Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 7 (2025) 2689

Journal of the Nigerian Society of Physical Sciences

Detecting network intrusions in cyber-physical systems using deep autoencoder-based dimensionality reduction approach and deep neural networks

A. E. Ibor[a,*], D. O. Egete[a], A. O. Otiko[b], D. U. Ashishie[a]

[a]*Department of Computer Science, University of Calabar, Calabar, Nigeria*
[b]*Department of Computer Science, University of Cross River State, Calabar, Nigeria*

Abstract

Cyber-Physical Systems (CPSs) that integrate computational and physical processes are the foundation of reliability in prominent areas of critical infrastructure, including transportation, energy, and manufacturing. The expansion in connected CPSs has made them vulnerable to various and changing intrusions into their networks. This research proposes a hybrid deep learning architecture that integrates the utilisation of a denoising autoencoder as a feature dimensionality reduction component with a five-layer deep feedforward neural network as an effective intrusion classifier. The model is trained and tested on CICIDS2017 and UNSW-NB15 datasets with a rich collection of attack patterns such as DoS, DDoS, Shellcode, and Worm attacks. The denoising autoencoder effectively learns higher-level representations of network traffic data, whereas the deep feedforward network facilitates precise multi-class classification. Empirical results demonstrate that the model achieves 99.99% and 99.95% detection accuracies o n CICIDS2017 and UNSW-NB15 datasets, respectively, at very low false positive rates. Comparative analysis with state-of-the-art techniques further confirms the superior performance and generalisability of the presented solution, highlighting its applicability to real-time CPS threat detection systems.

DOI:10.46481/jnsps.2025.2689

Keywords: Adversarial attacks, Deep autoencoder, Deep learning, Intrusion detection, Cyber-physical systems

Article History :
Received: 17 February 2025
Received in revised form: 25 April 2025
Accepted for publication: 08 May 2025
Published: 12 June 2025

Communicated by: O. Akande

1. Introduction

Transportation systems, industrial control processes, power systems, healthcare systems, and critical infrastructures all use Cyber-Physical Systems (CPSs) [1, 2]. Cyber Physical Systems are types of computer systems that combine computation and physical processes and have been the target of sophisticated cyberattacks. Recent on CPSs, such as the StuxNet worm, ransomware, Denial of Service (DoS), Distributed DoS (DDoS), replay, covert, integrity, stealthy deception, and false data injection attacks, have resulted in the modification of system parameters that control the behaviour of sensors, controllers, and actuators in a cyber-physical ecosystem [3–9].

Cyber-physical systems, like traditional computational and physical systems viewed individually, are vulnerable to failures and cyberattacks, including new vulnerabilities and exploits with no rapid fix [10, 11]. The dangers of exploiting

*Corresponding author Tel. No: +2348024947230.
Email address: ayei.ibor@unical.edu.ng (A. E. Ibor)

the vulnerabilities revealed in cyber-physical systems, according to Ref. [12], can spark a worldwide security crisis with unforeseeable effects on control and automated systems. Attacks against cyber-physical systems, in particular, can disrupt vital infrastructure and industrial control systems, as well as result in massive data dumps, intellectual property theft, and confidential strategic operations.

The distinctive qualities of cyber-physical entities imply that they are persistently separate from traditional IT systems [3, 4, 13]. This also means that CPSs face unique threats and implications, which are still being investigated. This study offers a deep learning model to detect primarily DoS, DDoS, Shellcode, and Worm intrusions on cyber-physical systems to better understand these threats and their potential for escalation and disruption of CPS confidentiality, integrity, and availability. The unavailability of resources on a smart grid, for example, as a result of DoS and DDoS attacks, might cause important physical processes in an open-loop to become unstable.

A successful DoS or DDoS attack on the sensor measurements at this point may distract the controller from averting serious system and resource damage [14, 15]. Similarly, a shellcode attack can leverage software vulnerability in a CPS to compromise a server on the CPS network, according to Ref. [16]. Furthermore, based on security flaws on the target devices, a worm can swiftly spread across a CPS network by reproducing itself across all servers and workstations. Because worms proliferate using a recursive technique based on the idea of experimental growth, this attack can infect multiple machines in a CPS network in a very short time [17].

Sensor networks in CPSs capture a large quantity of data, which is currently the focus of DoS, DDoS, Shellcode, and Worm intrusions. Similarly, various intrusions have targeted the commands provided by controllers, including the actions taken by actuators [18–21]. Significantly, preventing attacks from succeeding may be impossible. Minor changes in the control systems, such as irregular process flow or sensor behaviour, can, however, be detected as a result of an intrusion. Deep learning can be used to detect such tiny changes in the process flow, as well as sensor and actuator behaviour. Deep learning has recently gained popularity in the field of cybersecurity, with a particular focus on the detection and prediction of threats [3, 22–25].

In spite of extensive application of conventional Intrusion Detection Systems (IDS) [26, 27], such systems are commonly unable to deal with the high dimensional and dynamic Cyber-Physical Systems (CPS) traffic flows. Recent deep learning models have demonstrated excellent potential in identifying fine-grained anomalies that are not detected by signature-based systems. For example, Ajagbe *et al.* [1] suggested a new convolutional neural network (CNN) designed specifically for intrusion detection in Internet of Things (IoT) services and reported better performance over traditional classifiers. Similarly, Awotunde *et al.* [2] highlighted the necessity of blockchain and AI-driven approaches in CPS security, particularly in the face of covert and stealthy attacks.

To this effect, this paper proposes a modular deep learning pipeline for network intrusion detection and classification in CPSs. Our contributions are as follows:

1. We propose a two-level system consisting of an unsupervised denoising autoencoder to reduce dimensionality, and a supervised deep feedforward neural network for classification tasks.
2. Each step of methodology is stringently described based on an "Input → Process → Output" approach for improving reproducibility and transparency.
3. The model is evaluated using two widely used benchmark datasets, CICIDS2017 and UNSWNB15, and the results are compared with those of well-known state-of-the-art approaches.
4. We provide a detailed discussion on architectural choices, performance trade-offs, dataset-specific behaviours, and real-world applicability.

Deep learning's capacity to learn representations from raw data is a significant strength. By learning the representation and behaviour of intrusions directly from captured malicious network traffic, we used deep learning to develop a model that can detect attacks on CPSs. To achieve this, we extracted attack samples from the CICIDS2017 and UNSW NB datasets including Benign (normal), DoS, DDoS, Shellcode, and Worms to represent the collected malicious network traffic from which our model can learn.

We used an autoencoder and a deep feedforward neural network (DFFNN) on a python environment testbed to train, validate, and test the model. In addition, we evaluated the model's performance on the extracted attack data in terms of detection accuracy, precision rate, recall rate, false positive rate, and loss, and compared it to similar models. Based on the results, our model outperformed existing attack detection methods, indicating that it is fit for purpose in the detection of intrusions on CPS networks.

The rest of the paper is organized as follows: Section 2 gives a discussion of related works while in Section 4, Materials and method is presented. Results and discussion are presented in Section 4 and the Conclusion in Section 5.

2. Related work

Computer-based algorithms and procedures manage and monitor mechanisms in cyber-physical systems. Adversarial attacks are aimed at these algorithms and processes. To model integrity attacks on CPSs, Mo and Sinopoli [27] employed a discrete linear time invariant system. An attacker could compromise a CPS by introducing external control inputs and fake sensor data, according to the strategy. The authors were able to characterize the reachable components of the system state and estimate the error under attack to assess the system's resilience to integrity attacks. They also used an ellipsoidal approach to find the accessible set's outer approximations.

Tabassum *et al.* [28] investigated inverter-based microgrid anomaly detection using an autoencoder neural network. Its novelty lies in its implementation on real power systems data

Table 1. Summary of samples used for the experiment.

CICIDS2017 Dataset Class	Description	Number of Samples
Benign (Normal)	Normal network traffic	5003
DoS	The Denial of Service (DoS) attack temporarily or indefinitely disrupts services on a host machine connected to the Internet. These services then become unavailable to the intended users for the period of the attack	11, 936
DDoS	Usually results from a botnet of compromised machines flooding the bandwidth or resources of a victim machine	127, 538
UNSW_NB15 Dataset Class		
Normal (Benign)	(See above)	56,000
DoS	(See above)	12,264
Shellcode	The exploitation of a software vulnerability using a small piece of code as a payload	1,133
Worms	An attack that can replicate itself across multiple connected systems or networks	130

and specific focus on voltage and frequency anomalies. However, its application is restricted to some physical system abnormalities without the capability of making broader classifications on a range of cyber threats, including network-based attacks. Aljehane [29] proposed a parameter-tuned deep stacked autoencoder to defend CPS against intrusions. Hyperparameter optimization was the emphasis of the research to improve detection accuracy. While the model achieves satisfactory detection performance, the computational overhead and tuning complexity reduce its feasibility to real-time applications in heterogeneous CPS setups.

In a comprehensive review, Haq *et al.* [30] surveyed a number of autoencoder and Restricted Boltzmann Machine (RBM) models with special tailoring to CPS. The authors categorically classify generative models based on architecture and training methods and provide a strong theoretical background. The review is, however, lacking in empirical benchmarking and comparative performance. Rajathi *et al.* [31] introduced a new autoencoder model using reinforcement learning for adaptive intrusion detection in CPS. The system adapts optimal feature selection policies in real time, hence minimizing false positives. Although innovative, the lack of large-scale validation and benchmark comparison restricts its generalisability.

Kaur *et al.* [32] employed a Bayesian deep learning approach combined with CNN-based feature engineering for smart grid networks. The Bayesian approach introduces predictive uncertainty quantification, which is an untapped potential in CPS security. However, the increased computational overhead raises deployment concerns. Nuiaa Al Ogaili *et al.* [33] created PhishNetVAE, a VAE-DNN hybrid framework for phishing attack detection. Not CPS specific, their architecture does provide some insight into learning latent structure that can be transferable to detecting zero-day attacks. The applicability of the study to CPS environments would need to be adapted to time-series and control signal inputs.

Sugunaraj and Ranganathan [34] surveyed the use of autoencoders in power system applications with emphasis on their application in anomaly detection and load forecasting. The survey, though extensive, does not delve into temporal modelling and adversarial robustness, which are crucial in CPS. Kousar *et al.* [35] employed a deep autoencoder for dimensionality reduction in smart grid anomaly detection. The curse of dimensionality is alleviated by the pretraining strategy with an unsupervised method. Nevertheless, it is not compared with conventional reduction techniques such as PCA or t-SNE. Alsaade and AlAdhaileh [36] proposed a deep autoencoder-based model for securing vehicular CPS networks. While the model performs effectively in vehicular anomaly detection, its utilization in non-automotive fields is limited since it requires significant domain-specific tuning. Ortega-Fernandez *et al.* [37] targeted DDoS detection in Industrial Control Systems (ICS) using a deep autoencoder-based intrusion detection system. The model is effective at volumetric traffic filtering but is not tested against adversarial or zero-day attacks.

Kukkala *et al.* [38] proposed the use of recurrent autoencoders for real-time intrusion detection in vehicle CPS. With the capability of learning temporal dependencies, their model shows proficiency in handling time-series attack signatures. However, it remains CAN bus specific. Saranya and Valarmathi [39] suggested a multilayer autoencoder approach to cross-layer attack detection for IoT networks. The fusion of multiple protocol layers improves the accuracy of classification. The research is relevant to the illustration of hierarchical learning in CPS but misses complexity analysis. Harrou *et al.* [40] utilized deep autoencoders for anomaly detection in power grids. Their work, based on SCADA data, is efficient and effective but lacks coverage of attack generalization and interpretability issues. Zhang *et al.* [41] conducted a comprehensive survey on CPS attack detection through deep learning. The taxonomy provided is complete and insightful, though some of the latest architectural advancements like Transformers are not covered.

D'Angelo and Palmieri [42] proposed a hybrid architec-

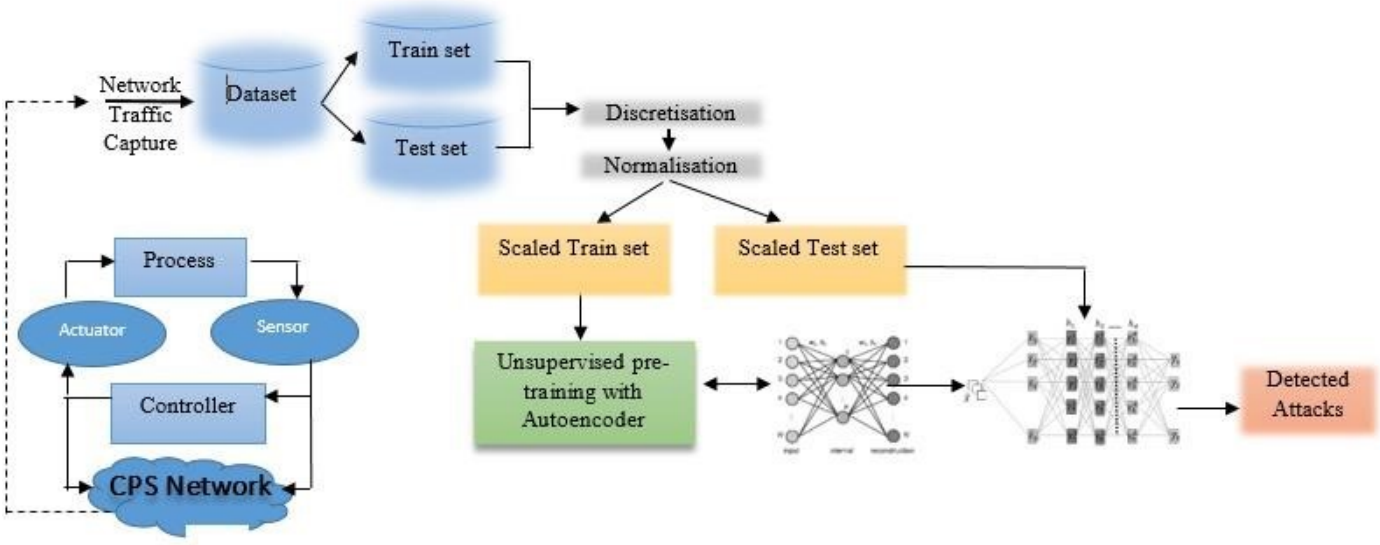

Figure 1. Architecture of the proposed intrusion detection model.

Table 2. System properties of the implementation machine.

Host Operating System	Ubuntu 18.04
Processor	Intel ® Core ™ i3 6100U CPU @2.30 GHz 2.30GHz
RAM	4.00GB
System Type	64-bit Operating System, x-64 based processor

Table 3. Model's evaluation metrics for CICIDS2017 samples.

	Benign	DDoS	DoS
ACC	99.99	99.86	99.98
PR	99.99	99.98	99.99
RR	99.99	99.99	99.99
F1	99.99	99.99	99.99
L	0.000046	0.00014	0.000092

Table 4. Confusion matrix of the CICIDS2017 samples.

Benign	761	1	0
DDoS	1	19135	2
DoS	0	2	1770
	Benign	DDoS	DoS

Table 5. Model's evaluation metrics for UNSW_NB15 samples.

	DoS	Normal	Shellcode	Worms
ACC	97.87	93.15	99.01	99.91
PR	98.92	99.39	99.61	99.96
RR	99.28	99.25	99.41	99.95
F1	99.10	99.32	99.51	99.96
L	0.0107	0.0060	0.0039	0.0//004

ture based on autoencoders, CNNs, and RNNs for detecting attacks in interdependent power control systems. Their extensive ablation studies demonstrate architectural merits, though system scalability remains unaddressed. Zideh et al. [44] proposed an adversarial autoencoder for power grid security. The model enhances generalization through the addition of a generative adversarial component but fails to compare its performance with VAEs or transformer-based models. Ma *et al.* [44]

Table 6. Confusion matrix of the UNSW˙NB15 samples.

DoS	1731	46	25	3
Normal	61	8341	15	1
Shellcode	47	14	126	0
Worms	3	2	0	15
	Dos	Normal	Shellcode	Worms

reported deep learning applications in secure communication for CPS. Though theoretical, the paper broadens the cybersecurity paradigm to integrity and authentication, in addition to intrusion detection. Roshanzadeh *et al.* [45] suggested a CNN-AE ensemble model to detect attacks in AC microgrids from multivariate time-series. Their work achieves high sensitivity and low FPRs from real control data but is plagued by high computational overhead and poor portability. These works collectively highlight the effectiveness of CPS intrusion detection using autoencoder-based deep learning. Nevertheless, this paper's proposed model is designed to address limitations in real-time usability, scalability, and interpretability.

Chen *et al.* [46] used a finite state model and a hidden Markov chain to explain the detection of malicious intrusions on CPSs. This method was developed to detect multi-stage intrusions against CPSs to protect key resources and infrastructure. Their findings were promising, with an accuracy of 86.2% in detecting joint attacks, a PR of 93.2%, and an RR of 84.1%. From the standpoint of the system, Nizam *et al.* [14] looked at the detection and prevention of cyber-physical system attacks. To recognize and classify DDoS and fake data injection attacks, they used the Chi-Square detector and the Fuzzy Logic-based attack classifier (FLAC). They identified these attacks with 96% accuracy using activity profiling, average packet rate, change point detection algorithm, cosum algorithm, and other criteria.

In a DoS attack scenario, Nur *et al.* [15] investigated a unique probabilistic packet marking system to discover forward pathways from an attacker to a victim site. They discovered that constructing a forward path from the attacker requires an average of 23 packets. The victim site assembles the forward path by combining the recorded IP addresses in the option field from all sources. Starting with an empty forward pathways graph, the victim site can populate it with sub-paths found by the record route. Covert and zero dynamics attacks are extremely difficult to execute and rely heavily on complete system knowledge.

To this end, Ref. [6] proposed a method for inserting a modulation matrix in the path of the control variables to modify the process's input behaviour and reveal an adversary's attacks. The modulation matrix was designed to identify clandestine attacks. They hypothesized, however, that the modulation matrix may also change the input orientations, revealing zero dynamics attacks. An adaptive design was presented by Jin *et al.* [47] to prevent attacks on a CPS's sensors and actuators. The adaptive controller was created to ensure consistent ultimate boundedness of the closed loop dynamical system while the sensors and actuators are under attack, to address security and safety in CPSs.

A semi-supervised technique for detecting unfamiliar CPS attacks is investigated in Ref. [48]. The method might incorporate knowledge about unfamiliar malware from both labelled and unlabelled data into the detection methodology automatically. They extracted dynamic changes in malware attack patterns from unlabelled data using unsupervised clustering. The cluster data was extracted using a Support Vector Machine (SVM) technique using global K-means clustering using term frequency, inverse document frequency, and cosine similarity as the distance measure. The SVM classifier had a static analysis of malware data accuracy of 95.79% and a dynamic analysis accuracy of up to 100%.

Beg *et al.* [49] investigated how to detect false-data injection attacks on cyber-physical DC microgrids. The detection challenge entails identifying a change in a set of inferred candidate invariants. They used Simulink Stateflow diagrams to simulate the physical plant and software controller of the CPS, and defined invariants as microgrid properties that do not change over time. The potential invariants are compared to the real invariants to detect a false-data injection attack, and any disparity is notified as an attack. Their model's performance was encouraging. Chhetri *et al.* [50] proposed a novel method for identifying kinetic cyberattacks. The method was created to identify zero day kinetic cyberattacks on an Additive Manufacturing CPS by detecting abnormal analogue emissions that indicate the presence of an attack. Their model, which had an accuracy of 77.45%, was based on a statistical estimate of functions mapping the relationship between analogue emissions and cyber domain data to predict system behaviour. In a CPS, Sadreazami *et al.* [51] introduced a unique distributed blind architecture for detecting intrusions. Sensor readings are treated as the goal graph signal in this technique. In addition, the graph-statistical signal's features are employed to detect intrusions. The proposed technique is based on a modified Bayesian likelihood ratio test, and the test statistic's closed-form expressions are produced a 99.75% accuracy rate. Ferdowsi and Saad [52] investigated the detection of adversarial attacks on the Internet of Things (IoT). When deployed on privacy-preserving IoT networks, the authors proposed a distributed intrusion detection strategy utilizing generative adversarial networks (GAN), which generated highly promising results. Their method may detect unusual behaviour in a node without the need for a centralized controller, which is available in standalone intrusion detection systems (IDS). GAN, on the other hand, is based on the adversarial learning concept, which requires the balancing and synchronization of two adversarial networks throughout the training process [53, 54]. In the absence of this balance and synchronisation, good training results may be difficult to achieve.

Thakur *et al.* [55] proposed an intrusion detection system

Table 7. Overall performance of our model for the extracted samples.

	CICIDS2017 Samples	UNSW_NB15 Samples
ACC	99.99	98.95
FPR	0.00131	0.03404
PR	99.99	99.27
RR	99.99	99.45
F1	99.99	99.36
L	0.00005	0.007233

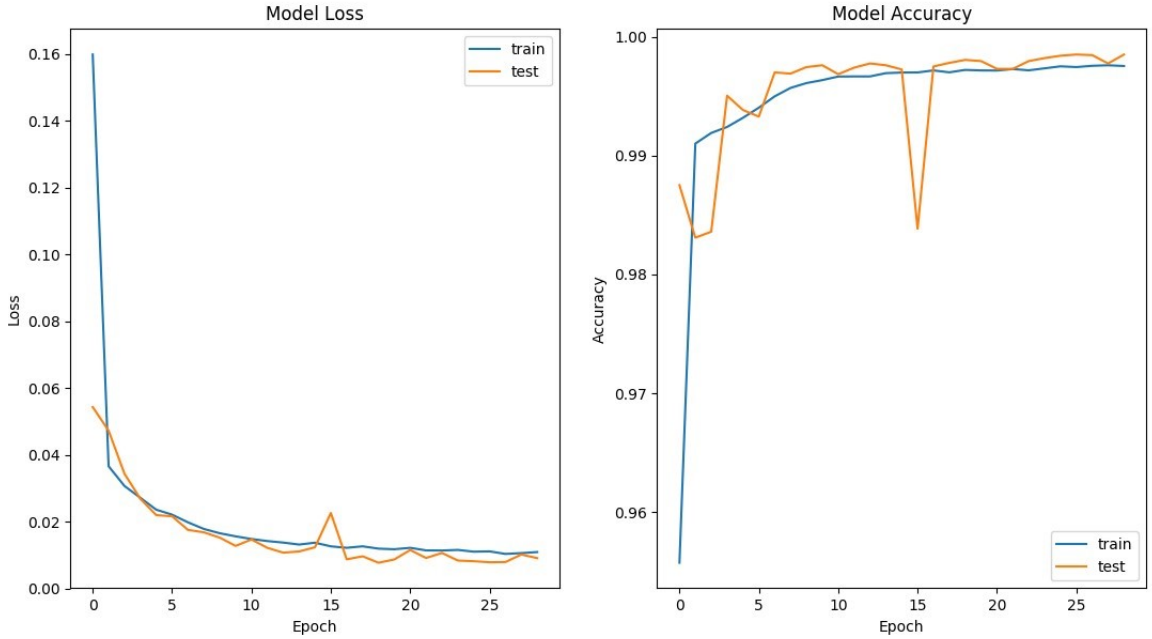

Figure 2. Accuracy and loss of our model for the CICIDS2017 samples.

for cyber-physical systems based on a model that uses generic and domain-specific autoencoders. The model classifies intrusions using a deep learning method that learns the features that are common to all types of network intrusions based on unique generic autoencoder architecture. Specific traits are also learned concerning the problem domain in their model, with encouraging results from experiments. However, they only evaluated their approach on a single dataset, which may be insufficient for evaluating a model in a complex attack scenario like the cyber-physical ecosystem.

Although numerous methodologies for the analysis, identification and detection of attacks on CPSs are addressed here, there are several key gaps. First, most existing models are bound by computationally prohibitive costs, making them less suitable for real-time applications. Second, many works evaluate their models on one dataset only, which frustrates generalisability. Third, few models present a formal, reproducible methodology that is easily extendable to new attack types or CPS setups. While significant advancements have been made in applying deep learning to intrusion detection, current methods tend to function as black boxes with minimal insight into feature transformation processes. In addition, methods lack cross-dataset validation or suffer from overfitting due to inadequate pre-training strategies. These drawbacks limit the realistic application of these models to CPS settings where efficiency and interpretability are critical.

Moreover, by creating packets that can avoid network defences, attackers have recently discovered new techniques to thwart numerous detection and prevention approaches. Fundamentally, as technology advances, new vulnerabilities on CPSs arise regularly, and a technique that can learn the representation of an attack with less computing cost will become increasingly important as the threat landscape evolves. This will aid in the detection of adversarial attacks by modelling the data representation rather than the set of tasks used to detect them.

3. Materials and method

3.1. Datasets

To understand the behaviour and attributes of DoS, DDoS, Shellcode, and Worm attacks, we extracted samples of these

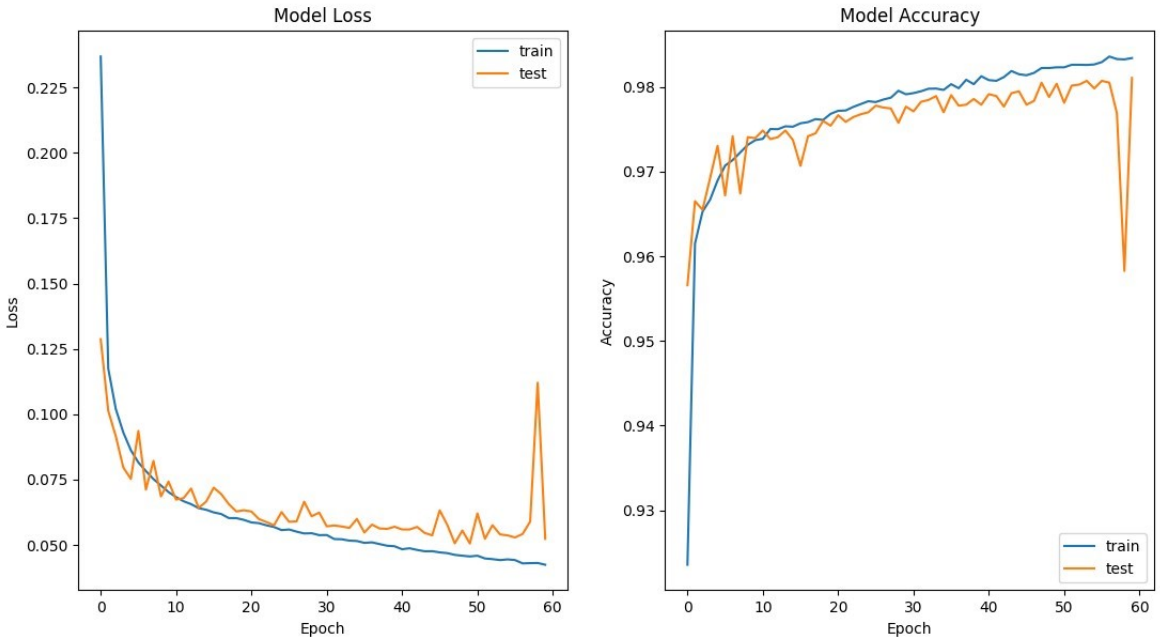

Figure 3. Accuracy and loss of our model for the UNSW_NB15 samples.

attacks from the CICIDS2017 and UNSW_NB datasets. The CICIDS2017 dataset is a time-based dataset generated over five days. It has 80 features with 13 attack types and 1 benign (or normal) traffic class [56–58]. These 13 attack types were classified into 7 broad attack types [59]. From the 7 broadly classified attack types, we extracted 11, 936 DoS samples and 127, 538 DDoS samples as well as 5003 benign network traffic samples for the training, validation, and testing of our model. The CICIDS2017 dataset has attack diversity, complete network configuration, complete traffic, labelled dataset, heterogeneity, and feature set. It has been used as the benchmark dataset for attack prediction and detection systems [60–62], and several other works.

Similarly, the UNSW_NB15 dataset is a time-based dataset generated over 16 hours for the training set, and 15 hours for the test set. It has 9 attack types and 49 features [63], and also a benchmark dataset for evaluating intrusion prediction and detection systems [64, 65]. From the 9 attack types in the dataset, we extracted 12,264 DoS samples, 56,000 normal (benign) network traffic samples, 1,133 shellcode samples, and 130 worm samples, which are also used in this model. The model is trained with the training set, validated with the validation set, and tested with out-of-sample data in the test set using a parentage split of 70:15:15 for all experiments. A summary of the samples used for the experiments is given in Table 1.

3.2. Methodology

The captured network traffic data, which is the input to the model, undergoes two learning processes. First, unsupervised pre-training with a denoising autoencoder is used to perform dimensionality reduction (feature selection and extraction) to overcome the curse of dimensionality on the input data [66]. Denoising autoencoders have been used in speech enhancement, fault diagnosis, collaborative filtering and several other real-world problems [67–69]. The main task of the denoising autoencoder used in this work is to reduce the risk of learning the identity function during training, which arises when there are more hidden layers than inputs in an autoencoder. This usually leads to the output becoming equal to the input, thus rendering the autoencoder useless [68, 70].

Data collected by sensors in a cyber-physical network is enormous. Identifying the malign data from this large dataset requires a high-level representation of the data to avoid overfitting, high computational overheads and inevitably extensive training time that may lead to a complex model. To give a better understanding of the intrinsic structure of data, a denoising autoencoder is used as a crucial pre-processing phase to map the high dimensional data to a higher-level representation or lower-dimensional space that helps to remove bias from the data at the time of building the model. Thus, our model achieves two things in the unsupervised pre-training phase:

1. A reasonably stable high-level representation of the modelled data, which is robust to the corruption of the input
2. The extraction of features that capture intrinsic structure in the distribution of the input.

Furthermore, unsupervised pre-training is significant for overcoming the distortions in the gradient caused by the multiple layers of the deep neural network [71]. Pre-training our model with a deep denoising autoencoder provides a good initialisation for the deep feedforward neural network to fine-tune

it. At the next stage, we used a 5-layer supervised deep feedforward neural network to train the model for making predictions on test data. Goodfellow *et al.* [72] describe deep feedforward networks as function approximation machines, which are designed to achieve statistical generalisation. Our deep feedforward network consists of many functions chained together like a directed acyclic graph of *degree n*. This structure includes five layers; the input layer, 3 hidden layers, and an output layer. The model performs cascaded learning through pre-activation and activation processes at each layer to learn the representation of each attack type from the dataset. The model also optimises the hyperparameters in the hidden layers to accurately detect the modelled attacks from the test data.

3.2.1. The proposed model

(a) Model Architecture. As depicted in the architecture of our model in Figure 1, a CPS integrates physical processes, ubiquitous computation, efficient communication and control. Hence, network control systems, wireless sensor and actuator networks, and wireless industrial sensor networks are a subgroup of CPS [73, 74]. In our approach, we consider three (3) attack locations in the CPS infrastructure. These include the actuator, sensor and controller. Actuator networks, sensor networks and communication networks are prone to common attacks on cyber-physical systems such as DoS, DDoS, Shellcode and Worms. Consequently, our network intrusion detection system can be located at any strategic point in the CPS network to capture traffic that is used by the model to flag the presence of an attack.

From Figure 1, both malign and benign traffic is captured at different time stamps from the CPS network as input to the model. This traffic may contain different attack connections. However, we are only interested in the DoS and DDoS attack connections, which are simulated in this work using samples extracted from the CICIDS2017 and UNSW_NB15 datasets. The captured network traffic is split into the train (including validation) and test sets using an 85:15 ratio. 15% of the train set is used as the validation set. The training set is used as input to the autoencoder for the unsupervised pre-training of the model after the processes of discretisation and normalisation. The validation set provides an unbiased evaluation of the model and fits the training data while tuning the hyperparameters of the model. The test set is used to evaluate the trained deep feedforward neural network to classify the benign and malign traffic. A deep learning model only works on discrete and continuous values. In this sense, the captured network traffic is discretised and normalised to have a set of continuous values for the inputs and class values for the outputs. The network traffic used as the dataset consists of x inputs and y outputs.

(b) Dataset normalisation. Let $X \in \mathbb{R}^{n\times d}$ represent the dataset, where n is the number of samples or instances, and d is the number of features in each sample. We denote each sample as given in Equation (1).

$$x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T, \quad i \in \{1, 2, \ldots, n\}. \tag{1}$$

The dataset contains both normal and attack samples, denoted as y_i, where $y_i = 0$ depicts normal or benign samples, and $y_i \geq 1$ depicts intrusions.

Z-score normalisation, also called standardisation, is used to transform the dataset to comparable scales, to achieve accurate predictions in our model. This method of normalisation is discussed in Ref. [75] and used in Ref. [24]. The normalisation process is defined in Equation (2).

$$\bar{x}_i = \frac{x_i - \mu_i}{\sigma_i}, \quad \mu_i = \frac{1}{n}\sum_{j=1}^{n} x_{ji}, \quad \sigma_i = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(x_{ji} - \mu_i)^2}, \tag{2}$$

where x_i represents the original samples in the dataset, μ_i is the mean, and σ_i is the standard deviation of the samples. $\bar{x}_i$ denotes the normalised samples.

(c) Dimensionality reduction with denoising autoencoder. The scaled training samples from the normalisation process are further learned by a denoising autoencoder through nonlinear feature reduction to remove bias from the dataset, allowing each variable or feature to contribute equally to the analysis of the data. The autoencoder performs compression of the samples to achieve a lower-dimensional code, which is then used to reconstruct the original input. Our unsupervised pre-training approach extends the works of [76–79].

In the unsupervised pre-training phase, the compressed code represents the latent-space representation of the input and approximates the original samples optimally. This is useful for overcoming the curse of dimensionality on the input data, thereby reducing the computational overhead of the model.

Given an n-dimensional input vector, $x \in \mathbb{R}^n$, an m-dimensional output vector, $y \in \mathbb{R}^m$, a weight matrix W, and a bias vector b, we use the autoencoder to perform the following operations on the original samples:

Encoding: This involves the transformation of the original data into a compressed code in the hidden layers after introducing corruption through stochastic mapping. The Rectified Linear Unit (ReLU) activation function is used in the hidden layer as the activation function, as given in Equation (3).

$$z = f_{\text{enc}}(x; \theta_{\text{enc}}), \tag{3}$$

where $z \in \mathbb{R}^k$ $(k < d)$ is the compressed latent representation, and θ_{enc} represents the parameters of the encoder network.

A further expansion of Equation (3), as adopted from the work of Wang et al. [77], gives:

$$z = g_\theta(x) = \alpha(W_{\text{enc}}x + b_{\text{enc}}), \quad z \in \mathbb{R}^k,\ k \ll d, \tag{4}$$

where W_{enc} is the encoder weight matrix, b_{enc} is the encoder bias vector, and α denotes the ReLU activation function.

Decoding: The autoencoder reconstructs the original input data from the compressed code. The reconstructed output $\hat{x}$ is defined as:

$$\hat{x} = h_\phi(z) = \alpha(W_{\text{dec}}z + b_{\text{dec}}), \quad \hat{x} \in \mathbb{R}^d, \tag{5}$$

where W_{dec} and b_{dec} are the decoder weight matrix and bias vector, respectively. These weights and biases are randomly initialised and updated iteratively during training through back-propagation.

Reconstruction Error: The reconstruction error allows the autoencoder to learn meaningful latent representations. It is defined as:

$$\mathcal{L}_{\text{AE}} = \frac{1}{N}\sum_{i=1}^{N}\|x_i - \hat{x}_i\|^2. \tag{6}$$

The output $\hat{x}$ of the autoencoder is used as input to our deep feedforward neural network.

(d) Classification by Deep Neural Network. To overcome overfitting (low bias, high variance) or underfitting (high bias, low variance), each neuron in the deep neural network undergoes pre-activation, which is the weighted sum of inputs plus a bias. Following the pre-activation, the ReLU activation function α is applied to determine whether the neuron will fire. The first neuron in the first hidden layer is connected to each of the input weights W_i.

The pre-activation at each layer is the weighted sum of the inputs from the preceding layer and the bias [22]. The deep neural network (DNN) performs the classification of the latent representation z into normal (benign) or intrusion (attack) categories.

Let z be the input to the DNN from the deep autoencoder. The DNN consists of multiple layers, and the forward pass is given by:

$$o^{(l)} = \alpha\left(W^{(l)}o^{(l-1)} + b^{(l)}\right), \quad l = 1, 2, \ldots, L, \tag{7}$$

where $o^{(0)} = z$, the input to the DNN, $W^{(l)}$ and $b^{(l)}$ are the weights and biases for the l^{th} layer, and α is the activation function in the hidden layers. The pre-activation and activation processes enable our model to learn the representations of the attack classes from the datasets using the train and validation samples. The model also optimizes the hyperparameters in the hidden layers to produce highly accurate outputs during testing. During the test phase, the test samples are introduced to the trained model to make predictions using the training samples as memory.

At the final or output layer, the Softmax activation function is used since we are modeling a multi-class classification problem. The predicted output is given by:

$$\hat{y} = \text{softmax}(W^{(L)}o^{(L-1)} + b^{(L)}). \tag{8}$$

At the output layer, the Softmax function is employed to classify the attack types. The Softmax function partitions the output such that the total sum equals 1, which corresponds to a categorical probability distribution [80]. In other words, the final layer consists of a single neuron for each of the attack classes, and each neuron outputs a value between 0 and 1, interpreted as a probability.

The total sum of these probabilities is 1. The probability of an attack or a normal connection is computed using:

$$\hat{y}_i = \text{softmax}(W^{(L)}o^{(L-1)} + b^{(L)}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{9}$$

where $\hat{y}_i$ is the predicted probability for the i^{th} class, and z_i is the i^{th} element of the logit vector z containing K real-valued elements. The Softmax function applies the exponential function to each element z_i and normalizes the result by the sum of exponentials over all K elements.

We denote the final output of the DNN classifier as:

$$\hat{y}_i = f_{\text{DNN}}(z_i; \theta_{\text{DNN}}), \tag{10}$$

where $\hat{y}_i \in [0, 1]$ is the probability of an intrusion, and θ_{DNN} represents the learnable parameters of the DNN.

Performance Metrics

We used the following performance metrics to evaluate the performance of our model:

- Accuracy (ACC):

$$\text{ACC} = \frac{TP + TN}{TP + TN + FN + FP}, \tag{11}$$

where TP is True Positive, TN is True Negative, FN is False Negative, and FP is False Positive.

- False Positive Rate (FPR):

$$\text{FPR} = \frac{FP}{TN + FP}. \tag{12}$$

- Precision Rate (PR):

$$\text{PR} = \frac{TP}{TP + FP}. \tag{13}$$

- Recall Rate (RR):

$$\text{RR} = \frac{TP}{TP + FN}. \tag{14}$$

- F-Measure (F1-Score):

$$F1 = \frac{2 \cdot PR \cdot RR}{PR + RR}. \tag{15}$$

- Loss (L):

$$\mathcal{L} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i). \tag{16}$$

In Equation 16, n is the number of classes, y_i is the true class label, and $\hat{y}_i$ is the predicted probability for class i. A good model should yield a loss value close to 0.

Table 8. Comparison of contemporary intrusion detection models with the proposed approach.

Author(s)	Model	Dataset			ACC (%)	FPR	Remarks
		UNSW-NB15	CICIDS2017	Other			
Ajagbe *et al.* [1]	3-layer CNN + BatchNorm + Dropout	√	-	-	98.7	1.5	Tailored for IoT traffic patterns; inference latency ~0.015s per sample
Tabassum *et al.* [28]	4-layer Autoencoder Neural Network	-	-	√	97.8	2.5	Reconstruction error thresholding; model footprint ≈ 1.2 MB
Aljehane [29]	Deep-stacked Autoencoder (5 hidden layers)	-	√	-	98.9	1.2	Hyperparameters tuned via grid search; training time ≈ 6 h on a single GPU
Kaur *et al.* [32]	Bayesian Deep Learning + Conv. Feature Eng.	-	-	√	99.2	0.8	Provides Bayesian uncertainty estimates; robust under concept drift
Alsaade and Al-Adhaileh [36]	6-layer Deep Autoencoder	-	-	√	99.4	0.6	Demonstrates resilience to sensor noise; optimized for vehicular CPS networks
Ortega-Fernandez *et al.* [37]	5-layer Deep Autoencoder	-	-	√	99.5	0.4	Real-time DDoS detection; low computational overhead
Our Model	Denoising Autoencoder + Deep Neural Network	√	√	-	98.95 UNSW-NB15	0.034	Consistent cross-dataset performance; strong balance between sensitivity and specificity
					99.99 CICIDS2017	0.0013	Robust latent representation; minimal overfitting; FPR reduced by two orders of magnitude

Experimental Testbed

We implemented our model using Python 3.6.7 and TensorFlow on the Ubuntu 18.04 64-bit operating system. TensorFlow is a symbolic math library designed for machine learning applications such as neural networks. It expresses computations as stateful dataflow graphs, enabling high-performance numerical computations on Central Processing Units (CPUs) and Graphics Processing Units (GPUs) [81, 82].

Table 2 lists the system properties of the machine used for implementation.

4. Experimental results and discussion

In this section, the results of experimentation are discussed. An important factor of the proposed model is the tuning of the hyperparameters to overcome the overfitting or underfitting of the training data to generalise on the test samples. In particular, we use the ReLU activation function in the hidden layers of the autoencoder and deep feedforward neural network, and an Adadelta optimizer with a learning rate of 1. The performance of our model is then benchmarked against extant approaches for detecting attacks on CPSs. Our findings are very promising for the accurate detection of attacks on cyber-physical systems.

4.1. Hyperparameter configuration and tuning

The hyperparameters for the optimal performance of our model were chosen using a random search process. Through several experiments, we choose a depth of 3 hidden layers for our model with the first layer having 250 neurons, the second layer with 200 neurons, and the third layer with 100 neurons. At

the hidden layers, the model uses the ReLU activation function to learn the compressed representation of the attack data. The learning process at the hidden layers is described as follows:

Let x denote the input vector to the model, and y denote the output of the model. The network is structured such that:

$$h_1 = \text{ReLU}(W_1 x + b_1), \tag{17}$$

$$h_2 = \text{ReLU}(W_2 h_1 + b_2), \tag{18}$$

$$h_3 = \text{ReLU}(W_3 h_2 + b_3), \tag{19}$$

where $W_1 \in \mathbb{R}^{250\times d}$, $W_2 \in \mathbb{R}^{200\times 250}$, and $W_3 \in \mathbb{R}^{100\times 200}$ denote the weight matrices for the three hidden layers, and d is the dimension of the input. b_1, b_2, and $b_3 \in \mathbb{R}^n$ represent the biases for the three layers. Also, the function $\text{ReLU}(z) = \max(0, z)$ is the element-wise activation function applied in the hidden layers of the network.

We computed the output of the model based on Equation 20:

$$\hat{y} = W_4 h_3 + b_4, \tag{20}$$

where W_4 is the weight matrix that connects the final hidden layer to the output layer. The output bias is b_4.

The output of ReLU is sparsely activated; thus, for all negative inputs, the output is zero. This helps the function to converge faster so it can be trained and run in a relatively short time. We optimised ReLU using the Adadelta optimizer. The Adadelta optimizer is an adaptive gradient descent algorithm. In other words, it is a per-parameter learning rate technique for gradient descent [83, 84]. This optimizer has negligible computational overhead and does not require manual tuning of the learning rate. Consequently, it is efficient when applied to noisy gradient information, diverse network architectures and datasets, as well as different combinations of hyperparameters. A description of the Adadelta optimizer is given in [84].

The deep feedforward network is optimised using the default values of the Adadelta optimizer, which consists of a learning rate of 1 and a decay factor of 0.95. However, this learning rate decreases gradually over different training iterations. In this sense, Adadelta performs smaller changes on parameters that are frequently updated and larger changes on parameters that are not frequently updated [83].

4.2. Model training and testing

To set the number of epochs for the training and testing of the model, the number of epochs was initialised to 100. Early stopping was then used when fitting the model to determine at which epoch the model converged. The monitored quantity, in this case, is the validation loss.

Validation loss is similar to training loss, although it is not used to update the weights in the deep neural network. It is calculated by running the network forward over inputs $\hat{x}_i$, and then comparing the network outputs $\bar{y}_k$ with the ground truth values y_k. The validation loss is calculated using the loss function in Equation (21) as defined in [85]:

$$J = \frac{1}{N}\sum_{k=1}^{N} L(\bar{y}_k, y_k), \tag{21}$$

In Equation (21), J is the validation loss, N is the total number of samples, and L is the individual loss function based on the difference between the predicted and target values. We used validation loss as the monitor for early stopping in our model to measure how well the model is generalising on unseen samples. This is very significant for classifying the modelled attack types. Furthermore, a patience value of 10 epochs is used to specify when no improvement is made on the monitored quantity in order to stop the training of the model. Then, the model weights are restored from the epoch that produced the best value of the monitored quantity.

The weights were randomly initialised, and ReLU with backpropagation gradient descent, optimised with Adadelta, changed the weights from random to regular using the patterns extracted from the data. A loss function is used at the output layer to measure the model's classification performance. The output of the loss function is a probability value in the range [0, 1]. The loss decreased significantly during the training and testing of the model, with values close to 0 as shown in Tables 3 and 4.

4.2.1. Discussion of results

We recorded the performance of our model for the training, validation, and testing phases with the performance metrics discussed in Section 3.2.1. The experiments were based on samples extracted from 2 datasets, which are the Benign, DDoS and DoS samples from the CICIDS2017 dataset and the Benign, DoS, Shellcode and Worm samples from the UNSW_NB15 dataset. The results based on the CICIDS2017 samples are shown in Table 3 while the confusion matrix is presented in Table 4. Similarly, the results of experiments based on the UNSW_NB15 dataset are given in Table 5 and the confusion matrix for these samples is shown in Table 6. Using the data from Tables 3 and 5, as well as the confusion matrix of Tables 4 and 6, we were able to demonstrate the possibility of detecting attacks on CPSs using deep learning. Our model demonstrated very stable performance for all attack samples and was able to learn the representations of these samples to generalise on the test data.

From Table 3, our model achieved a very high accuracy of close to 100% and very low loss for the benign and attack classes over 13 epochs after restoring model weights from the end of the best epoch. The confusion matrix for the CICIDS2017 samples as shown in Table 4 further emphasises the performance of our model. The data in Table 4 shows that only 1 instance of the Benign samples is misclassified as DDoS. Similarly, only 3 samples of DoS are misclassified while only 2 samples of DoS are misclassified by our model. Other than this, the total misclassifications were only 0.028% for the samples in the CICIDS2017 dataset. Testing the accuracy of our model against the test samples in the UNSW_NB15 dataset, we observed that our model achieved very high accuracy as well after 60 epochs. The values of the evaluation metrics used in assessing the model performance as given in Table 5.

The 4 extracted samples from the UNSW_NB15 dataset were detected with high accuracy for each attack type as shown in Table 5. For instance, the samples were detected with the

highest accuracy of 9.91% for the worm samples and the lowest accuracy of 93.15% for the normal connection. The loss was also low. The confusion matrix for these samples is shown in Table 6.

From the values of the confusion matrix in Table 6, 74 of the DoS samples, 76 of the normal samples, 61 of the shellcode samples, and 5 of the worm samples were misclassified. This is a promising performance by our model as compared to similar models. The overall performance of our model for both datasets is represented in Table 7.

Using the data in Table 7, we observed that our choice of hyperparameters was able to overcome the overfitting (low bias and high variance) and underfitting (high bias and low variance) of the model, and as such could generalise on the unseen samples from the 2 datasets. Furthermore, we visualised the accuracy and loss of the model and the curves produced are depicted in Figures 2 and 3.

In Figures 2 and 3, we plotted the detection accuracy and loss of the model for samples from both datasets. We observed that the model was able to learn from the samples during the training phase, and classified the test samples to a high degree of accuracy. In the same sense, the gradient of the loss function concerning the weights of the model was computed and backpropagated layer-wise to produce highly accurate classifications. This is indicated by the absence of significant deviations between the training and test curves of the model.

Our model exhibits consistently superior performance on both datasets. On the CICIDS2017 dataset, the model attained an overall classification accuracy of 99.99% with precision, recall, and F1-score values approaching 100%. While the performance measures on the UNSW-NB15 dataset were somewhat reduced, they nonetheless reflected very high levels, with an overall accuracy of 98.95% and false positive rates maintained below 3.5% (Table 7). The findings substantiate the capability of the model to generalise well over datasets with varying traffic profiles and attack patterns.

4.3. Comparative analysis with state-of-the-art models

To contextualise the performance of our proposed denoising autoencoder-based dimensionality reduction fused with a deep neural network classifier, we conduct a thorough comparison against leading intrusion detection frameworks documented in recent literature. Table 8 summarizes the architectural features, benchmark datasets, detection accuracy, false positive rates (FPR), and key implementation notes for each model.

A false positive rate of less than 0.05% is crucial in cyber physical systems where false alarms will lead to unjustified shutdown or remedial action in industrial controllers. The FPR of our model, 0.0013% on CICIDS2017 and 0.034% on UNSW NB15, ensures continuity of operation and avoids unnecessary intervention. Furthermore, the robustness and stability of our model show a standard deviation in accuracy below 0.02% and little variance in FPR. This robustness is due to the fact that the denoising autoencoder can learn a compressed noise robust latent space, and it offers a reliable safeguard against overfitting, which is commonly observed for classifiers trained on raw feature vectors. Therefore, the model shows excellent generalisation to unknown intrusion patterns, validating the ability to deploy it in industrial grade CPS settings.

5. Conclusion

In this work, we studied the combination of a deep denoising autoencoder and deep feedforward neural network to detect adversarial attacks on cyber-physical systems. The expansion in attack surfaces is a significant challenge t o cybersecurity professionals and practitioners in academia and industry. Consequently, new models, especially those that exploit the power of deep learning are relevant. In this context, our model can learn the representation of attack samples extracted from two datasets (CICIDS2017 and UNSW_NB15 datasets) to detect common network intrusions on cyber-physical systems. We used a denoising autoencoder in the unsupervised pre-training phase of our model to avoid overfitting, high computational overheads and inevitably extensive training time that may lead to a complex model. The combination of techniques presented in our model is novel for detecting adversarial attacks on cyber-physical systems and the results obtained were very promising against benchmarked approaches. In future work, we intend to tune the model for predicting attacks in cyber-physical systems at the early stages to control the damage from such attacks.

Data availability

The data that support the findings of this study are openly available at https://www.unb.ca/cic/datasets/ids-2017.html for the CICIDS2017 dataset and at https://research.unsw.edu.au/projects/unswnb15-dataset for the UNSW-NB15 dataset.

References

[1] S. A. Ajagbe, J. B. Awotunde & H. Florez, "Ensuring intrusion detection for IoT services through an improved CNN", SN Computer Science **5** (2023) 49. https://doi.org/10.1007/s42979-023-02448-y.

[2] J. B. Awotunde, Y. J. Oguns, K. A. Amuda, N. Nigar, T. A. Adeleke, K. M. Olagunju & S. A. Ajagbe, "Cyber-physical systems security: analysis, opportunities, challenges, and future prospects", in *Blockchain for Cybersecurity in CyberPhysical Systems*, pp. 21–46, 2023. https://link.springer.com/chapter/10.1007/978-3-031-25506-9_2.

[3] R. Alguliyev, Y. Imamverdiyev & L. Sukhostat, "Cyber-physical systems and their security issues", Computers in Industry **100** (2018) 212. https://www.sciencedirect.com/science/article/abs/pii/S0166361517304244?via%3Dihub.

[4] A. O. de Sá, L. F. R. da Costa Carmo & R. C. Machado, "Covert attacks in cyberphysical control systems", IEEE Trans. Industrial Informatics **13** (2017) 1641. http://dx.doi.org/10.1109/TII.2017.2676005.

[5] A. Hoehn and P. Zhang, *Detection of covert attacks and zero dynamics attacks in cyberphysical systems*, Proc. 2016 American Control Conf. (ACC), 2016 pp. 302–307. https://doi.org/10.1109/ACC.2016.7524819.

[6] A. Humayed, J. Lin, F. Li & B. Luo, "Cyberphysical systems security–A survey", *IEEE Internet Things J.* **4** (2017) 1802. https://doi.org/10.1109/JIOT.2017.2703172.

[7] S. Ntalampiras, "Automatic identification of integrity attacks in cyber-physical systems", Expert Systems with Applications, **58** (2016) 164. https://doi.org/10.1016/j.eswa.2016.04.006.

[8] F. Pasqualetti, F. Dörfler & F. Bullo, "Attack detection and identification in cyberphysical systems", IEEE Trans. Automatic Control **58** (2013) 2715. https://doi.org/10.1109/TAC.2013.2266831.

[9] A. O. Bajeh, M. O. Olaoye, F. E. UsmanHamza, I. S. Olatinwo, P. Ogirima, Sadiku & A. B. Sakariyah, "An adaptive neuro-fuzzy inference system for multinomial malware classification", J. Nigerian Soc. Phys. Sci. **6** (2025) 2172. https://doi.org/10.46481/jnsps.2025.2172.

[10] H. A. Kholidy and A. Erradi, "VHDRA: a vertical and horizontal intelligent dataset reduction approach for cyber-physical power aware intrusion detection systems", Security and Communication Networks (2019) 1283081. https://doi.org/10.1155/2019/6816943.

[11] G. Bernieri, E. Miciolino, F. Pascucci & R. Setola, "Monitoring system reaction in cyber-physical testbed under cyber-attacks", Computers & Electrical Engineering **59** (2017) 86. https://doi.org/10.1016/j.compeleceng.2017.02.010.

[12] J. Wurm, Y. Jin, Y. Liu, S. Hu, K. Heffner, F. Rahman & M. Tehranipoor, "Introduction to cyber-physical system security: a cross-layer perspective", *IEEE Trans. Multi-Scale Computing Systems* **3** (2016) 215. https://doi.org/10.1109/TMSCS.2016.2569446.

[13] A. G. Busygin, A. S. Konoplev & D. P. Zegzhda, "Providing stable operation of self-organizing cyber-physical system via adaptive topology management methods using blockchain-like directed acyclic graph", Automatic Control and Computer Sciences **52** (2018) 1080. https://doi.org/10.3103/S0146411618080059.

[14] F. Nizam, S. Chaki, S. Al Mamun & M. S. Kaiser, *Attack detection and prevention in the cyber physical system*, in Proc. 2016 Int. Conf. Computer Communication and Informatics (ICCCI), Jan. 2016, pp. 1–6. https://doi.org/10.1109/ICCCI.2016.7480022.

[15] A. Y. Nur and M. E. Tozal, *Defending cyber-physical systems against DoS attacks*, in Proc. 2016 IEEE Int. Conf. Smart Computing (SMARTCOMP), May 2016, pp. 1–3. https://doi.org/10.1109/SMARTCOMP.2016.7501685.

[16] C. Zimmer, B. Bhat, F. Mueller & S. Mohan,*Time-based intrusion detection in cyber-physical systems*, in Proc. 1st ACM/IEEE Int. Conf. Cyber-Physical Systems, Apr. 2010, pp. 109–118. https://doi.org/10.1145/1795194.179521.

[17] S. Karnouskos, *Stuxnet worm impact on industrial cyber-physical system securit*, in Proc. IECON 2011 - 37th Annu. Conf. IEEE Ind. Electronics Society, Nov. 2011, pp. 4490–4494. https://doi.org/10.1109/IECON.2011.6120048.

[18] P. Kreimel, O. Eigner & P. Tavolato, *Anomaly-based detection and classification of attacks in cyber-physical systems*, in Proc. 12th Int. Conf. Availability, Reliability and Security (ARES), Aug. 2017, pp. 1–6. https://doi.org/10.1145/3098954.3103155.

[19] C. Cheh, K. Keefe, B. Feddersen, B. Chen, W. G. Temple & W. H. Sanders,*Developing models for physical attacks in cyber-physical systems*, in Proc. 2017 Workshop on Cyber-Physical Systems Security and Privacy, Nov. 2017, pp. 49–55. https://doi.org/10.1145/3140241.3140249.

[20] A. Khaled, S. Ouchani, Z. Tari & K. Drira, "Assessing the severity of smart attacks in industrial cyber-physical systems", ACM Transactions on Cyber-Physical Systems, **5** (2020) 1. https://doi.org/10.1145/3422369.

[21] L. S. Lü, X. Jin, L. Ding & Q. Tan, "Adaptive sliding-mode control of a class of disturbed cyber–physical systems against actuator attacks", *Computers & Electrical Engineering* **96** (2021) 107492. https://doi.org/10.1016/j.compeleceng.2021.107492.

[22] X. Fang, M. Xu, S. Xu & P. Zhao, "A deep learning framework for predicting cyber attacks rates", EURASIP Journal on Information Security **2019** (2019) 11. https://doi.org/10.1186/s13635-019-0090-6.

[23] A. E. Ibor, F. A. Oladeji, O. B. Okunoye & O. O. Ekabua, "An improved cyberattack prediction technique with intelligent clustering and deep neural network", FUW Trends in Science and Technology Journal **5** (2020) 15. http://www.ftstjournal.com/Digital%20Library/51%20Article%203.php.

[24] A. E. Ibor, F. A. Oladeji, O. B. Okunoye & O. O. Ekabua, "Conceptualisation of cyberattack prediction with deep learning", Cybersecurity **3** (2020) 14. https://doi.org/10.1186/s42400-020-00053-7.

[25] A. Ibor, F. Oladeji, O. Okunoye & K. Abdulsalam, *Network intrusion prediction model based on bio-inspired hyperparameter search*, in Proc. Int. Conf. Electrical, Computer and Energy Technologies (ICECET), pp. 1–5, Dec. 2021. https://ieeexplore.ieee.org/abstract/document/9698491/.

[26] U. C. Obini, C. Jeremiah & S. A. Igwe, "Development of a machine learning based fileless malware filter system for cybersecurity", J. Nigerian Soc. Phys. Sci. **5** (2024) 2192. https://doi.org/10.46481/jnsps.2024.2192.

[27] P. U. Emmoh & T. Moses, "A feature selection and scoring scheme for dimensionality reduction in a machine learning task", J. Nigerian Soc. Phys. Sci. **6** (2025) 2273. https://doi.org/10.46481/jnsps.2025.2273.

[28] T. Tabassum, O. Toker & M. R. Khalghani, "Cyber–physical anomaly detection for inverter-based microgrid using autoencoder neural network", *Applied Energy*, vol. 355, 122283, 2024. https://doi.org/10.1016/j.apenergy.2023.122283.

[29] N. O. Aljehane, "A secure intrusion detection system in cyberphysical systems using a parameter-tuned deep-stacked autoencoder", Computers, Materials & Continua **68** (2021) 3915. https://doi.org/10.32604/cmc.2021.017905.

[30] Q. E. U. Haq, M. Imran, K. Saleem, T. Zia & J. Al Muhtadi, "Review on variants of restricted Boltzmann machines and autoencoders for cyber-physical systems", in *Internet of Things Security and Privacy*, CRC Press, 2023, pp. 188–207. https://doi.org/10.1201/9781003199410.

[31] N. Rajathi, G. Saritha & V. J. Ramya, *Adaptive intrusion detection in cyberphysical systems using reinforcement learning-based autoencoders*, Proc. Int. Conf. Integrated Intelligence and Communication Systems (ICIICS), Nov. 2024, pp. 1–7. https://ieeexplore.ieee.org/abstract/document/10859561.

[32] D. Kaur, A. Anwar, I. Kamwa, S. Islam, S. M. Muyeen & N. Hosseinzadeh, "A Bayesian deep learning approach with convolutional feature engineering to discriminate cyber-physical intrusions in smart grid systems", IEEE Access **11** (2023) 18910. https://doi.org/10.1109/ACCESS.2023.3247947.

[33] R. R. Nuiaa Al Ogaili, M. I. Mahdi, A. F. Neamah, S. A. A. Alradha Alsaidi, A. H. Alsaeedi, Z. A. Dashoor & S. Manickam, "PhishNetVAE cybersecurity approach: an integrated variational autoencoder and deep neural network approach for enhancing cybersecurity strategies by detecting phishing attacks", Int. J. Intelligent Engineering & Systems **18** (2025) 123. .

[34] N. Sugunaraj & P. Ranganathan, *Applications for autoencoders in power systems*, in Proc. North American Power Symposium (NAPS), Oct. 2024, pp. 1–7. https://ieeexplore.ieee.org/abstract/document/10741685.

[35] A. Kousar, S. Ahmed, A. Altamimi, S. M. Kim & Z. A. Khan, *Deep learning-based dimensionality reduction for anomaly detection in smart grids*, in Proc. Int. Conf. Information and Communication Technology Convergence (ICTC), Oct. 2023, pp. 71–75. https://doi.org/10.1109/ICTC58733.2023.10393285.

[36] F. W. Alsaade & M. H. Al-Adhaileh, "Cyber attack detection for self-driving vehicle networks using deep autoencoder algorithms", Sensors **23** (2023) 4086. https://doi.org/10.3390/s23084086.

[37] I. Ortega-Fernandez, M. Sestelo, J. C. Burguillo & C. Piñón-Blanco, "Network intrusion detection system for DDoS attacks in ICS using deep autoencoders", Wireless Networks **30** (2024) 5059. https://doi.org/10.1007/s11276-022-03214-3.

[38] V. K. Kukkala, S. V. Thiruloga & S. Pasricha, "Real-time intrusion detection in automotive cyber-physical systems with recurrent autoencoders", in *Machine Learning and Optimization Techniques for Automotive Cyber-Physical Systems*, Springer International Publishing, Cham, 2023, pp. 317–347. https://doi.org/10.1007/978-3-031-28016-0_10.

[39] K. Saranya & A. Valarmathi, "A multilayer deep autoencoder approach for cross layer IoT attack detection using deep learning algorithms", Scientific Reports **15** (2025) 10246. https://doi.org/10.1038/s41598-025-93473-9.

[40] F. Harrou, B. Bouyeddou, A. Dairi & Y. Sun, "Exploiting autoencoder-based anomaly detection to enhance cybersecurity in power grids", Future Internet **16** (2024) 184. https://doi.org/10.3390/fi16060184.

[41] J. Zhang, L. Pan, Q. L. Han, C. Chen, S. Wen & Y. Xiang, "Deep learning based attack detection for cyber-physical system cybersecurity: a survey", IEEE/CAA Journal of Automatica Sinica **9** (2022) 377. https://doi.org/10.1109/JAS.2021.1004261.

[42] G. D'Angelo & F. Palmieri, "A stacked autoencoder-based convolutional and recurrent deep neural network for detecting cyberattacks in interconnected power control systems", International Journal of Intelligent Systems **36** (2021) 7080. https://doi.org/10.1002/int.22581.

[43] M. J. Zideh, M. R. Khalghani & S. K. Solanki, "An unsupervised adversarial autoencoder for cyber attack detection in power distribution grids", Electric Power Systems Research **232** (2024) 110407. https://ui.adsabs.harvard.edu/abs/2024EPSR..23210407Z/abstract.

[44] Z. Ma, G. Mei & F. Piccialli, "Deep Learning for Secure Communication

in Cyber-Physical Systems",IEEE Internet of Things Magazine **5** (2022) 63. Dhttps://ieeexplore.ieee.org/abstract/document/9889272.

[45] B. Roshanzadeh, J. Choi, A. Bidram & M. Martínez-Ramón, "Multivariate time-series cyberattack detection in the distributed secondary control of AC microgrids with convolutional neural network autoencoder ensemble", Sustainable Energy, Grids and Networks **38** (2024) 101374. https://doi.org/10.1016/j.segan.2024.101374.

[46] M. Chen, X. Shi, Y. Zhang, D. Wu & M. Guizani, "Deep feature learning for medical image analysis with convolutional autoencoder neural network", IEEE Transactions on Big Data **7** (2021) 750. https://doi.org/10.1109/TBDATA.2017.2717439.

[47] X. Jin, W. M. Haddad & T. Yucelen, "An adaptive control architecture for mitigating sensor and actuator attacks in cyber-physical systems", IEEE Transactions on Automatic Control **62** (2017) 6058. https://doi.org/10.1109/TAC.2017.2652127.

[48] Md. Shamsul Huda, Md. Suruz Miah, Mohammad Mehedi Hassan, Md. Rafiqul Islam, John Yearwood, Majed A. Alrubaian & Ahmad Almogren, "Defending unknown attacks on cyber-physical systems by semisupervised approach and available unlabeled data", Information Sciences **379** (2017) 211. https://doi.org/10.1016/j.ins.2016.09.041.

[49] O. A. Beg, T. T. Johnson & A. Davoudi, "Detection of false-data injection attacks in cyber-physical DC microgrids", IEEE Transactions on Industrial Informatics **13** (2017) 2693. https://doi.org/10.1109/TII.2017.2656905.

[50] S. R. Chhetri, A. Canedo & M. A. Al Faruque, *Kcad: kinetic cyber-attack detection method for cyber-physical additive manufacturing systems*, in Proc. 35th Int. Conf. Computer-Aided Design (ICCAD), San Jose, CA, USA, Nov. 2016, pp. 1–8. https://doi.org/10.1145/2966986.2967050.

[51] H. Sadreazami, A. Mohammadi, A. Asif & K. N. Plataniotis, "Distributed-graph-based statistical approach for intrusion detection in cyber-physical systems", IEEE Transactions on Signal and Information Processing over Networks **4** (2017) 137. https://doi.org/10.1109/TSIPN.2017.2749976.

[52] A. Ferdowsi & W. Saad, *Generative adversarial networks for distributed intrusion detection in the internet of things*, in Proc. 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, Dec. 2019, pp. 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9014102.

[53] Y. Hong, U. Hwang, J. Yoo & S. Yoon, "How generative adversarial networks and their variants work: an overview", ACM Computing Surveys **52** (2019) 1. https://doi.org/10.1145/3301282.

[54] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng & F. Y. Wang, "Generative adversarial networks: introduction and outlook", IEEE/CAA Journal of Automatica Sinica **4** (2017) 588. https://doi.org/10.1109/JAS.2017.7510583.

[55] S. Thakur, A. Chakraborty, R. De, N. Kumar & R. Sarkar, "Intrusion detection in cyberphysical systems using a generic and domain specific deep autoencoder model", Computers & Electrical Engineering **91** (2021) 107044. https://doi.org/10.1016/j.compeleceng.2021.107044.

[56] I. Sharafaldin, A. Gharib, A. H. Lashkari & A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset", Software Networking **2018** (2018) 177. https://www.researchgate.net/publication/318286637_Towards_a_Reliable_Intrusion_Detection_Benchmark_Dataset.

[57] I. Sharafaldin, A. H. Lashkari & A. A. Ghorbani, *Toward generating a new intrusion detection dataset and intrusion traffic characterization*, in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), Funchal, Madeira, Portugal, Jan. 2018, pp. 108–116. https://doi.org/10.5220/0006639801080116.

[58] I. Sharafaldin, A. Habibi Lashkari & A. A. Ghorbani, "A detailed analysis of the CICIDS2017 data set," in *Information Systems Security and Privacy*, P. Mori, S. Furnell & O. Camp (Eds.), Communications in Computer and Information Science, vol. 977, Springer, Cham, 2019, pp. 172–188. https://doi.org/10.1007/978-3-030-25109-3_9.

[59] R. Panigrahi & S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems", International Journal of Engineering and Technology **7** (2018) 479. https://www.sciencepubco.com/index.php/ijet/article/view/22797.

[60] T. Chadza, K. G. Kyriakopoulos & S. Lambotharan, *Contemporary sequential network attacks prediction using hidden Markov model*, Proc. 17th Int. Conf. Privacy, Security and Trust (PST), Montreal, Canada, July 2019, pp. 1–3. https://doi.org/10.1109/PST47121.2019.8949035.

[61] O. Faker & E. Dogdu, *Intrusion detection using big data and deep learning techniques*, Proc. ACM Southeast Conf., Huntsville, AL, USA, Apr. 2019, pp. 86–93. https://doi.org/10.1145/3299815.3314439.

[62] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat & S. Venkatraman, "Deep learning approach for intelligent intrusion detection system", IEEE Access **7** (2019) 41525. https://doi.org/10.1109/ACCESS.2019.2895334.

[63] N. Moustafa & J. Slay, , "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)", 2015 Military Commun. & Inf. Syst. Conf. (MilCIS), Canberra, Australia, Nov. 2015, pp. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942.

[64] T. Janarthanan & S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets", 2017 IEEE 26th Int. Symp. Industrial Electronics (ISIE), Edinburgh, UK, June 2017, pp. 1881–1886. https://doi.org/10.1109/ISIE.2017.8001537.

[65] N. Moustafa & J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set", Inf. Security J.: A Global Perspective **5** (2016) 18. https://doi.org/10.1080/19393555.2015.1125974.

[66] Q. Meng, D. Catchpoole, D. Skillicom & P. J. Kennedy, *Relational autoencoder for feature extraction*, Proc. 2017 Int. Joint Conf. Neural Networks (IJCNN), Anchorage, AK, USA, May 2017, pp. 364–371. https://doi.org/10.1109/IJCNN.2017.7965877.

[67] S. Li, J. Kawale & Y. Fu, *Deep collaborative filtering via marginalized denoising auto-encoder*, Proc. 24th ACM Int. Conf. Information and Knowledge Management (CIKM), Melbourne, Australia, Oct. 2015, pp. 811–820. https://doi.org/10.1145/2806416.2806527.

[68] C. Lu, Z. Y. Wang, W. L. Qin & J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification", Signal Process. **130** (2017) 377. https://doi.org/10.1016/j.sigpro.2016.07.028.

[69] X. Lu, Y. Tsao, S. Matsuda & C. Hori, "Speech enhancement based on deep denoising autoencoder", in *Interspeech*, Lyon, France, Aug. 2013, pp. 436–440. https://cir.nii.ac.jp/crid/1360292619441675904.

[70] C. Xing, L. Ma & X. Yang, "Stacked denoise autoencoder based feature extraction and classification for hyperspectral images", J. Sensors (2016) 2975718. http://dx.doi.org/10.1155/2016/3632943.

[71] C. C. Aggarwal, *Neural networks and deep learning*, Springer, Cham, 2018. https://doi.org/10.1007/978-3-031-29642-0.

[72] I. Goodfellow, Y. Bengio & A. Courville, *Deep learning*, MIT Press, Cambridge, MA, USA, 2016. https://doi.org/10.4258/hir.2016.22.4.351.

[73] D. Ding, Q. L. Han, Y. Xiang, X. Ge & X. M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems", Neurocomputing **275** (2018) 1674. https://doi.org/10.1016/j.neucom.2017.10.009.

[74] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg & R. Candell, "A survey of physics-based attack detection in cyber-physical systems", ACM Comput. Surveys **51** (2018) 1. https://doi.org/10.1145/3203245.

[75] S. Patro & K. K. Sahu, "Normalization: a preprocessing stage", arXiv preprint arXiv:1503.06462, 2015. https://arxiv.org/abs/1503.06462.

[76] M. Chen, X. Shi, Y. Zhang, D. Wu & M. Guizani, "Deep feature learning for medical image analysis with convolutional autoencoder neural network", IEEE Trans. Big Data **7** (2021) 750. https://doi.org/10.1109/TBDATA.2017.2717439.

[77] Y. Wang, H. Yao & S. Zhao, "Auto-encoder based dimensionality reduction", Neurocomputing **184** (2016) 232. https://doi.org/10.1016/j.neucom.2015.08.104.

[78] W. Wang, Y. Huang, Y. Wang & L. Wang, *Generalized autoencoder: a neural network framework for dimensionality reduction*, in Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, June 2014, pp. 490–497. https://doi.org/10.1109/CVPRW.2014.79.

[79] M. Yousefi-Azar, V. Varadharajan, L. Hamey & U. Tupakula, "Autoencoder-based feature learning for cyber security applications", 2017 Int. Joint Conf. Neural Networks (IJCNN), Anchorage, AK, USA, May 2017, pp. 3854–3861. https://doi.org/10.1109/IJCNN.2017.7966342.

[80] A. F. Agarap, "Deep learning using rectified linear units (ReLU)", arXiv preprint arXiv:1803.08375, 2018. https://arxiv.org/abs/1803.08375.

[81] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin,

S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu & X. Zheng, *TensorFlow: A System for Large-Scale Machine Learning*, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, Nov. 2–4, 2016, pp. 265–283. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[82] B. Ramsundar & R. B. Zadeh, *TensorFlow for Deep Learning: From Linear Regression to Reinforcement Learning*, O'Reilly Media, Sebastopol, CA, USA, 2018. https://dl.acm.org/doi/abs/10.5555/3235300.

[83] Y. Wang, J. Liu, J. Mišić, V. B. Mišić, S. Lv & X. Chang, "Assessing optimizer impact on DNN model sensitivity to adversarial examples", IEEE Access **7** (2019) 152766. https://doi.org/10.1109/ACCESS.2019.2948658.

[84] M. D. Zeiler, "ADADELTA: an adaptive learning rate method", arXiv preprint arXiv:1212.5701, 2012. https://arxiv.org/abs/1212.5701.

[85] J. Hartford, G. Lewis, K. Leyton-Brown & M. Taddy, *Deep IV: A Flexible Approach for Counterfactual Prediction*, Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, Aug. 6–11, 2017, pp. 1414–1423. http://proceedings.mlr.press/v70/hartford17a.html.