The Journal of Supercomputing (2024) 80:1059–1079
https://doi.org/10.1007/s11227-023-05511-w

Comparative analysis of dimensionality reduction techniques for cybersecurity in the SWaT dataset

Mehmet Bozdal[1] · Kadir Ileri[2] · Ali Ozkahraman[3]

Accepted: 18 June 2023 / Published online: 8 July 2023

Abstract

The Internet of Things (IoT) has revolutionized the functionality and efficiency of distributed cyber-physical systems, such as city-wide water treatment systems. However, the increased connectivity also exposes these systems to cybersecurity threats. This research presents a novel approach for securing the Secure Water Treatment (SWaT) dataset using a 1D Convolutional Neural Network (CNN) model enhanced with a Gated Recurrent Unit (GRU). The proposed method outperforms existing methods by achieving 99.68% accuracy and an F1 score of 98.69%. Additionally, the paper explores dimensionality reduction methods, including Autoencoders, Generalized Eigenvalue Decomposition (GED), and Principal Component Analysis (PCA). The research findings highlight the importance of balancing dimensionality reduction with the need for accurate intrusion detection. It is found that PCA provided better performance compared to the other techniques, as reducing the input dimension by 90.2% resulted in only a 2.8% and 2.6% decrease in the accuracy and F1 score, respectively. This study contributes to the field by addressing the critical need for robust cybersecurity measures in IoT-enabled water treatment systems, while also considering the practical trade-off between dimensionality reduction and intrusion detection accuracy.

Keywords Intrusion detection · Secure water treatment dataset · Convolutional neural networks · Dimensionality reduction · Gated recurrent unit

✉ Kadir Ileri
kileri@bandirma.edu.tr

[1] Electrical and Electronics Engineering Department, Abdullah Gul University, Kayseri, Turkey

[2] Electrical and Electronics Engineering Department, Bandirma Onyedi Eylul University, Balikesir, Turkey

[3] Electronics and Communication Engineering Department, Istanbul Technical University, Istanbul, Turkey

1 Introduction

The Internet of Things (IoT) is a technology that enables the connection of everyday devices, such as appliances, vehicles, and industrial equipment, to the internet. This allows these devices to communicate with one another and with other systems, and to be controlled and monitored remotely. The increased connectivity provided by IoT has had a significant impact on industrial control systems, which were previously closed off from the outside world. In the past, industrial control systems (ICS) were primarily used to control and monitor industrial processes within a single facility or on a small scale. With the advent of IoT, however, these systems can now be connected to the internet, enabling remote monitoring and control. This allows for city-wide or nationwide distributed systems to work collaboratively and efficiently, with the ability to share information and coordinate actions across different locations. Although connectivity has many benefits, it also brings the danger of cyberattacks. An attacker can access the communication channel and control the system and implement an attack. The attack may have various effects from simply unavailability of service to catastrophic system failure. As industrial control systems become connected to the internet, they become more vulnerable to cyberattacks.

There are examples of cyberattacks targeting industrial control systems (ICS) in recent years. In 2000, a former employee maliciously commanded SCADA (Supervisory Control and Data Acquisition) radio-controlled sewage [1]. He caused hundreds of thousands of raw sewerages to spill out around various parts of the city in Australia. One of the most well-known examples of an ICS cyberattack is the Stuxnet worm [2], which was discovered in 2010. The worm specifically targeted the software used to control industrial processes at an Iranian nuclear facility. The attack caused physical damage to the centrifuges used to enrich uranium, setting back the facility's operations. In 2015 a malicious cyberattack targeted the Ukraine power grid [3], causing widespread power outages across the country. The attackers used spear-phishing emails to gain access to the network and then used malware to disrupt the operations of the power plants.

WannaCry is a ransomware computer worm that employs the RSA and AES encryption algorithms to encrypt files on the victim's computer. This notorious ransomware attack had a widespread impact, affecting thousands of computers, including industrial control plants [4]. The worm gains access to a computer by exploiting a vulnerability in the Server Message Block (SMB) protocol, which enables remote code execution [5]. Triton malware, which specifically targeted the industrial control systems used to operate critical infrastructure, was discovered in 2017 [6]. The malware manipulated the Triconex Safety Instrumented System (SIS) controllers, which are used to monitor and control industrial processes in facilities such as oil refineries and chemical plants. Although the full extent of damage caused by these attacks is not publicized, they resulted in significant outages in a petrochemical plant, posing the potential risk of chemical releases [7]. Attacks on ICS can range from simple disruption of service to catastrophic failures that can have major physical consequences. Given the potential consequences of successful attacks, it is important to take the necessary steps to protect industrial control systems, especially critical infrastructure. Therefore, organizations that deploy IoT-enabled industrial control systems need to be aware of these security risks and take appropriate measures to protect against them. This includes implementing robust security protocols, monitoring for and responding to potential security threats, and providing employee education and training to raise awareness of security risks. One common practice of protecting ICS is the use of an intrusion detection system (IDS). Researchers have proposed various IDSs to identify and detect intrusions and secure cyber-physical systems. However, the efficiency and effectiveness of IDSs can be improved through feature selection and feature reduction algorithms.

In this research, we propose a method for securing the Secure Water Treatment (SWaT) dataset by implementing an IDS using a one-dimensional Convolutional Neural Network (CNN) model enhanced with a Gated Recurrent Unit (GRU). Additionally, we explore various dimensionality reduction techniques, such as autoencoders, Generalized Eigenvalue Decomposition (GED), and Principal Component Analysis (PCA). The goal of the paper is to determine the optimal feature subset that can improve the efficiency of the model without compromising its accuracy.

In light of the above, the contributions of the paper are as follows:

- A novel IDS approach based on a 1D CNN model enhanced with a GRU for securing the SWaT dataset, which outperforms traditional IDS methods in terms of accuracy and robustness.
- Evaluates the impact of dimensionality reduction techniques.
- Provides insights into the trade-off between feature reduction and detection accuracy and demonstrates the importance of balancing these two factors for effective intrusion detection in cyber-physical systems.

The rest of this paper is organized as follows. Section 2 provides a brief background on the SWaT dataset and CNN. Section 3 presents the proposed method and implementation details. Section 4 includes experimental results and further discussion on the timing analysis of the approach, as well as an exploration of its limitations. Section 5 consists of a conclusion summarizing the findings and potential future research directions.

1.1 Related work

Rule-based anomaly detection is a widely used method for identifying unusual activity in a system based on predefined rules. These rules can be based on patterns and characteristics of known malicious activity, and if a known pattern is observed, it is considered an anomaly. The rules can also be based on the normal behavior of the system, such as setting threshold values for specific parameters. If these values are exceeded or not met, an alarm is triggered.

Adepu and Mathur [8] proposed a novel method for distributed attack detection by utilizing process invariants derived from Piping and Instrumentation Diagrams (P &IDs) based on physical properties of the system. The authors applied this method to a SWaT system, which has chemical processes as well, but due to the nonlinearity of these processes, only physical invariants were used. Although the method does not produce any false alarms, it fails to identify some attack types like denial of service. Furthermore, the process of deriving the invariants is currently a manual process, which may limit the scalability of the method. Future research should focus on automating this process to improve the overall performance of the method.

Another example of rule-based anomaly detection is Logical Analysis of Data (LAD) which was implemented by Das et al. [9]. This method allows for near-real-time processing with low computational power, making it an efficient and cost-effective way to detect some types of cyberattacks. However, it is important to note that reliance on predefined rules alone can be circumvented [10], highlighting the need to supplement rule-based methods with other security measures such as behavioral analysis and machine learning. Al-Dhaheri et. al [11] proposed hybrid intrusion detection system. Rule-based IDS that checks limits and safety values, model-based monitoring that implements physical model, and data-driven approach for nonlinear modeling.

Aboah et al. [12] proposed a neural network with a one-class objective function (NN-One-class) which improves the detection performance compared to some of the previous methods. However, the training time can be quite extensive, taking up to 110 min with an NVIDIA Tesla T4 GPU and a RAM of 32GB. Given the complexity of the data, this represents a significant amount of resources.

Kravchik and Shabtai [13] proposed a 1D Convolutional Neural Network (CNN) to identify cyberattacks on the SWaT dataset. They implemented dedicated anomaly detectors for each stage of the SWaT system to improve the performance. The results showed that independent analysis of each stage outperforms a single model for the whole system. However, as the stages of the SWaT system are dependent on each other, it is important to also investigate the inter-stage dependencies in order to further improve the performance of the detection system.

Xie et al. [14] investigated Stacked Denoising Autoencoders (SDA) with 1D Convolutional Neural Networks and Gated Recurrent Units (GRUs) to leverage correlations and dependencies between variables. The approach achieved successful detection of various attack types but encountered training challenges due to the large number of model parameters.

Goh et al. [15] proposed an unsupervised learning approach that utilizes a Long Short-Term Memory (LSTM) and Cumulative Sum (CUSUM) for anomaly detection. The method goes beyond traditional anomaly detection by specifically identifying the sensor that was targeted in the attack. However, it is important to note that this approach was only deployed in the first stage of the SWaT system, which consists of six stages.

Zhou et. al. [16] suggested to use temporal and spatial correlation as temporal correlation alone is not beneficial for high dimensional data. They have implemented Graph Attention Network (GAT) with Multihead Dynamic Attention (MDA). The implementation leverages of relationship between various sensors thanks to MDA.

Nedeljkovic and Jakovljevic [17] implemented semi-supervised IDS by using CNN-based auto regression. They applied Finite Impulse Response (FIR) filter to remove high frequency noise.

Dillon et. al [18] showed that design knowledge increases the efficiency of the IDS. One reason behind that is when data consist of binary values and analog ones, machine learning algorithms can be biased toward binary ones and ignore them. Experimental results show a 5% increase in the detection by using design knowledge.

There are also research papers that implement dimension reduction. Li et. al. [19] proposed a method called "end-to-end anomaly detection" for detecting anomalies using a digital twin. The proposed method uses a multidimensional deconvolutional network and attention mechanism with PCA to detect anomalies quickly in real-time. However, the performance of the method, F1=0.94, is not acceptable for critical infrastructure. Alimi et al. [20] applied PCA to various supervised learning algorithms. They achieved the best performance for the SWaT dataset with the J48 decision tree classifier; however, the F1 score was 0.814. Priyanga et al. [21] proposed a hyper-graph-based anomaly detection technique. The proposed algorithm involves two phases: dimensionality reduction using enhanced principal component analysis (EPCA) and anomaly detection with HG-based convolution neural network (CNN). El-Nour et al. proposed framework [22] involving two isolation forest models and PCA. In another study, Yazdinejad et al. [23] applied LSTM to get benefit of long-term dependancy of SWaT data and autoencoders to reduce number of features, achieving an accuracy of 96.3%. Although these methods implemented dimensionality reduction algorithms, they do not emphasize on dimensionality reduction. This article explores the limitations of current dimensionality reduction methods and discusses the importance of dimensionality reduction.

2 Background

2.1 Secure water treatment (SWaT) dataset

The Secure Water Treatment (SWaT) dataset [24], which is widely used as a testbed for water treatment, is used in this experiment. The SWaT system produces filtered water at a rate of 5 gallons per hour and was designed under the supervision of Singapore's Public Utility Board. It contains six stages labeled as P1 through P6 as shown in Fig. 1. Each stage is operated by a PLC (e.g., PLC 1 controls stage P1) using a distributed control strategy.

The system can be divided into two levels: Level 0 and Level 1. At Level 0, Programmable Logic Controllers (PLCs) acquire data from local sensors such as an acidity analyzer, water level sensor, and flow meter, and handle the actuators such as valves and pumps. At Level 1, the PLCs communicate with each other using a separate network, which connects all six stages to the Supervisory Control and Data Acquisition (SCADA) system.

PLC 1 controls the flow of raw water by opening or closing the valves connected to the inlet and outlet of the raw water tank in Stage 1. After chemical dosing in

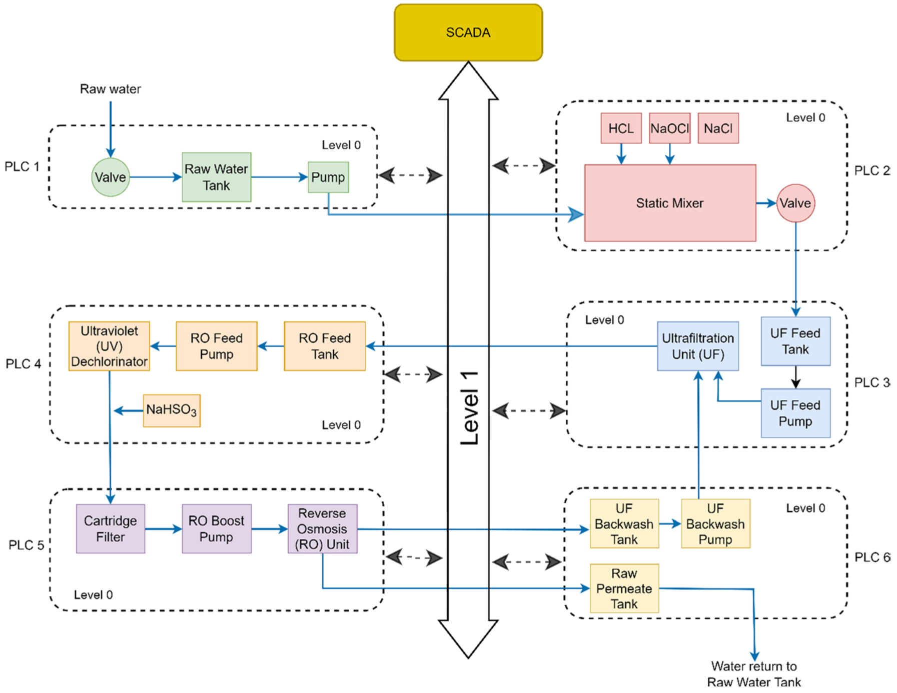


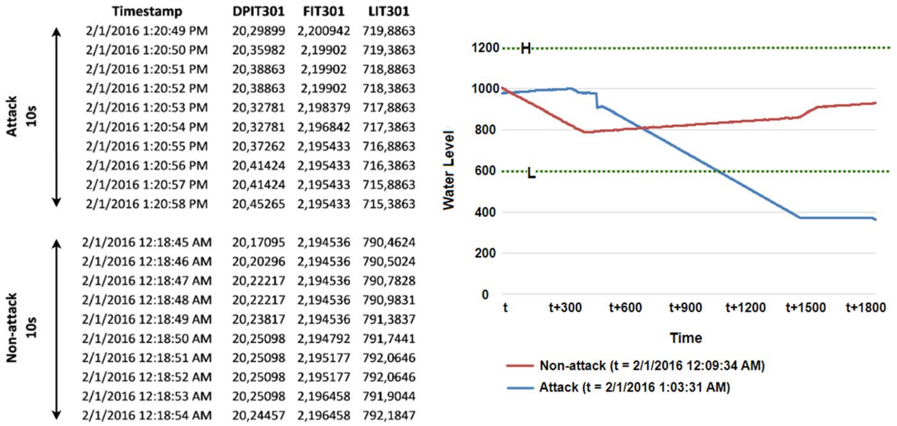
Fig. 1 Process flowchart of SWaT system

Stage 2, the water is fed to Stage 3 for the Ultra Filtration (UF). From there, the UF feed pump forwards the water to the Reverse Osmosis (RO) feed tank in Stage 4. Before entering the RO process, the water passes through an ultraviolet (UV) dechlorinator to remove any free chlorine. In Stage 5, the RO process removes inorganic impurities from the de-chlorinated water. The filtered water produced by the RO process is stored in the permeating tank in Stage 6 for distribution, and Stage 6 also handles the cleaning of the UF membranes through the backwash process.

The dataset comprises a total duration of 11 days, with the initial seven days being free from any attacks. It consists of 946,722 samples, each containing 51 attributes. Figure 2a presents the raw data, displaying three features to ensure clarity of presentation.

Various methods can be employed to carry out attacks, including physical access to the system, unauthorized network access to the SCADA infrastructure, and the installation of malicious firmware on the Programmable Logic Controllers (PLCs). The attacks on the SWaT dataset were implemented at Level 1, where the PLCs communicate with the SCADA system. At this level, data packets are manipulated, and malicious messages are transmitted to the SCADA system.

In Fig. 2b, a specific attack scenario is illustrated, involving tampering with the water level (LIT301) in a water tank. The normal behavior dictates that the water level should remain between the predefined Low (L) and High (H) levels. However, in this attack scenario, the water level gradually decreases at a rate of 0.5 mm per



	Timestamp	DPIT301	FIT301	LIT301
Attack 10s	2/1/2016 1:20:49 PM	20,29899	2,200942	719,8863
	2/1/2016 1:20:50 PM	20,35982	2,19902	719,3863
	2/1/2016 1:20:51 PM	20,38863	2,19902	718,8863
	2/1/2016 1:20:52 PM	20,38863	2,19902	718,3863
	2/1/2016 1:20:53 PM	20,32781	2,198379	717,8863
	2/1/2016 1:20:54 PM	20,32781	2,196842	717,3863
	2/1/2016 1:20:55 PM	20,37262	2,195433	716,8863
	2/1/2016 1:20:56 PM	20,41424	2,195433	716,3863
	2/1/2016 1:20:57 PM	20,41424	2,195433	715,8863
	2/1/2016 1:20:58 PM	20,45265	2,195433	715,3863
Non-attack 10s	2/1/2016 12:18:45 AM	20,17095	2,194536	790,4624
	2/1/2016 12:18:46 AM	20,20296	2,194536	790,5024
	2/1/2016 12:18:47 AM	20,22217	2,194536	790,7828
	2/1/2016 12:18:48 AM	20,22217	2,194536	790,9831
	2/1/2016 12:18:49 AM	20,23817	2,194536	791,3837
	2/1/2016 12:18:50 AM	20,25098	2,194792	791,7441
	2/1/2016 12:18:51 AM	20,25098	2,195177	792,0646
	2/1/2016 12:18:52 AM	20,25098	2,195177	792,0646
	2/1/2016 12:18:53 AM	20,25098	2,196458	791,9044
	2/1/2016 12:18:54 AM	20,24457	2,196458	792,1847

Fig. 2 **a** SWaT raw data including non-attack and attack (tampering water level – LIT301) along with **b** visual representation of data

second. As a consequence, after a certain period of time, the water level drops below the Low level, potentially causing damage to the system.

2.2 Convolutional neural network and gated recurrent unit

Feature extraction is a crucial step in data classification, and it can be done manually or automatically. Deep learning systems, particularly Convolutional Neural Networks (CNNs), are widely used as automatic feature extraction methods and often outperform manual feature extraction techniques. CNNs excel at capturing local information due to their convolutional and pooling operations. The convolution operation where filters are convolved with the input data to produce feature maps can be defined as follows:

$$G[n] = (f * h)[n] = \sum_{k=-\infty}^{\infty} h[k] * f[n-k] \tag{1}$$

where G[n] represents the output of convolution operation (feature maps), f and h are the input data and the convolutional filter (kernel), respectively.

Equation 1 calculates the weighted sum of the convolution between the filter h and the input f, where the filter is shifted across the input data by the index k.

After the convolutional operations, the feature maps obtained in CNNs are passed through nonlinear activation functions like Rectified Linear Unit (ReLU) to introduce nonlinearity into the network. Pooling layers, such as max pooling or average pooling, are commonly used to reduce the spatial dimensions of the feature maps while retaining important features.

To capture more complex and abstract representations, CNNs stack multiple convolutional and pooling layers. As the network deepens, it learns increasingly complex features. Toward the final layers, fully connected layers are typically employed to map these high-level features to specific output classes or predictions, enabling the network to make accurate classifications or predictions based on the learned representations.

However, when dealing with sequential data such as the SWaT dataset, CNNs alone may not effectively capture temporal dependencies. Recurrent Neural Networks (RNNs) are better suited for modeling sequences due to their ability to retain information over time. Specifically, Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) are commonly used RNN variants.

The SWaT dataset contains time-series data representing the state of an industrial control system at different time points. To analyze this sequential data, various neural network architectures can be employed. While a 1D Convolutional Neural Network (CNN) is commonly used to identify patterns and anomalies, it is worth noting that Recurrent Neural Networks (RNNs) are better suited for capturing temporal dependencies inherent in time-series data.

GRU and LSTM networks are two types of RNN variants specifically designed to address the vanishing gradient problem. While both models address this issue and capture long-term dependencies, there are key differences. GRU has a simpler architecture, combining the forget and input gates of LSTM into a single update gate and removing the output gate. This simplification reduces computational complexity and training difficulty. The equations governing the GRU update and reset gates are as follows:

$$\begin{aligned} r_t &= \sigma(x_t * U_r + H_{t-1} * W_t) \\ u_t &= \sigma(x_t * U_u + H_{t-1} * W_u) \end{aligned} \tag{2}$$

where x_t is the input at time step t and H_t is the hidden state at time step t. U_r, W_r, U_u, and W_u are the weight matrices specific to the reset and update gates and σ is the sigmoid activation function.

r_t is the reset gate that determines the extent to which the previous hidden state should be forgotten or reset. It influences the amount of information retained from the past by considering both the current input (x_t) and the previous hidden state (H_{t-1}).

U_t is an update gate that controls the flow of information from the previous hidden state (H_{t-1}) to the current hidden state (H_t). It determines the degree to which the previous hidden state should be combined with the candidate activation (generated from the current input) to update the current hidden state.

Indeed, the equations for determining the reset gate (r_t) and update gate (u_t) in the GRU architecture have a similar structure. The weight matrices (W_r, W_u, U_r, and U_u) differentiate the equations.

By leveraging the strengths of both CNNs and RNNs, a hybrid model can effectively capture local patterns and long-term dependencies, leading to improved anomaly detection, reliability, and security in industrial control systems.

2.3 Dimensionality reduction techniques

In machine learning, dimensionality reduction is a common technique used to reduce the number of features in a dataset. Feature reduction techniques can help to reduce the complexity of a dataset, remove noise, and improve the efficiency of machine learning algorithms. We have explored the most common dimensionality reduction techniques including Principal Component Analysis (PCA), Generalized Eigenvalue Decomposition (GED), and Autoencoders.

2.3.1 Principal component analysis (PCA)

Principal Component Analysis (PCA) is a commonly used statistical technique for dimensionality reduction in data analysis and machine learning. The main goal of PCA is to identify patterns and structure in high-dimensional data by reducing the number of variables and retaining the most important information.

Once the principal components are identified, data can be projected onto a lower-dimensional subspace by selecting a subset of the principal components. This new subspace retains the most important information from the original high-dimensional dataset while reducing the number of variables.

2.3.2 Generalized eigenvalue decomposition (GED)

Generalized Eigenvalue Decomposition (GED) is a dimension reduction technique that is used to reduce the dimensionality of high-dimensional data while preserving the information contained in the original data. GED finds a linear transformation of the original data that maximizes the ratio of between-class variance to within-class variance.

2.3.3 Autoencoders

An autoencoder is a neural network that can be used for dimensionality reduction by compressing high-dimensional data into a lower-dimensional latent representation as shown in Fig. 3. The autoencoder consists of an encoder that maps the input data to the latent layer, a bottleneck layer that represents the compressed data, and a decoder that reconstructs the original data from the latent representation. By training the network to minimize the difference between the input and reconstructed output, the network learns to identify the most important features in the data and discard the less important ones.

The size of the latent space is an important consideration when designing an autoencoder, as it determines how much information will be retained after compression. If the latent space is too small, the autoencoder may lose important information and result in the poor reconstruction of the input data. On the other

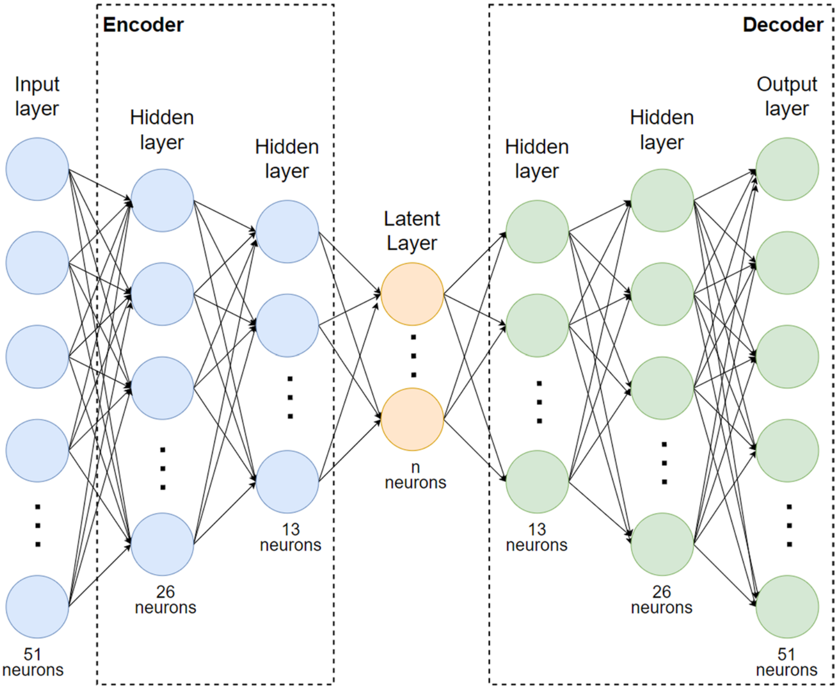


Fig. 3 Architecture of the autoencoder used in this experiment

hand, if the latent space is too large, the autoencoder may overfit and memorize the training data, resulting in poor generalization to new data.

3 Proposed method and experiment

3.1 Proposed method

In this research, a novel approach has been employed to enhance the performance of the One-dimensional Convolutional Neural Network (1D CNN) combined with GRU. The proposed approach aims to enhance performance by utilizing the strengths of both 1D CNN and GRU as presented in Fig. 4. This integration allows for the learning of spatial and temporal features of input data, facilitating the capturing of complex patterns in time-series data.

In the proposed 1D CNN model, we introduced three convolutional layers with different kernel lengths (2, 3, and 5) operating in parallel, as depicted in Fig. 4a. This design allows us to extract diverse features from the data, as each kernel size has the potential to capture distinct patterns. By limiting the features to a length of 5, we specifically chose kernel sizes that are smaller or equal to 5 to ensure the extraction of local features while considering the reduced feature space. The resulting feature maps obtained from each convolutional layer are concatenated to form a unified

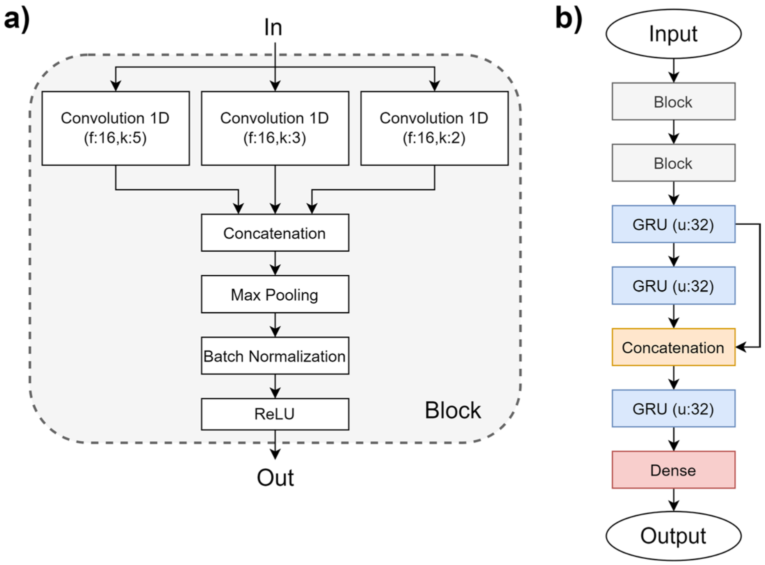


Fig. 4 **a** Architecture 1D CNN block used in the experiment along with **b** the whole architecture including GRU layers

layer. This merging of feature maps serves the purpose of combining the learned features from various levels of abstraction into a cohesive representation. Therefore, the model can leverage both local and global features, enhancing its ability to extract meaningful patterns and representations from the input data.

The concatenated layer is then fed into a max pooling layer, which reduces the spatial dimensions of the tensor by taking the maximum value within a specified window. This helps to extract the most salient features from the input sequence while reducing the computational cost of the network.

Batch normalization is then applied to normalize the output of the previous layer, which helps to speed up training and improve the generalization of the model. Finally, the ReLU activation function is applied elementwise to the output of the batch normalization layer, which introduces nonlinearity to the model and helps to extract more complex features.

As shown in Fig. 4b, after the two blocks of the CNN, the output is fed into GRU layers, which can capture longer-term dependencies in the input sequence. To tackle the vanishing gradients problem, the output of the GRU layer is then passed through a dense layer, facilitating more effective gradient propagation throughout the network.

3.2 Parameter selection and experimental setup

In order to achieve optimal performance of the proposed method, it is important to carefully tune its hyperparameters. Hyperparameters are settings that are

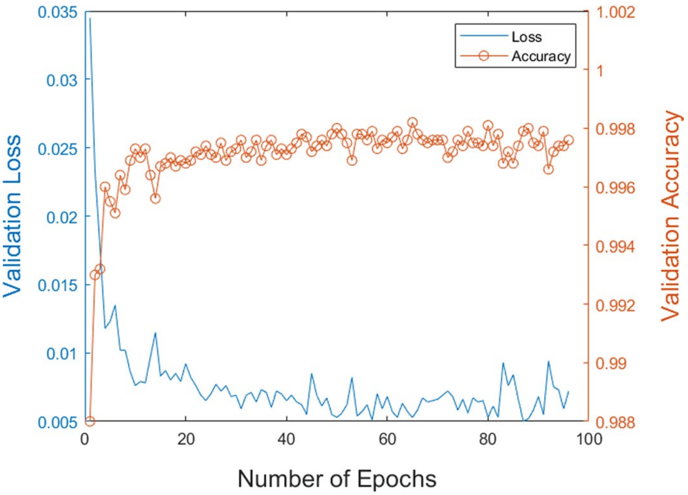


Fig. 5 Determining the optimal number of epochs for model training using validation loss and accuracy

not learned during training but are set before training and can have a significant impact on the performance of the model.

The number of epochs is an important hyperparameter that determines the number of times the entire dataset is used to train the model. In this research, we conducted an epoch analysis to determine the optimal number of epochs for training the 1D CNN-GRU model on the SWaT dataset. We trained the model for different numbers of epochs ranging from 1 to 100 and evaluated its performance.

Figure 5 illustrates the trends of validation loss and accuracy as a function of the number of epochs. The optimal epoch number is located at epoch number 10, where the minimum validation loss and maximum validation accuracy intersect.

Other hyperparameters such as batch size, learning rate, and optimizer can significantly impact the performance of a deep-learning model. Therefore, it is important to optimize these hyperparameters to achieve the best possible performance. In this study, we selected a batch size of 32, a learning rate of 0.001, and the Adam optimizer based on their effectiveness in previous studies and our experimentation on the SWaT dataset.

4 Results and discussion

The proposed method is compared with the state-of-the-art techniques on the SWaT dataset to demonstrate its effectiveness. Additionally, the results of our dimensionality reduction analysis using PCA, GED, and autoencoders are presented to show the impact of feature reduction on the performance of the intrusion detection system.

4.1 Evaluation method

The proper testing of an Intrusion Detection System (IDS) is a crucial step in evaluating its effectiveness. To ensure the accuracy and reliability of the proposed IDS model, we conducted a comprehensive analysis of its performance.

Our proposed model employs a binary classifier to differentiate between authentic messages and potential attacks. As a result, there are four possible outcomes: false negative (FN), false positive (FP), true negative (TN), and true positive (TP). A true positive occurs when an attack is correctly identified by the system, while a true negative occurs when an authentic message is correctly accepted as such. In contrast, a false positive occurs when an authentic message is labeled as an attack, and a false negative occurs when an attack is labeled as an authentic message.

To assess the performance of our proposed IDS model, we calculated the values for FN, FP, TN, and TP. These values provide important insights into the system's accuracy and effectiveness in detecting potential attacks. Additionally, we calculated several key metrics such as accuracy, precision, and recall values.

The accuracy, Eq. (3), metric evaluates the percentage of correct predictions made by the model, whereas the precision metric assesses the percentage of true positives among all positive predictions. Recall metric evaluates the percentage of true positives detected by the system among all actual attacks. By considering all of these metrics, we can assess the overall performance of the IDS model and determine its efficiency in detecting potential attacks.

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \tag{3}$$

Precision, Eq. (4), is a performance metric used in evaluating the effectiveness of an Intrusion Detection System (IDS). Specifically, precision evaluates the percentage of true positive predictions made by the system out of all positive predictions. It provides an important measure of the system's ability to accurately identify potential attacks while minimizing the number of false positives.

$$\text{Precision} = TP/(TP + FN) \tag{4}$$

Recall (also known as sensitivity or detection rate), Eq. (5), is a performance metric used in evaluating the effectiveness of an IDS. Specifically, recall measures the percentage of true positive predictions made by the system out of all actual positive cases.

$$\text{Sensitivity(Recall)} = TP/(TP + FN) \tag{5}$$

The F1 score, Eq. (6), is defined as the harmonic mean of recall and precision, where a higher score indicates better performance. By taking the harmonic mean,

Table 1 Comparison of methods that use the SWaT dataset

Reference	Accuracy	F1	Precision	Recall
CNN-GRU-SDA [14]	–	0.91	0.99	0.85
CNN-FIR [17]	97.846	0.902	0.988	0.830
1D CNN [13]	97.195	0.871	0.968	0.791
NN-PCA [25]	97.408	0.885	0.911	0.860
Monitoring System [11]	–	0.925	**1**	0.861
STAE-AD [26]	–	0.880	0.960	0.815
NN-one class [12]	–	0.870	0.940	0.820
EPCA-HG-CNN [21]	98.02	0.9805	0.9771	0.9839
Digital-twin [19]	–	90.59	0.923	0.961
DIF [22]	97.375	0.882	0.935	0.835
1D CNN-GRU (This Paper)	**0.9968**	**0.9869**	0.9855	**0.9882**

Best results are bold

the F1 score places more emphasis on the lower of the two metrics, meaning that a model with high precision but low recall (or vice versa) will have a lower F1 score than a model with both high recall and high precision.

$$F1 = (2 * (\text{Precision} * \text{Recall}))/(\text{Precision} + \text{Recall}) \tag{6}$$

4.2 Comparison with the state-of-the-art techniques

As many researchers use this dataset, it serves as a common benchmark for evaluating and comparing the performance of different methods. The proposed method is compared with other state-of-the-art methods.

Table 1 presents the performance metrics for the proposed method along with other state-of-the-art proposals. There is a trade-off between Precision and Recall. These two metrics measure different aspects of a classifier's performance, and optimizing one metric often comes at the expense of the other.

Table 1 depicts that some models achieved high precision scores but lower recall scores (e.g., CNN–FIR), while others achieved higher recall scores but lower precision scores (e.g., EPCA-HG-CNN). The proposed CNN-GRU model achieved the best overall performance, achieving an impressive accuracy score of 0.9968, F1 score of 0.9869, precision of 0.9855, and recall of 0.9882.

4.3 Dimensionality reduction

Dimensionality reduction is a commonly used technique in machine learning for reducing the number of features in a dataset. It helps in reducing the complexity of the dataset, removes noise, and improves efficiency. In this research, we explored the effectiveness of three commonly used dimensionality reduction techniques:

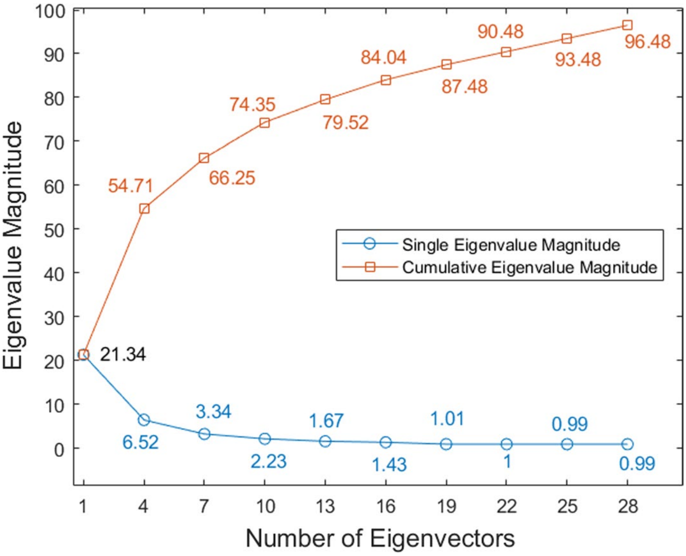


Fig. 6 Magnitudes of eigenvalues for eigenvectors

Table 2 Performance analysis of generalized eigenvalue decomposition

# of eigen-vectors	Accuracy	F1	Precision	Recall	TP	TN	FP	FN
5	0.9692	0.8681	0.8993	0.8391	9122	78092	1021	1749
10	0.9927	0.9700	0.9687	0.9713	10559	78772	341	312
15	0.9947	0.9781	0.9834	0.9727	10575	78935	178	296
20	0.9949	0.9789	0.9843	0.9735	10583	78945	168	288
25	0.9950	0.9793	0.9805	0.9781	10633	78901	212	238

Generalized Eigenvalue Decomposition (GED), Autoencoders, and Principal Component Analysis (PCA) for improving the performance of the proposed IDS.

4.3.1 Generalized eigenvalue decomposition

The magnitude of the eigenvalues obtained through GED can provide important information about the quality of the dimensionality reduction. Figure 6 presents the Magnitudes of eigenvalues for eigenvectors.

Table 2 presents experimental results for GED. The accuracy increases from 0.9692 for 5 eigenvectors to 0.9950 for 25 eigenvectors. Similarly, the F1 score consistently increases from 0.8681 to 0.9793. The precision of the IDS also increases as the number of eigenvectors increases, with the highest precision of 0.9843 achieved with 20 eigenvectors. The recall of the IDS is highest for 25 eigenvectors with a value of 0.9781, indicating that the IDS with 25 eigenvectors is better at detecting

Table 3 Performance analysis of autoencoder

# of latent layer (n)	Accuracy	F1	Precision	Recall	TP	TN	FP	FN
5	0.9849	0.9479	0.9868	0.8864	9637	78984	129	1234
10	0.9932	0.9714	0.9918	0.9518	10347	79028	85	524
15	0.9926	0.9685	0.9926	0.9455	10279	79036	77	592
20	0.9921	0.9665	0.9887	0.9452	10275	78996	117	596
25	0.9965	0.9855	0.9823	0.9881	10742	78925	188	129

true positive cases. The true positive (TP) values increase with the number of eigenvectors, while the false negative (FN) values decrease, indicating that the IDS is more capable of detecting true positive cases with a higher number of eigenvectors. However, the false positive (FP) values slightly increase as the number of eigenvectors increases, which suggests that increasing the number of eigenvectors may result in a higher rate of false alarms.

4.3.2 Autoencoder

Choosing the appropriate number of latent layers can be a challenging task, and it often requires experimentation and tuning to find the optimal number for a given problem. Typically, the number of latent layers is determined by balancing the tradeoff between model complexity and performance on the validation set.

Table 3 presents the performance analysis of an autoencoder with different numbers of latent layers (n). The accuracy of the model increases with the number of latent layers, reaching its highest value of 0.9965 with 25 latent layers. Similarly, the F1 score, precision, and recall increase with the number of latent layers, with the highest values being 0.9855, 0.9823, and 0.9881, respectively, for 25 latent layers.

4.3.3 Principal component analysis

PCA aims to retain the most important information from the original high-dimensional dataset while reducing the number of variables. The number of principal components that should be retained depends on the amount of variance they explain. Figure 7 shows the variance of each principal component and accumulated one. It is observed that the first principal component explains the most variance, followed by the second and third principal components. As more principal components are added, the amount of explained variance gradually decreases. In this specific case, it seems that retaining the first 5 principal components can capture a significant amount of the variation in the data, as they explain over 99.5% of the variance.

Table 4 depicts the performance analysis of the proposed method using PCA for different numbers of components. The result shows that the performance of the intrusion detection model does not degrade significantly even when the number of principal components is reduced by 90.2%. Specifically, when the number of principal components is reduced to 5, the model achieves an accuracy of 0.9909 and an F1

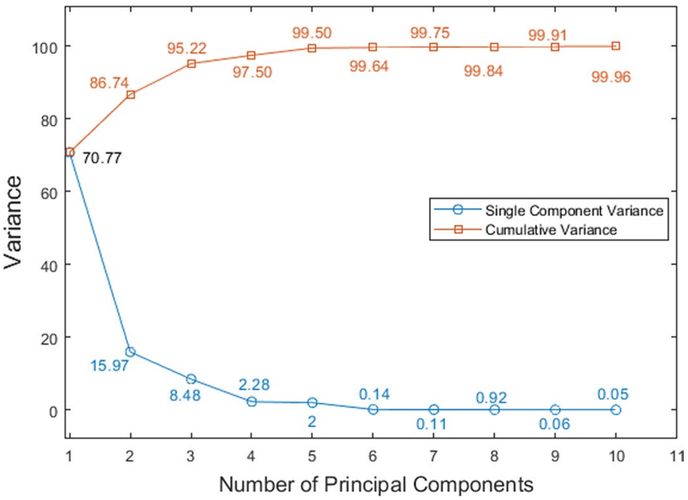


Fig. 7 Variance of PCA components

Table 4 Performance analysis of PCA

# of component	Accuracy	F1	Precision	Recall	TP	TN	FP	FN
5	0.9909	0.9613	0.9892	0.9351	10165	79002	111	706
10	0.9967	0.9863	0.9857	0.9869	10729	78957	156	142
15	0.9967	0.9862	0.9839	0.9886	10747	78937	176	124
20	0.9969	0.9873	0.9832	0.9915	10778	78929	184	93
25	0.9961	0.9839	0.9781	0.9898	10760	78873	240	111

score of 0.9613. When the number of principal components is increased to 20, the model achieves an accuracy of 0.9969 and an F1 score of 0.9873.

The analysis reveals an interesting trend regarding the trade-off between true positive and false negative values. As the number of components increases, the true positive values consistently increase while the false negative values decrease. This finding suggests that the proposed method becomes more capable of correctly detecting positive cases as the number of components increases. In contrast, the false positive values remain relatively low across all component numbers, indicating that the proposed method can maintain a low rate of false alarms even with an increased number of components.

4.4 Discussion and limitations

Our findings, summarized in Table 5, suggest that carefully balancing dimensionality reduction with the need for accurate intrusion detection is critical for achieving optimal performance. It is found that PCA was the most effective dimension

Table 5 Comparison of the dimensionality reduction techniques with 1D CNN-GRU

Method	Accuracy	F1	Precision	Recall	TP	TN	FP	FN
1D CNN-GRU	0.9968	0.9869	0.9855	0.9882	10743	78955	158	128
Autoencoder	0.9965	0.9855	0.9823	0.9881	10742	78925	188	129
GED	0.9950	0.9793	0.9805	0.9781	10633	78901	212	238
PCA	0.9969	0.9873	0.9832	0.9915	10778	78929	184	93

Table 6 Time analysis of proposed CNN+GRU model

Hardware	Model	Train (sec/batch)	Test (sec/batch)
CPU (Intel(R) Xeon(R) CPU @ 2.20GHz)	CNN+GRU - with reduction (5 features)	0.004241	0.001734
	CNN+GRU - without reduction (51 features)	0.012813	0.003462
GPU (Tesla T4)	CNN+GRU - with reduction (5 features)	0.008120	0.002221
	CNN+GRU - without reduction (51 features)	0.008548	0.002481

reduction technique among the three methods evaluated, as it resulted in the best balance between the number of dimension and accuracy. PCA can slightly improve the accuracy and F1 score of CNN-GRU architecture with 20 components. On the other hand, reducing the input features by 90.2% using PCA resulted in only a 2.6% decrease in the F1 score of the intrusion detection system. When the number of components is decreased, the pure CNN-GRU model outperforms all experimented dimensionality reduction methods. This suggests that there may be tradeoffs between reducing dimensionality and maintaining accuracy and that each situation may require a different approach depending on the specific goals and constraints of the system being used. Overall, the findings suggest that careful consideration and testing of different dimensionality reduction techniques is necessary for optimization.

4.4.1 Timing analysis

In addition to the critical balance between dimensionality reduction and accurate intrusion detection discussed in the previous sections, it is essential to consider the aspect of time when evaluating the performance of dimensionality reduction techniques. It is important to note that latency is dependent on hardware capabilities. In this research, the performance evaluation was conducted on the Google Colaboratory ("Google Colab," n.d.) cloud service, utilizing the following system configuration at runtime: an Intel(R) Xeon(R) CPU @ 2.2 GHz, Nvidia Tesla T4 GPU, and 12 GB of RAM.

The time cost of the proposed CNN+GRU model with the full 51-feature set and the reduced 5-feature set is summarized in Table 6. The results indicate that reducing the feature set leads to enhanced efficiency, resulting in reduced processing time

Table 7 Time analysis of feature reduction methods

Hardware	Reduction method	Reduction time (sec)
CPU (Intel(R) Xeon(R) CPU @ 2.20GHz)	PCA	4.34250
	GED	0.41328
	Autoencoder	2206.18420
GPU (Tesla T4)	PCA	4.76829
	GED	0.43418
	Autoencoder	4369.23116

per batch for both the CPU and GPU. This improvement in computational efficiency highlights the advantages of employing dimensionality reduction methods.

However, it is worth noting that alongside the reduction in processing time, the accuracy may decrease. This trade-off between speed and accuracy should be carefully evaluated and considered when selecting a dimensionality reduction technique for the intrusion detection system. While faster processing times can be advantageous for real-time applications, the impact on accuracy should not be overlooked, as maintaining high detection performance is paramount in intrusion detection systems.

Furthermore, it is important to consider the time required for transforming data from a high-dimensional feature space to a low-dimensional feature space. In Table 7, the reduction times for different dimensionality reduction techniques on both CPU and GPU are presented. The result indicates that GED is the most efficient technique in terms of reduction time compared to PCA and Autoencoder. PCA has considerably lower reduction times compared to the Autoencoder method, which requires significantly higher epochs for training.

4.4.2 Limitations

Although promising results were achieved by the proposed method in securing the SWaT dataset, there are several limitations that should be acknowledged. Firstly, the research focused on detecting known attack types within the specific context of the SWaT dataset. The system's effectiveness in detecting new and emerging attack types remains unknown. As cyber threats constantly evolve, it is crucial to regularly update and enhance the IDS to address new attack vectors and techniques that may arise in the future.

Furthermore, the transferability of the proposed IDS to other industrial control systems or real-world deployments should be considered. Evaluating the performance of the IDS on different datasets or real-world scenarios is necessary to assess its generalizability and applicability in diverse environments.

Lastly, the time to detect attacks is a critical aspect that was not addressed in the study. While achieving high accuracy is important, the effectiveness of an IDS also relies on its ability to detect attacks in a timely manner. Further investigations should explore the system's response time and consider optimizing the detection process to minimize the time gap between attack occurrence and detection, thereby reducing potential damages or consequences.

5 Conclusion

This research explored the application of a 1D CNN and GRU model on the SWaT dataset with the aim of improving its performance by leveraging the strengths of both models. The findings clearly demonstrate that the fusion of the 1D CNN and GRU models leads to substantial enhancements in accuracy compared to using each model individually.

Furthermore, the study emphasizes the crucial role of dimensionality reduction in optimizing the model's performance. It highlights the significance of selecting relevant features, as this process can have a profound impact on the effectiveness of the intrusion detection system. Through an analysis of various dimensionality reduction techniques such as PCA, GED, and autoencoders, the research demonstrates that reducing the number of features while preserving critical information can lead to enhanced performance.

While the results obtained from this study are promising, there are still opportunities for further improvements in the current system. Future work should focus on investigating the feasibility of implementing the proposed method in real-time scenarios, enabling continuous monitoring and early detection of potential cyberattacks on critical infrastructure systems.

Furthermore, it is recommended to extend the scope of research beyond the SWaT dataset. Evaluating the effectiveness of the proposed method on other datasets related to critical infrastructure, such as power grids or transportation systems, would provide valuable insights into the generalizability and applicability of the approach in diverse contexts.

Author contributions All authors wrote the main manuscript text and reviewed the manuscript. All authors contributed equally to this work.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Abrams M, Weiss J (2008) Malicious control system cyber security attack case study-maroochy water services. The MITRE Corporation, McLean
2. David K (2013) The real story of stuxnet. IEEE Spect 50(3):48–53
3. Case DU (2016) Analysis of the cyber attack on the Ukrainian power grid. Electr Inform Shar Anal Center 388:1–29
4. Kovacs E (2023) Industrial systems at risk of wannacry ransomware attacks, https://www.securityweek.com/industrial-systems-risk-wannacry-ansomware-attacks, accessed: 2023-01-11
5. Electric S (2023) Important security notification security notification-wannacry ransomware attack, https://www.se.com/ww/en/download/document/SEVD-2017-135-01/, accessed: 2023-06-02

6. Di Pinto A, Dragoni Y, Carcano A (2018) Triton: the first ICS cyber attack on safety instrument systems. In: Proc. Black Hat USA, Vol. 2018, pp 1–26
7. Kovacs E (2023) Triton is the world's most murderous malware, and it's spreading - MIT technology review, https://www.technologyreview.com/2019/03/05/103328/cybersecurity-critical-infrastructure-triton-malware, accessed: 2023-06-02
8. Adepu S, Mathur A (2018) Distributed attack detection in a water treatment plant: method and case study. IEEE Trans Dependable Secure Comput 18(1):86–99
9. Das TK, Adepu S, Zhou J (2020) Anomaly detection in industrial control systems using logical analysis of data. Comput Secur 96:101935
10. Gold D (2023) Is signature- and rule-based intrusion detection sufficient?, https://www.csoonline.com/article/3181279/is-478signature- and-rule-based-intrusion-detection-sufficient.html, accessed: 2023-02-28
11. Al-Dhaheri M, Zhang P, Mikhaylenko D (2022) Detection of cyber attacks on a water treatment process. IFAC-PapersOnLine 55(6):667–672
12. Boateng EA, Bruce J, Talbert DA (2022) Anomaly detection for a water treatment system based on one-class neural network. IEEE Access 10:115179–115191
13. Kravchik M, Shabtai A (2018) Detecting cyber attacks in industrial control systems using convolutional neural networks. In: Proceedings of the 2018 workshop on cyber-physical systems security and privacy, pp 72–83
14. Xie X, Wang B, Wan T, Tang W (2020) Multivariate abnormal detection for industrial control systems using 1D CNN and GRU. IEEE Access 8:88348–88359
15. Goh J, Adepu S, Tan M, Lee ZS (2017) Anomaly detection in cyber physical systems using recurrent neural networks, In: 2017 IEEE 18th international symposium on high assurance systems engineering (HASE). IEEE 140–145
16. Zhou L, Zeng Q, Li B (2022) Hybrid anomaly detection via multihead dynamic graph attention networks for multivariate time series. IEEE Access 10:40967–40978
17. Nedeljkovic D, Jakovljevic Z (2022) CNN based method for the development of cyber-attacks detection algorithms in industrial control systems. Comput Secur 114:102585
18. Sung DCL, MR GR, Mathur AP (2022) Design-knowledge in learning plant dynamics for detecting process anomalies in water treatment plants
19. Li Z, Duan M, Xiao B, Yang S (2022) A novel anomaly detection method for digital twin data using deconvolution operation with attention mechanism, IEEE Trans Indust Inform
20. Alimi OA, Ouahada K, Abu-Mahfouz AM, Rimer S, Alimi KOA (2022) Supervised learning based intrusion detection for scada systems. In: 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), IEEE, pp 1–5
21. Krithivasan K, Pravinraj VSSS (2020) Detection of cyberattacks in industrial control systems using enhanced principal component analysis and hypergraph-based convolution neural network (epca-hg-cnn). IEEE Trans Indust Appl 56(4):4394–4404
22. Elnour M, Meskin N, Khan K, Jain R (2020) A dual-isolation-forests-based attack detection framework for industrial control systems. IEEE Access 8:36639–36651
23. Yazdinejad A, Kazemi M, Parizi RM, Dehghantanha A, Karimipour H (2023) An ensemble deep learning model for cyber threat hunting in industrial internet of things. Digital Commun Netw 9(1):101–110
24. iTrust Laboratory, Secure water treatment (swat), https://itrust.sutd.edu.sg/itrust-labs_datasets/#SWaT, accessed: 2023-01-11
25. Kravchik M, Shabtai A (2019) Efficient cyber attacks detection in industrial control systems using lightweight neural networks. arXiv preprint arXiv:1907.01216
26. Macas M, Wu C (2019) An unsupervised framework for anomaly detection in a water treatment system. In: (2019) 18th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp 1298–1305

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.