhafa, and L. Barolli, "Geometrical and topological approaches to Big Data," *Future Gener. Comput. Syst.*, vol. 67, pp. 286–296, 2017.
[25] C. Xenofontos *et al.*, "Consumer, commercial, and industrial IoT (in) security: Attack taxonomy and case studies," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 199–221, 2021.
[26] P. R. Kanna and P. Santhi, "Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features," *Knowl.-Based Syst.*, vol. 226, p. 107132, Aug. 2021, doi: 10.1016/j.knosys.2021.107132.

[27] R. M. A. Ujjan *et al.*, "Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN," *Future Gener. Comput. Syst.*, vol. 111, pp. 763–779, Oct. 2020, doi: 10.1016/j.future.2019.10.015.
[28] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time Web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020.
[29] S. N. Mighan and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *Int. J. Inf. Secur.*, vol. 20, no. 3, pp. 387–403, Jun. 2021, doi: 10.1007/s10207-020-00508-5.
[30] W. Yao, L. Hu, Y. Hou, and X. Li, "A Lightweight Intelligent Network Intrusion Detection System Using One-Class Autoencoder and Ensemble Learning for IoT," *Sensors*, vol. 23, no. 8, p. 4141, Apr. 2023, doi: 10.3390/s23084141.
[31] Q. Xu, C. Zhang, L. Zhang, and Y. Song, "The Learning Effect of Different Hidden Layers Stacked Autoencoder," in *Proc. 8th Int. Conf. Intell. Human-Machine Syst. Cybern. (IHMSC)*, Hangzhou, China, Aug. 2016, pp. 148–151, doi: 10.1109/IHMSC.2016.280.
[32] S. Wei, Z. Zhang, S. Li, and P. Jiang, "Calibrating Network Traffic with One-Dimensional Convolutional Neural Network with Autoencoder and Independent Recurrent Neural Network for Mobile Malware Detection," *Secur. Commun. Netw.*, vol. 2021, pp. 1–10, Feb. 2021, doi: 10.1155/2021/6695858.