electronics

MDPI

Article

Deep Learning of Sensor Data in Cybersecurity of Robotic Systems: Overview and Case Study Results

Wojciech Szynkiewicz , Ewa Niewiadomska-Szynkiewicz * and Kamila Lis

Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland; wojciech.szynkiewicz@pw.edu.pl (W.S.); kamila.lis@nask.pl (K.L.)
* Correspondence: ewa.szynkiewicz@pw.edu.pl; Tel.: +48-884-260-036

Abstract: Recent technological advances have enabled the development of sophisticated robotic and sensor systems monitored and controlled by algorithms based on computational intelligence. The deeply intertwined and cooperating devices connected to the Internet and local networks, usually through wireless communication, are increasingly used in systems deployed among people in public spaces. The challenge is to ensure that physical and digital components work together securely, especially as the impact of cyberattacks is significantly increasing. The paper addresses cybersecurity issues of mobile service robots with distributed control architectures. The focus is on automatically detecting anomalous behaviors possibly caused by cyberattacks on onboard and external sensors measuring the robot and environmental parameters. We provide an overview of the methods and techniques for protecting robotic systems. Particular attention is paid to our technique for anomaly detection in a service robot's operation based on sensor readings and deep recurrent neural networks, assuming that attacks result in the robot behaving inconsistently. The paper presents the architecture of two artificial neural networks, their parameters, and attributes based on which the potential attacks are identified. The solution was validated on the PAL Robotics TIAGo robot operating in the laboratory and replicating a home environment. The results confirm that the proposed system can effectively support the detection of computer threats affecting the sensors' measurements and, consequently, the functioning of a service robotic system.

Keywords: cyber-physical system; robotic system; cybersecurity; service robot; sensor data; artificial neural network; deep learning

check for updates

Citation: Szynkiewicz, W.; Niewiadomska-Szynkiewicz, E.; Lis, K. Deep Learning of Sensor Data in Cybersecurity of Robotic Systems: Overview and Case Study Results. *Electronics* **2023**, *12*, 4146. https://doi.org/10.3390/electronics12194146

Academic Editor: Ana Rosa Cavalli

Received: 22 August 2023
Revised: 22 September 2023
Accepted: 3 October 2023
Published: 5 October 2023

1. Introduction

Social and service robots are increasingly involved in our daily lives. Robotic platforms are examples of sophisticated cyber–physical systems (CPS) that integrate physical and cyber components connected through the Internet to execute a given task [1–4]. A robot is a complex machine that is challenging to design and control. The tight coupling of cyber components with physical systems introduces, among other things, challenges in addressing the cybersecurity of social and service robots. Designing attack-resistant systems composed of robots cooperating with other devices, such as sensor networks, is even more challenging. The greater the number of devices and communication links, especially wireless ones, the more possible vulnerabilities and cyberattack vectors there are. It is evident that due to their direct contact with people, it is necessary to protect robotic platforms operating in social spaces against cyberattacks and implement efficient security mechanisms. Unfortunately, cybersecurity is often a lower priority since it adds even more complexity to building robot systems. Typical industrial robots do not require direct interaction with the outside, especially not with people. The focus when building robot systems is generally on making them safe, reliable, and efficient. However, the current trend towards cyber–physical systems built of cooperating robots and sensors, both connected to the Internet, faces all the same threats as computer networks. In general, most of today's robot systems are vulnerable to attacks because they do not implement mechanisms and tools successfully used in computer networks, such as firewalls, encrypted communication, etc.

A robotic system consists of many components that can be affected by multiple vulnerabilities. These include inadequately secured sensors, network connections, remote control applications, software, operating systems, unencrypted backups, and open physical ports. Cyberattacks are also facilitated by the specific characteristics of this type of system, i.e., a distributed control structure, including connection to cloud computing, autonomous devices, and wireless communication. An attack is generally considered as interference with sensors, effectors, or controllers that adversely affects the robot's execution of a task or the safety of the robot and the user. It can include denial of service (DoS) attacks, command and false data injection, disruption of communication with the operator, the introduction of delays, and ordering unnecessary calculations that load the processor [5–11]. The result of unauthorized action can be the remote takeover of the robot, using it to spy on the user, access services and sensitive data, run software leading to the robot's destruction, and, in extreme cases, threatening human life.

Nowadays, individual protection of autonomous systems using a simple analysis of transmitted messages needs to be improved. There is a clear need to create new, holistic solutions utilizing a fusion of data from multiple sources, integrating different methods, mechanisms, and algorithms. The volume, quality, reliability, and timeliness of measurements; the states of all components of the robotic system; and the robotic system's environment determine the effectiveness of protection by preventing threats, identifying the type of attack, and responding to their occurrence. The challenge is to increase situational awareness by detecting known threats and recognizing anomalies in robotic systems that may be symptomatic of malicious activities. Addressing this challenge requires advanced methods for distributed monitoring and aggregation capabilities to derive events, measures, and metrics for deeper processing and analysis. Data fusion, advanced statistics, machine learning techniques, and deep learning are often used to correlate a broad range of security contexts for cyberthreat intelligence. A further challenge is to extend the trust and integrity of data and the execution environment by including access control, identity management, and trust mechanisms that ensure high security. When building a secure CPS, it is essential to remember that human beings are in the loop [12]. Therefore, behavioral, social, and human aspects must be included in the engineering process to avoid the risk of neglecting or underestimating security threats.

Finally, the widespread use of Internet of Things (IoT) devices has significantly increased the interest of cybercriminals in IoT systems in recent years. Modern robotic systems often cooperate with IoT networks. Hence, they are exposed to various new forms of attacks. Insecure IoT devices are already exploited and used in massive attacks and rapidly increase the number of victims.

This paper focuses on cybersecurity strategies and methods for protecting cyber–physical systems composed of mobile service robots and sensors [1]. The main contribution is pointing out the problems of designing and developing cybersecurity methods and tools. We first survey different strategies and systems described in the literature. Next, we propose our solution for detecting the abnormal behavior of a mobile service robot. In particular:

- We design and evaluate two deep learning models for the real-time analysis of multimodal data collected periodically by the robot's onboard sensors.
- We develop a prototype of a Service Mobile Robot Intrusion Detection System (SMR_IDS) for detecting anomalies on a real service robot and test it for two different types of attacks (false command injection and sensor spoofing).
- We experimentally investigate how model complexity affects the effectiveness of anomaly detection.

The SMR_IDS is designed for mobile service robots operating in the environment of a typical home with small and cluttered rooms. Due to the direct contact with people and high exposure to cyberattacks, intrusion detection systems should protect all robotic platforms operating in a social space.

The paper is structured as follows. An overview of methods for anomaly and cybersecurity incidents detection is presented in Section 2. We pay attention to techniques implementing machine learning and deep learning. Section 3 describes our approach to intrusion detection for a mobile robot. The results of evaluating our method through testbed implementation are presented and discussed in Section 5. Finally, conclusions are drawn in Section 6.

2. Cybersecurity of Robotic Systems

2.1. Cybersecurity Challenges and Strategies

In recent years, intensive research has been carried out on protecting cyber–physical systems, especially those built with service robots and wireless sensors [2,13–15]. Due to the complex control and decision mechanisms, the need for real-time processing of measurement data and operational decision making in the design and development of secure CPSs is a non-trivial challenge. Overall, strategies and methods for protecting robotic systems draw from the solutions developed for computing networks. Cyber–physical systems have a representation in the physical world, which yields two security-relevant aspects, i.e., (i) robots can be physically manipulated, which can be exploited by an attacker, and (ii) robots can have a significant impact on their workspace and the safety of people and other machines in the surrounding area. The consequences of an attack can be much more significant than in typical computer systems. Therefore, robotic security adds a dimension of physical interaction with humans and other machines to the requirements of general cybersecurity [8,13,16].

Similar to the protection of IT networks, we can distinguish two security strategies:

- Prevention;
- Detection.

Prevention-based security relies on known security methods, such as encryption of communication [17], software updates, machine authentication and authorization, user identification [18], and network access control. Setting a secure default configuration, reporting security errors, and performing audits and penetration tests are also essential. Using a virtual private network (VPN) for the robot's communication with the cloud and verifying the robot control software downloaded by the network is recommended. The requirement to use keys when logging in via secure shell (SSH) on the robot's onboard computer and the provision of a strong user password form the basis for securing the operating system. Work is also underway to look for security vulnerabilities and to secure software systems commonly used in robotics, such as the Robot Operating System (ROS) software platform [19]. It should be noted that the initial version of ROS was not designed with security in mind. The implemented publish–subscribe model allows all system components to register as publishers, subscribers, or both. The absence of mechanisms to restrict registration induces vulnerability to malicious components or messages from the Internet. Securing the API (Application Programming Interface) is a crucial issue. However, it is insufficient because it addresses only the "cyber" part of the cyber–physical system that is a robot. A comprehensive security design on the level of orchestrating mechanisms is required. Unfortunately, implementing prevention-based security mechanisms does not guarantee protection against all types of threats.

Detection-based security relies on reactive techniques for online threats. Two basic approaches can be distinguished:

- Detection and classification of specific attacks;
- Detection of anomalies in the operation of cyber–physical systems.

Intrusion Detection Systems (IDS) commonly analyze events from the system log and local network interfaces. However, they cannot detect attacks based on observed anomalies in robot behavior, nor do they collect and process information from physical sensors. Multimodal sensory signals can help detect a wide range of anomalies. However, the fusion of high-dimensional and heterogeneous modalities is a challenging problem for model-based anomaly detection. Hence, there is a need to build specialized IDS that protect the physical systems of robots cooperating with various sets of sensors.

2.2. Overview of Security Methods and Intrusion Detection Systems

To meet today's robotic systems' security requirements, ensuring efficient and effective response activities to identify and quickly respond to cybersecurity incidents is essential. Ensuring data transmission security is one of the most crucial problems. The quick detection of threats allows for protecting systems from the possibility of damage or destruction. The fundamental problem is the high rate of spreading of attacks and the vast amount of data necessary to process and identify them. Strategies and mechanisms for the effective and rapid detection of threats are the main elements of many defense systems. Many malware detection techniques can be listed. The most important of these are the following:

- Anomaly detection, which involves detecting abnormal behavior, including deviations from typical robot operation, sensor measurements, network traffic loads, etc.;
- Signature analysis, i.e., comparing the content of analyzed data with a set of previously created threat patterns.

Intrusion detection systems collect, analyze, and correlate data to detect cyberattacks. Various techniques for securing robots have been investigated, implemented, and tested through simulation as well as in testbeds constructed of real robots. The effort is put into intra-robot network security in hardware and software components, focusing on ROS and inter-robot network security. Robot Intrusion Detection Systems (R_IDSs) play a significant role in detecting external cyberattacks. Hence, we turn our attention to this type of system. The authors of the paper [2] distinguish three types of intrusion detection systems for CPSs:

- Knowledge-based, in which operations that fit a specific pattern of abnormal behavior are detected based on accumulated knowledge about attacks;
- Behavior-based, in which operations that deviate from the pattern of correct behavior are detected;
- Behavior-specification-based, an extension of the behavior-based model.

Regular behavior learning and anomaly detection can be implemented using mathematical statistics or machine learning. Anomaly detection involves determining to what extent the characteristics of the measured signals deviate from those of the correct behavior model. These characteristics are, e.g., minimum value, maximum value, standard deviation, etc. [20]. The advantage of this approach is universality—the system does not require signatures or specific knowledge of attacks. The disadvantage is the higher false alarm rate and the required learning phase. In the behavior-specification-based method, the correct behavior model is formed by a set of rules defined by the operator based on knowledge of the specific behavior of the system [8]. It allows intrusion detection in systems with limited computing resources.

In summary, recently developed intrusion detection systems for CPSs detect attacks based on the analysis of events in the transport and application layers of the OSI reference communication model and sensor data reporting on the robot's status. Authors' attentions are usually focused on selected attacks and specific robots. Often, for detection attacks, outlier detection methods, a set of defined rules [10], machine learning [7,20], or a model of the dynamics of a physical system [21] are used. Surveys of the prominent cyberattacks on robotic platforms; vulnerabilities, risks, and problems in managing cybersecurity in robotics; and critical cybersecurity countermeasures are provided in [2,6,11]. Special attention is paid to the systems particularly vulnerable to cyberattacks. The authors investigate various cybersecurity techniques and discuss recommendations made by various researchers to improve the level of security. These papers also review security solutions for robotic systems described in recent literature.

Designing systems to protect against cyberattacks requires performing numerous experiments to confirm the effectiveness of the developed technologies. A common approach is conducting preliminary simulator tests for synthetically generated attacks. An example simulation framework to test different solutions and evaluate their effectiveness in detecting threats unique to robotic systems is described in [22,23]. The focus is on cyberattacks targeting communication links and applications fundamental to the cyber–physical systems.

2.2.1. Sensors and Actuators

A typical robotic system comprises robots, a variety of sensors, actuators, and a communication network. It is clear that a robot recognizes its working environment through sensors. It is essential to develop mechanisms that can automatically detect attacks on sensors and actuators, including zero-day attacks. Therefore, sensors should be able to learn their own "normal" operating conditions autonomously and independently detect abnormal conditions. The authors of [20] introduce a model to observe sensor data streams' signal characteristics, including noise level patterns. They compare signatures of each sensor signal and classify signals into normal and abnormal robot behavior characteristics. Finally, they describe an automated threat detection system implemented based on their classification algorithm. Tests of the system conducted on a real robot confirmed the effectiveness of their solution.

2.2.2. Mobile Robots

Intrusion detection systems designed for conventional computer networks are usually unsuitable for mobile robots. They are geared towards attacks of a different nature and only consider some aspects vital to mobile platforms, such as mobility, energy consumption, and limited computing resources. Hence, a significant problem addressed in recent literature is the automatic detection of cyberattacks on mobile robots. Bezemskij et al. describe in [5] a binary classifier for anomaly detection. It performs real-time monitoring of a robotic vehicle and employs Bayesian networks. Data from many sources, i.e., sensors, networks, and equipment, are collected and correlated. The authors claim that experimental results show that this approach is efficient for many types of attacks.

A decision tree-based method for detecting cyberattacks on a small-scale mobile robot is presented in [9,10]. The authors considered cyber and physical features measured by the robot's onboard sensors. The method was evaluated experimentally. Various scenarios involving DoS attacks, false command injection, and two types of malware attacks were considered. The final observation is that involving physical features noticeably improves the detection accuracy and reduces the detection latency.

Wang et al. [24] investigated how to apprehend cyberattacks at the design and implementation stage of an autonomous mobile robotic platform. They designed a directed, feedback-driven fuzzing framework, RoboFuzz, for testing the ROS. In general, the idea of RoboFuzz is to study critical environmental parameters that affect the robot's state transitions. Next, rational but harmful sensor measurements are injected into the control program to compromise the robot. In [24], RoboFuzz is used for the systematic testing of an autonomous mobile robot. The robot's state and its surrounding environment are checked. The authors describe a novel approach to mitigating the impact of RoboFuzz within reasonable additional calculation burden and time. The idea is to use historical records of obstacles in the workspace to detect inconsistent obstacle-resulting data. The untrustworthy sensors are detected. Only reliable measurements are addressed, and the robot is navigated to continue moving and carrying on a planned task. The experiments conducted on a testbed mobile platform confirmed the high rate (93.3%) of detection of the concrete threads imposed on the robot.

The tracking control of mobile robots in the presence of DoS attacks is investigated in [25]. The authors describe a hybrid model considering DoS attack and event-driven control mechanisms. Moreover, they formulate a set of event-triggering conditions to ensure tracking convergence. The presented solution was verified in the laboratory. The experiments were performed on the Amigobot mobile robot and a DoS attack on a wireless network.

In the case of mobile robots, the key is determining the current position of all devices in the workspace. Hence, to disrupt the operation of such a robot, a common attack vector is a vulnerability in its localization system. Guerrero-Higueras et al. [26] applied various supervised learning techniques to detect attacks on localization systems. They tested the effectiveness of detection methods against Denial of Service (DoS) and spoofing cyberattacks. The results of experiments conducted on a wheeled robot with a commercial real-time location system based on ultra-wideband beacons are presented and discussed.

2.2.3. Rescue Robots

Securing rescue robotic platforms is a vital problem, especially for the protection of people. Vuong et al. [16] investigated the influence of cyberattacks on rescue robot operations in an emergency. They focus on identifying physical indicators of an ongoing cyberattack to design more efficient defense mechanisms. The experiments conducted on an Arduino-based robot confirmed that the cyberattack's effects have evident physical features that can improve the robot's robustness against a given type of attack.

2.2.4. Industrial Robots

Autonomous mobile robotic platforms adopted in the industry are often navigated by auto-pilot and rely on sensors and actuators in their workspace. Dash et al. [27] demonstrated the vulnerabilities in control-based intrusion detection methods. They proposed three kinds of attacks that evade detection and seriously disrupt the robotic platform operation.

2.3. Deep Learning-Based Algorithms for Anomaly Detection

Deep learning techniques have recently been increasingly used to protect robots and make them safe. Long Short-Term Memory (LSTM)-based autoencoders are widely used for anomaly detection and security incidents in robotic systems. The following is a selection of solutions proposed by researchers and the identified limitations found in their approaches.

A comparative analysis of the use of autoencoders for robot security anomaly detection is presented in [28]. The robotic systems' logs are extracted from external variables. Next, they are transformed into images that are used to train the autoencoders. The results of the experiments conducted with autonomous boats and service robots confirm the efficiency of the proposed solution.

Azzalini et al. [29] describe their deep learning-based, minimally supervised method for anomaly detection in autonomous robots. A crucial part of the solution is a variational autoencoder architecture (VAE) that models long sensor logs and the novel training method. The training dataset consists mainly of unlabeled observations, and only a tiny amount of nominal behavior is labeled. The results of the experiment presented in the paper confirm the effectiveness of the anomaly detector, both in terms of false positives and false negatives. Moreover, the detector can be used in offline and online modes.

The authors of [7] point out the precise correlation of interaction times between sensors, effectors, and computational elements, which can be successfully used in robot cybersecurity. They present and compare the effects of the application of different techniques for intrusion detection using the example of a mobile robot, i.e., standard classifiers, multilayer perceptron (MLP) deep networks, and LSTM recurrent neural networks. Similarly, it was shown in [30] that by modeling a time series describing the correct behavior of a robot using deep LSTM networks, deviations from the typical performance could be easily detected. However, this approach assumes that the time series is somewhat predictable, which is not guaranteed for a robot operating in a dynamic environment. The authors of [31] demonstrate that this limitation does not occur when the LSTM is implemented in an encoder–decoder architecture. This is confirmed by a study presented in [32], where recurrent convolutional networks in an encoder–decoder architecture were used to detect anomalies in multidimensional time-domain waveforms.

A variational LSTM-based autoencoder (LSTM-VAE) for anomaly detection in robot-assisted feeding behavior is presented in [33]. The anomaly was reported when the reconstruction-based anomaly score exceeded the state-based threshold. The authors claim that the experimental results obtained are better than those for other methods from the literature. The experimental results confirmed that the method effectively detected anomalies from over a dozen raw sensory signals without data pre-processing.

It should be pointed out that in commonly used LSTM-based algorithms, the identification thresholds are calculated from the prediction-error sequences [16]. Hence, fixed-length input signals with the same length are required. Unfortunately, in robotics, this is rarely the case; the recorded executions often have different lengths for a specific movement. Wu, H. et al. [34] extend the concept of LSTM to predict anomalies by admitting multivariate input time series of different lengths. They propose a probabilistic model for tackling the temporal uncertainty in modeling prediction errors of varying lengths. Finally, threshold representation is learned from the trained probabilistic model for abnormal movement detection. A self-designed robot manipulation task comprising six individual movements was used for performance evaluation. The experiments confirmed the high efficiency of the method. The average accuracy of anomaly detection was about 94%. The authors claim that their algorithm outperforms the baseline methods.

Other deep learning models used in robotic systems are residual neural networks. The authors of [35] employed a deep residual neural network architecture for origin–destination trip matrix estimation from traffic sensors. A detection model based on residual learning to address the network degradation problem is proposed in [36]. A proactive anomaly detection scheme for robot navigation in unstructured and uncertain environments using ResNet-18 is described in [37]. The probability of future failure is predicted based on the planned motions and the current observation. Data from multiple sensors are aggregated, correlated, and analyzed to provide robust anomaly detection. The efficiency of this method was tested on field robots. The method captures anomalies in real-time while maintaining a low false detection rate in cluttered fields. Wellhausen et al. [38] employ the deep support vector data description (Deep-SVDD) algorithm and real-valued non-volume preserving (real NVP) transformation for anomaly detection based on RGB-D images of the environment.

The main problem with applying supervised learning algorithms is acquiring abnormal samples. Therefore, in many cases, supervised learning has shown limited use in cybersecurity systems. The use of anomaly detection for mobile robots using a limited abnormal system operation dataset to train the detection algorithm is investigated in [39]. In recent years, unsupervised or semisupervised anomaly detection algorithms have become more widely used in anomaly detection. Generative adversarial networks (GANs) can be successfully used in anomaly detection. Adversarial networks can make abnormal inferences using adversarial learning of the representation of samples. An overview of GAN-based anomaly detection techniques described in the vast literature is provided in [40]. The paper summarizes more than 330 references related to GAN-based anomaly detection. The focus is on the theoretical basis, model implementations, and practical applications of GAN-based detectors. Moreover, the current outstanding issues encountered by GAN-based anomaly detection and future research directions are discussed. The authors provide detailed technical information. The paper can be a guide for researchers who are trying to apply GANs to protect robotic systems against threats.

The literature review shows great interest in using deep learning to detect anomalies in robotic systems. However, it is worth noting that, despite its advantages, deep learning is susceptible to adversarial attacks. This precludes using deep learning in real-life critical applications unless its vulnerabilities are mitigated. A comprehensive survey of adversarial attacks on artificial neural networks is presented in [41]. Ilahi et al. investigate vulnerabilities that can be exploited and provide recommendations for researchers and practitioners to design and develop robust deep learning systems.

The summary of anomaly and cyberattack detection techniques for robotic systems covered by related works is presented in Table 1.

2.4. Legal Framework

Finally, the legal framework is essential in designing social and service robotic systems. Fosch-Villaronga and Mahler [4] discuss cybersecurity challenges and their subordinate security implications based on the specific example of social and service robots. It should be noted that the issue of the interplay between robots, cybersecurity, and security from a European law perspective is not fully explored in the current technical and legal literature. European law does not regulate robots per se, and numerous and overlapping legal requirements focus on specific contexts, such as product safety and medical devices. In addition, the recently enacted European Cybersecurity Act establishes a cybersecurity certification framework that can be used to define cybersecurity requirements for robots. However, specific requirements for the implementation of cyber–physical systems still need to be defined. The authors seek to answer whether the current European legal framework is prepared to counter cyber and physical threats against service and social robots and ensure safe human–robot interactions.

Table 1. Summary of techniques for the cybersecurity of robotic systems covered by related works.

Authors	Technique	Attack Types	Scope	Testbed	Score	Characteristics
Machine learning and other methods						
Bezemskij et al. [5]	Bayesian networks	Sensor spoofing	Cyber–physical threat detection	Robotic vehicle	AUC = 95.3–98.3%	Sensor agnostic and flexibility
Vuong et al. [9]	Decision tree	Command inject., CDoS, malware	Cyberattacks detection	Robotic Vehicle, camera	ACC = 93.8% AUC = 73–97%	Light weight mechanism
Bezemskij et al. [20]	Comparison of signatures of sensor signal characteristics	DoS, sensor spoofing	Anomaly detection	Robotic vehicle	AUC = 93.8–100%	Sensor agnostic
Wang et al. [24]	Measurements comparison with historical data	Sensor spoofing	Attacking defending	iRobot Create 2 iRobot Create 2	Successful rate = 93.3%	Attack detection and mitigation
Vuong et al. [16]	Combination of physical indicators with network IDS	DoS	Cyberattack detection	Arduino-based robot	–	Hybrid attack detection in real-time
Tang et al. [25]	Event-based tracking control	DoS	Tracking control under cyberattack	Amigobot robot	–	Event-triggering strategy
Deep learning						
Ji et al. [37]	ResNet-18	Sensor spoofing	Anomaly Anomaly	TerraSentia robot	F1 = 64.5% AUC = 82.8%	Proactive technique for uncertain workspace
Mantegazza et al. [39]	Real-NVP with outlier exposure	Sensor spoofing	Anomaly detection	Robomaster S1 robot	AUC = 80%	Anomaly detection for cases with limited data of anomalies
Wellhausen et al. [38]	Deep SVDD, Real-NVP	Sensor spoofing	Anomaly detection	Robotic vehicle, RealSense camera	AUC = 95% AUC = 95%	Scalable solution based on multi-modal anomaly detection
Guerrero-Higueras et al. [26]	NB, KNN, SVD, LR, RF, MLP	DoS, sensor spoofing	Cyberattack detection	RB-1 robot commercial RTLS	ACC = 42.9–94.8% (DoS) ACC = 46.8–76.8% (Spoof.)	Attacks on localisation system
Azzalini et al. [29]	VAE with incremental training	Sensor spoofing	Anomaly detection	Simulations on real datasets	FPR = 0–6.5%	Training in a minimally supervised fashion
Park et al. [33]	LSTM-VAE	Sensor spoofing	Anomaly detection	PR2 robot	AUC = 87.1%	Sensitive anomaly det. with low false alarms
Loukas et al. [7]	RNN LSTM	Command injection, DoS, malware	Cyberattack detection	Robotic vehicle	ACC = 82.2–95.4% Acc = 66.9% (0-day)	Cloud-based computing offloading to reduce latency
Wu et al. [34]	Stacked LSTM	Sensor spoofing	Anomaly detection	Baxter robot	ACC = 94%	Dynamic threshold learned from the probabilistic model for abnormal movement

3. Service Mobile Robot Intrusion Detection System

From the literature review, most of today's anomaly and threat detection techniques use learning methods, including deep learning. We aimed to develop and validate a deep learning-based anomaly detection system for a real-world service robot operating in the environment of a typical home with small and cluttered rooms. Moreover, we wanted to verify and compare the effectiveness of deep network models of different complexities in detecting various attacks.

We developed a Service Mobile Robot Intrusion Detection System (SMR_IDS). We assumed the attack vectors were on sensor data used in the robot navigation system. False measurements can result in incorrect and even dangerous behavior from the robot, endangering the environment and the robot itself. The SMR_IDS aims to detect cyberattacks based on identified deviations from normal service robot behavior.

3.1. Architecture and Implementation of SMR_IDS

The SMR_IDS was designed to run on a robot onboard computer as a package executed in the ROS (Robot Operating System) environment [19]. The SMR_IDS consists of three main components (software modules) depicted in Figure 1.

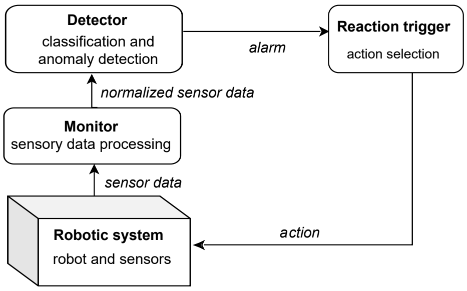


Figure 1. SMR_IDS architecture.

The tasks of the SMR_IDS modules are as follows:

1. Monitor—the module responsible for collecting measurements from selected sensors, normalizing data, and transmitting them to the anomaly detector.
2. Detector—the deep learning model for detecting anomalies in robot behavior based on data provided by the monitor module.
3. Reaction trigger—the module responsible for launching the corresponding reaction procedures in response to a detected cyberattack.

The architecture of the SMR_IDS in the ROS is depicted in Figure 2. All enumerated components were implemented in the ROS. In Figure 2, they are represented by three ROS nodes: `data_collector`, `anomaly_detector`, and `reaction_component`. The SMR_IDS uses the following measurements to detect anomalies: wheel encoder readings, laser scanner measurements, and data from inertial measurement units (linear accelerations and angular velocities). In Figure 2, they are represented by three nodes: joint_states, scan, and base_imu.

3.2. Anomaly Detection Methods

An essential component of the SMR_IDS is the detection module (*detector*). Its task is to check whether the current measurement data indicate the correct behavior of the robot. Any deviation is treated as an anomaly.

Deep learning was applied to anomaly detection based on the authors' experience and research results presented in the literature [31–33,42,43]. In our approach, the neural network's task is to predict the correct behavior of the robot based on measurements collected in given periods. Hence, the learning data set consists of historical data (time series) collected during regular robot operation.

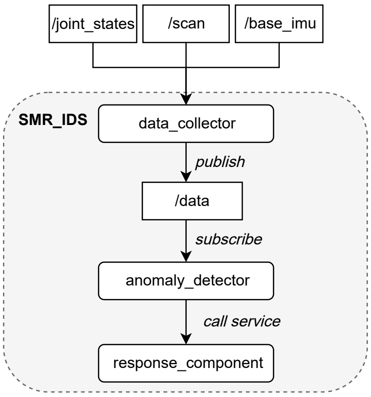


Figure 2. SMR_IDS implementation in ROS.

LSTM networks in an encoder–decoder architecture were used to predict the robot's behavior. The detection module was implemented in two variants with different network types, i.e., Long Short-Term Memory Encoder–Decoder (LSTM-ED) and Convolutional Long Short-Term Memory Encoder–Decoder (ConvLSTM-ED).

An LSTM network can learn long-term dependencies [44–46]. It originated as a modification to the classical Recurrent Neural Network (RNN). RNN networks have units whose outputs are connected to the next layer, and the unit itself is an input. An LSTM network's basic unit, i.e., a memory block, is composed of one or more memory cells and gating functions shared by all cells in the block. Their task is to control the data flow, in particular to accumulate relevant information and forget unuseful information. The gates are as follows:

- i—the input gate controls read access, i.e., selects the information to accumulate in the memory cell;
- f—the forget gate forgets the previous state of the memory cell;
- g—the cell gate squashes the cell input;
- o—the output gate propagates the state of the memory cell to the LSTM unit output (the hidden state).

LSTM networks can analyze behavioral sequences and learn relationships over long periods, which is crucial in our application.

The convolutional LSTM network is an extension of the LSTM network designed to process time-dependent spatial data [23,47]. The main difference between ConvLSTM and LSTM is the number of dimensions of the input data. In LSTM, the input data are one-dimensional. In ConvLSTM, the input data are two-dimensional, and a convolution operator replaces a matrix multiplication operator. In this way, the network captures the basic spatial features of multidimensional data. Convolutional neural networks are computationally efficient and allow the automatic detection of relevant information without human supervision.

Let us focus on the mathematical model of the ConvLSTM's cell (1)–(6). x_t and h_t denote the input and the output (the hidden state) of the ConvLSTM memory cell at time t. c_t is the state of the memory cell. At each time instance t, $t = 1, 2, \ldots, \tau$, the ConvLSTM layer performs the following computations:

$$i_t = \sigma(W_i * x_t + U_i * h_{t-1} + V_i \circ c_{t-1} + b_i), \tag{1}$$

$$f_t = \sigma(W_f * x_t + U_f * h_{t-1} + V_f \circ c_{t-1} + b_f), \tag{2}$$

$$g_t = \tanh(W_g * x_t + U_g * h_{t-1} + b_g), \tag{3}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t, \tag{4}$$

$$o_t = \sigma(W_o * x_t + U_o * h_{t-1} + V_o \circ c_t + b_o), \tag{5}$$

$$h_t = o_t \circ \tanh(c_t), \tag{6}$$

where i_t, f_t, g_t, and o_t are gate outputs at the time t. $W = [W_i, W_f, W_g, W_o]^T$, $U = [U_i, U_f, U_g, U_o]^T$, and $V = [V_i, V_f, V_g, V_o]^T$ are the weight matrices related to the input data x_t, past hidden state h_{t-1}, and previous and current cell states c_{t-1} and c_t, respectively. $b = [b_i, b_f, b_g, b_o]^T$ is the bias vector. σ and tanh denote sigmoid and hyperbolic tangent activation functions. In the above model, all the inputs x_t, cell states c_t, hidden states h_t, and gates i_t, f_t, g_t, o_t calculated at each time t are matrices. Therefore, the convolution operator "$*$" is used for state-to-state and input-to-state transitions. The operator "$\circ$" denotes the Hadamard product.

Figure 3 shows our ConvLSTM-ED network structure, including the data dimensions at each stage.

Let us assume that n denotes the number of robot features (sensor measurements) selected to be used in anomaly detection. Every time t, $t = 1, 2, \ldots, \tau$, we obtain a new data sample $u_t = [u_t^1, u_t^2, \ldots, u_t^n]$ that is added to our data set of measurements. Hence, at the time instant τ, we can create an $n \times \tau$ matrix U_τ containing all measurements.

$$U_\tau = \begin{bmatrix} u_1^1 & u_2^1 & \ldots & u_\tau^1 \\ u_1^2 & u_2^2 & \ldots & u_\tau^2 \\ u_1^3 & u_2^3 & \ldots & u_\tau^3 \\ \ldots & \ldots & \ldots & \ldots \\ u_1^n & u_2^n & \ldots & u_\tau^n \end{bmatrix}. \tag{7}$$

To detect a robot's anomalous behavior, the ConvLSTM-ED input data x_t should contain information about relationships between the readings of different sensors. To encode information about these relationships every time t, we calculate the $n \times n$ autocorrelation matrix $E(U_t U_t^T)$, taking into account the last ω samples (autocorrelation matrix window) from the matrix (7). Finally, we can create τ input matrices x_t:

$$x_t = E(U_t U_t^T) = \begin{bmatrix} \frac{U_t^1 (U_t^1)^T}{\omega} & \frac{U_t^1 (U_t^2)^T}{\omega} & \ldots & \frac{U_t^1 (U_t^n)^T}{\omega} \\ \frac{U_t^2 (U_t^1)^T}{\omega} & \frac{U_t^2 (U_t^2)^T}{\omega} & \ldots & \frac{U_t^2 (U_t^n)^T}{\omega} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{U_t^n (U_t^1)^T}{\omega} & \frac{U_t^n (U_t^2)^T}{\omega} & \ldots & \frac{U_t^n (U_t^n)^T}{\omega} \end{bmatrix}, \tag{8}$$

where $U_t^i = [u_{t-\omega+1}^i, u_{t-\omega+2}^i, \ldots, u_t^i]$, $i = 1, 2, \ldots, n$.

The data processed in this way were fed to the input of the convolutional encoder presented in Figure 3. The SELU (Scaled Exponential Linear Unit) in each of the four encoder layers was used as the activation function of the convolutional layers. The parameters of the layers were as follows:

- Thirty-two filter kernels with a size of 3 × 3;
- Sixty-four filter kernels with a size of 3 × 3;
- One hundred and twenty-eight filter kernels with a size of 2 × 2;
- Two hundred and fifty-six filter kernels with a size of 2 × 2.

Our ConvLSTM network's last layer is an aggregation (max pooling) with a size of 2 × 2. The output of each successive convolutional layer is additionally passed to the ConvLSTM cells. The task of the recurrent layers is to extract temporal relationships. The size and number of filters are the same as in the encoder. The input of the recurrent layer was the 5 × 5 matrix. The ConvLSTM layer returns a matrix, which is then passed to the decoder. The outputs of the previous recurrent layers are attached to the subsequent decoder layers and then passed on. Finally, the network output returns a single matrix of dimension equal to the dimension of the input data (autocorrelation matrix).

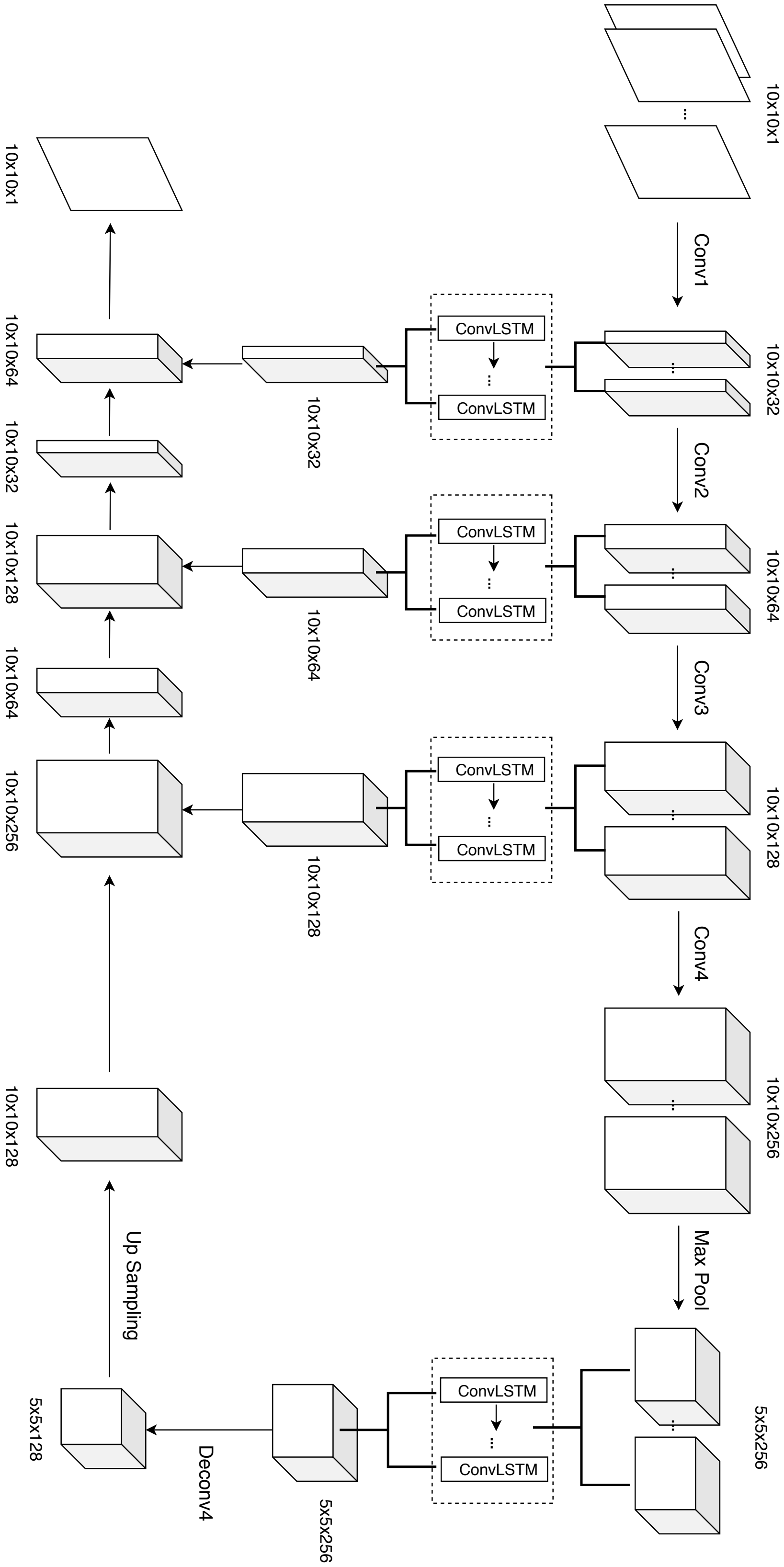


Figure 3. The structure of the ConvLSTM-ED network implemented in SMR_IDS.

3.3. Anomaly Detection Module Implementation

The current version of the SMR_IDS includes two variants of the LSTM-based anomaly detection modules: LSTM-ED and ConvLSTM-ED. They were implemented using the TensorFlow library [48]. The following features were selected to detect attacks on the sensor data used in the autonomous navigation system

1. Angular velocity (rad/s);
2. Linear acceleration in the x axis (m/s^2);
3. Linear acceleration in the y axis (m/s^2);
4. Orientation vector;
5. Right wheel velocity (rad/s);
6. Left wheel velocity (rad/s);
7. Four distance values returned by the laser scanner (minimum values for the right, center, and left parts and the average value) (m).

The anomaly detection scheme is presented in Figure 4.

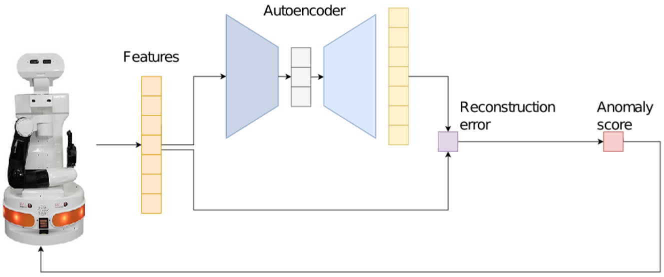


Figure 4. Anomaly detection scheme.

The output from the neural network model (LSTM-ED or ConvLSTM-ED) is compared with the input. The number of incorrectly predicted values, i.e., values for which the difference between the original input and the reconstruction output in the autoencoder is higher than a threshold value η, is computed. In our system, the anomaly score is equal to this value. An attack is reported when the anomaly score exceeds the anomaly score threshold value α.

4. Anomaly Detection Module Training and Validation

4.1. Testbed and Experimental Setup

The testbed used for the development and experimentation of our SMR_IDS was a real service robotic platform operating in the laboratory and replicating a home environment (Figure 5). It was a small room (20 m^2). The laboratory scheme is presented in Figure 6. The SMR_IDS was executed on a PC with the following components: Intel i7-5500U, RAM 8 GB, and GeForce 920 M. The experiments were carried out on the TIAGo mobile service robot from PAL Robotics equipped with the following onboard sensors: a laser scanner, an RGB-D camera, an IMU (inertial measurement unit), and wheel encoders. Furniture and moving people were taken into account. The home automation system communicated between the robot and the sensors, allowing multiple devices to be managed within the local network.

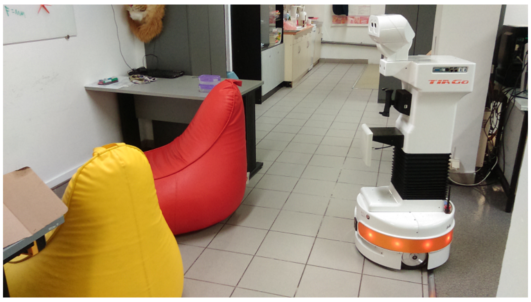
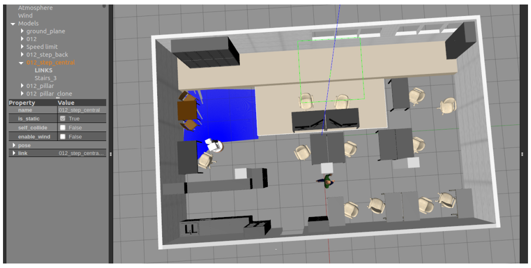


Figure 5. The TIAGo mobile service robot in the laboratory.



Property	Value
name	012_step_central
is_static	☑ True
self_collide	□ False
enable_wind	□ False

Figure 6. Visualization of the laboratory environment in the Gazebo simulation platform.

4.2. Evaluation Metrics

Commonly used metrics were taken to assess the quality of the classifiers in the validation procedure:

- TPR—true positive rate (sensitivity, recall)

$$TPR = \frac{TP}{TP + FN}, \tag{9}$$

- PPV—positive predictive value (precision)

$$PPV = \frac{TP}{TP + FP}, \tag{10}$$

- $F1$ score

$$PPV = 2\frac{PPV \cdot TPR}{PPV + TPR}, \tag{11}$$

where TP and FP denote the number of correctly and incorrectly identified anomalies, and TN and FN denote true and false predictions of the correct robot behavior.

4.3. Data Acquisition

A key point in the training and validation of LSTM networks is the preparation of a high-quality dataset. We conducted a series of experiments to collect data for the LSTM-ED and ConvLSTM-ED detectors' training and validation.

The robot was allowed to move around the laboratory. Its task was to reach designated target points. The robot determined a feasible motion trajectory based on the onboard sensor readings. After reaching each target point, we assumed a standstill time of 2 to 10 s. The robot's orientation at each target point was randomized from the range $(-2\pi, 2\pi)$. Different target points were given to the navigation system.

The measurements of the features listed in Section 3.3 were taken every 0.2 s. Due to the need for real-time anomaly detection, the detector analyzed the data within the specified time window. The feature vectors published by the *monitor* were collected in sequences containing samples gathered over nineteen hours. We applied a median filter to avoid large single values, falsely suggesting anomalies resulting from a change in the robot's behavior (e.g., moving after stopping, stopping, changing orientation). Next, all measurement data were normalized using the mean value and standard deviation calculated for the training dataset. Finally, a total of 106,099 samples, i.e., vectors of sensor measurements, were collected. They were divided into a learning set (84,879 samples) and a validation set (21,220 samples). The results of the initial experiments were used to determine the attributes and parameters of the LSTM and ConvLSTM networks. They are presented in Table 2.

Table 2. Parameters of the LSTM-ED and ConvLSTM-ED detectors.

Parameter	LSTM-ED	ConvLSTM-ED
Cost function	MSE	MSE
Optimizer	Adam	Adam
Metric	loss	loss
Activation function	tanh	SELU
Number of epochs	12	12
Sequence length k	10	5
Dropout	0.3	0.3
Number of hidden neurons	300	300
Autocorrelation matrix window ω	–	10
Reconstruction error threshold η	0.4	0.5
Anomaly score threshold α	4	6

4.4. LSTM-ED and ConvLSTM-ED Detectors' Training and Validation

Finally, we trained and validated both detectors on our training and validation datasets. Figures 7 and 8 depict training and validation loss for LSTM-ED and ConvLSTM-ED detectors. The training loss metric assesses how a model fits the training data. The validation loss shows the performance of a trained model on the validation data. Both plots illustrate that the optimal fit is in epoch 12. They indicate that further training results in overfitting, i.e., both models perform well on the training data but cannot generalize on new data. Moreover, the ConvLSTM-ED model fits the training data faster and performs better.

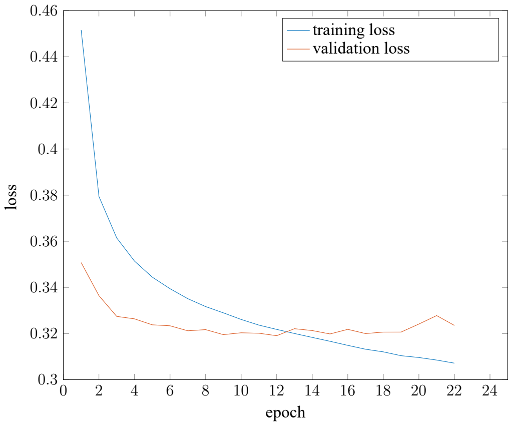


Figure 7. Training and validation loss (LSTM-ED).

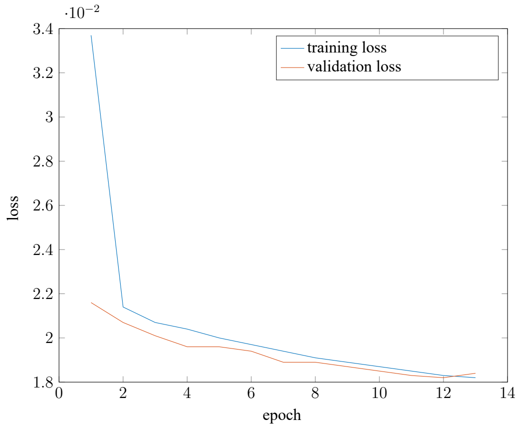


Figure 8. Training and validation loss (ConvLSTM-ED).

5. Performance Evaluation of the SMR_IDS System

5.1. Experimental Setup

The performance of the SMR_IDS was tested in the testbed described in Section 4.1. Experiments were conducted for both detectors, i.e., LSTM-ED and ConvLSTM-ED. Two test scenarios involving attacks on the robot navigation system were considered:

- Test 1—false command injection;
- Test 2—laser scanner spoofing.

Test 1 involved gaining access to one of the networked computers or the robot and initiating manual control of the robot. The tests were designed to demonstrate whether it is possible to distinguish between a robot controlled by a navigation system and one controlled manually.

In Test 2, the robot received false laser scanner readings. False laser scanner measurements can involve:

- The identification of obstacles that do not exist in the workspace;
- Incorrect robot localization;
- Incorrect robot navigation.

Figures 9 and 10 show possible effects of attacks involving laser scanner spoofing.

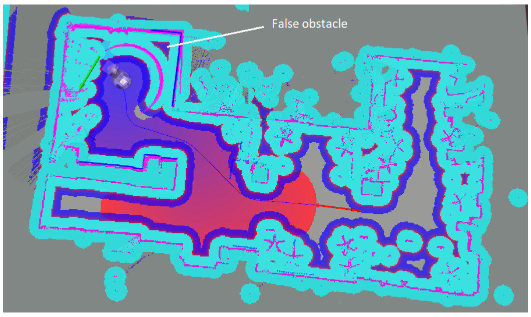


Figure 9. The robot’s navigation system identifies a non-existent obstacle at a distance of 1 m from the robot (half-circle).

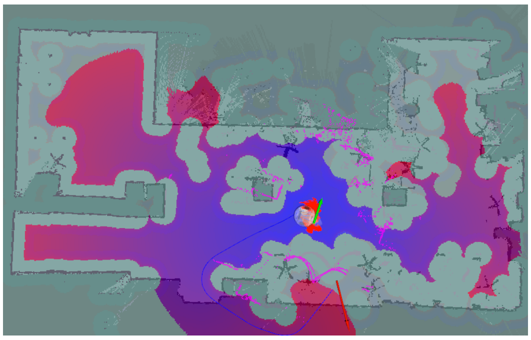


Figure 10. False measurements affect navigation errors; the robot's navigation system can determine unacceptable paths.

Test 2 was implemented in two variants:

- Variant 1: the measured distance was replaced by a distance of 9 m;
- Variant 2: the measured distance was replaced by a distance of 1 m.

The measurement range of the scanner ranged from 0.5 m to 10 m.

5.2. Results of the Experiments

Figures 11–14 show selected experimental results. The periods during which the attack was conducted are shaded in gray. All figures present the anomaly score detected by LSTM-ED and ConvLSTM-ED at each second, based on five samples of selected features described in Section 3.3. In Test 1, five features (speed and acceleration) were used. In Test 2, four features (laser scanner measurements) were taken into account.

Twelve experiments to take control of the robot (Test 1) were performed. The SMR_IDS worked properly, as shown in Figures 11 and 12. Only minor differences in the detectors' performance were observed. LSTM-ED detected all attacks but occasionally reported false alarms. ConvLSTM-ED missed only a few attacks without reporting any false alarms.

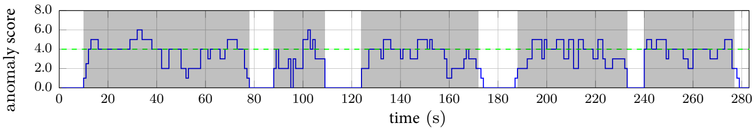


Figure 11. The anomaly score for the LSTM-ED detector; Test 1 scenario.

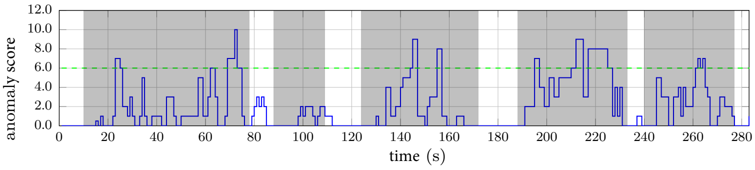


Figure 12. The anomaly score for the ConvLSTM-ED detector; Test 1 scenario.

Nine experiments were performed for each variant of Test 2. Figures 13 and 14 show some experimental results for variant 1.

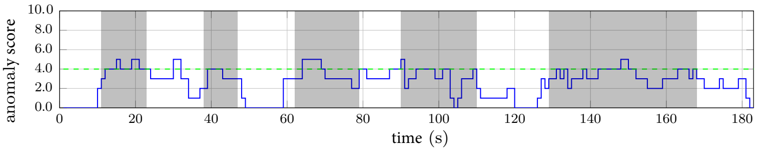


Figure 13. The anomaly score for the LSTM-ED detector; Test 2 scenario (variant 1).

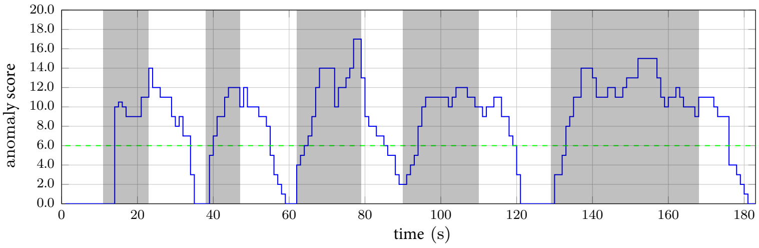


Figure 14. The anomaly score for the ConvLSTM-ED detector; Test 2 scenario (variant 1).

The superiority of the convolutional network in intrusion detection performance can be seen in the case of laser scanner spoofing attacks (Test 2). ConvLSTM-ED uses the correlation between all features. The false large distance of 9 m (variant 1) significantly affected the network outputs, causing a large reconstruction error. In the case of a cluttered small room (20 m^2), it is a rather unusual value. ConvLSTM-ED detected all attacks, regardless of their duration. The effectiveness of LSTM-ED was much lower.

In the case of the laser scanner spoofing attack in variant 2, both detectors did not identify the attack. The distance to the obstacle equal to 1 m is typical in the case of a cluttered small room, and such false measurements do not significantly affect the robot's behavior. The only anomalies noticed were the changes in the direction of movement that were difficult to predict, which also occurred in the learning process. Therefore, the detectors did not raise the alarm.

The experimental results are summarized in Table 3. The Table contains the mean values of three standard metrics, i.e., precision, sensitivity, and F1-score, for all experiments carried out for both attack scenarios, i.e., Test 1 and Test 2 (variant 1).

Table 3. SMR_IDS performance efficiency in anomaly detection.

	Test 1			**Test 2 (Variant 1)**		
Method	**PPV**	**TPR**	**F1**	**PPV**	**TPR**	**F1**
LSTM-ED	92%	100%	96%	83%	56%	67%
ConvLSTM-ED	100%	83%	91%	100%	100%	100%

In conclusion, the experimental results confirmed the high efficiency of the SMR_IDS in anomaly detection. In the case of taking control of a robot, the results of applying the LSTM-ED and ConvLSTM-ED detectors were similar. However, the ConvLSTM-ED network is recommended for detecting sensor spoofing attacks.

To summarize our research, we test our solution against other LSTM-based anomaly detection architectures used to protect robotic systems in Table 4.

Table 4. Various applications of LSTM-based anomaly detectors.

Authors	Deep Learning Method	Training	Anomaly/Attack Types	Application	Testbed Evaluation
Park et al. [33]	LSTM variational autoencoder	Supervised Training dataset: normal and abnormal behaviors	Anomalies: abnormal behaviors (12 anomalies)	Multimodal anomaly detection in robot-assisted feeding. Onboard detection in real-time.	PR2 robot with two arms sensors: force/torque, current, joint encoders, microphone, RGB-D camera
Loukas et al. [7]	LSTM	Supervised Training dataset: normal data, synthetic attacks	Attacks: false command injection, DoS, malware	Cyber and physical attacks on robotic vehicle detection. Onboard and offloaded detection.	Four-wheel vehicle sensors: electronic compass, energy meter, encoders, accelerometer, camera
Wu et al. [34]	Stacked LSTM (multiple LSTM layers)	Supervised Training dataset: normal and abnormal movements	Anomalies: abnormal movements (6 movements)	Social robot abnormal movement detection. Onboard detection in real-time.	Social robot sensors: force-torque, joint encoders, tactile
SMR_IDS	LSTM and convolution LSTM autoencoders	Unsupervised Training dataset: normal sensor readings, normal commands	Anomalies: abnormal sensor readings, abnormal commands (10 anomalies)	Service robot navigation system anomaly detection. Onboard detection in real-time.	Service mobile robot sensors: laser scanner, encoders, IMU

6. Conclusions

Cybersecurity breaches could have devastating consequences and undermine the trust between humans and robots. Moreover, it is crucial to understand how robots will affect tomorrow's cybersecurity strategy. Unfortunately, robotic system security is still too little researched and developed. Cyberattacks may become even larger and more frequent if no action is taken to secure such systems. This is one reason why the security of service and social mobile robots is such important for people and industry. At the same time, numerous manufacturers neglect security aspects. They often ignore vulnerabilities found in devices and software.

This paper overviews the strategies and methods for protecting robotic systems composed of robots and sensors against cyberattacks. We outline the main properties and criteria to consider when developing intrusion-detecting systems. Moreover, we present and investigate the intrusion detection SMR_IDS that uses deep learning for anomaly detection. In order to verify the effectiveness of the SMR_IDS in detecting anomalies, we prepared two attack scenarios: false command injection and sensor spoofing. The experiments were conducted on the TIAGo robot equipped with onboard sensors and executing autonomous navigation tasks, i.e., robot localization on a map, planning a path based on the global cost map, and following the desired path, avoiding static and dynamic obstacles. The effectiveness of the anomaly detection of the two types of detectors based on LSTM and convolutional LSTM autoencoders was presented and discussed. The obtained F1 scores were 67–96% (LSTM-ED) and 91–100% (ConvLSTM-ED), depending on the type of attack. The experimental evaluation confirms that deep recurrent neural networks with encoder–decoder architectures can effectively detect anomalies in mobile robotic systems. Moreover, results show that detecting more sophisticated attacks requires more complex deep learning models, such as convolutional LSTM.

The strength of our anomaly detection system is that it can be implemented on robotic platforms equipped with typical sensors and can operate in the environment of a typical home with small and cluttered rooms. In addition, it can work in real-time and operate in a continuous learning mode. However, it should be noted that both LSTM-ED and ConvLSTM-ED models detect abnormal sensor readings and commands but cannot definitively determine whether an attack has occurred.

There is a need to develop appropriate response mechanisms. Therefore, future work on the SMR_IDS will include grouping anomalies to determine their nature and considering the attack's duration and the feature's influence on the calculated severity of the anomaly. This will allow the development of an effective response mechanism for detected events.

The second challenge we plan to address in future studies is expanding the system's functionality to include detecting a substitution attack on readings with typical values. To this end, it is possible to increase the observation window to determine whether this will allow the network to detect anomalies in feature values and how the task is performed.

Author Contributions: Conceptualization, W.S. and K.L.; methodology, K.L. and W.S.; literature review E.N.-S.; formal analysis and investigation, K.L., W.S. and E.N.-S.; software, K.L.; validation, K.L., W.S. and E.N.-S.; data curation, K.L.; writing—original draft preparation, E.N.-S., W.S. and K.L.; writing—review and editing, E.N.-S.; supervision, W.S.; project administration, W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, F.; Lu, Y.; Vasilakos, A.; Hao, Q.; Ma, R.; Patil, Y.; Zhang, T.; Lu, J.; Li, X.; Xiong, N. Robust Cyber-Physical Systems: Concept, models, and implementation. *Future Gener. Comput. Syst.* **2015**, *56*, 449–475. [CrossRef]
2. Mitchell, R.; Chen, I. A Survey of Intrusion Detection Techniques for Cyber-Physical Systems. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 55. [CrossRef]

3. Dudek, W.; Szynkiewicz, W.; Winiarski, T. Nao Robot Navigation System Structure Development in an Agent-Based Architecture of the RAPP Platform. In Proceedings of the Challenges in Automation, Robotics and Measurement Techniques, Warsaw, Poland, 2–4 March 2016; Szewczyk, R., Zielinski, C., Kaliczynska, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 440, pp. 623–633. ._54. [CrossRef]
4. Fosch-Villaronga, E.; Mahler, T. Cybersecurity, safety and robots: Strengthening the link between cybersecurity and safety in the context of care robots. *Comput. Law Secur. Rev.* **2021**, *41*, 105528. [CrossRef]
5. Bezemskij, A.; Loukas, G.; Gan, D.; Anthony, R. Detecting cyber-physical threats in an autonomous robotic vehicle using Bayesian networks. In Proceedings of the IEEE International Conference on Internet of Things (iTh-ings) and IEEE Green Computing and Communications (Green-Com) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; pp. 98–108.
6. Lacava, G.; Marotta, A.; Martinelli, F.; Saracino, A.; La Marra, A.; Gil-Uriarte, E.; Vilches, V. Cybsersecurity Issues in Robotics. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. (JoWUA)* **2021**, *12*, 1–28.
7. Loukas, G.; Vuong, T.; Heartfield, R.; Sakellari, G.; Yoon, Y.; Gan, D. Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning. *IEEE Access* **2018**, *6*, 3491–3508. [CrossRef]
8. Mitchell, R.; Chen, I.R. Behavior Rule Specification-Based Intrusion Detection for Safety Critical Medical Cyber Physical Systems. *IEEE Trans. Dependable Secur. Comput.* **2015**, *12*, 16–30. [CrossRef]
9. Vuong, T.; Loukas, G.; Gan, D. Performance Evaluation of Cyber-Physical Intrusion Detection on a Robotic Vehicle. In Proceedings of the IEEE International Conference on Computer and Information Technology, Dhaka, Bangladesh, 21–23 December 2015; pp. 2106–2113. [CrossRef]
10. Vuong, T.; Loukas, G.; Gan, D.; Bezemskij, A. Decision tree-based detection of denial of service and command injection attacks on robotic vehicles. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Rome, Italy, 16–19 November 2015; pp. 1–6.
11. Yaacoub, J.P.; Noura, H.N.; Salman, O.; Chehab, A. Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations. *Int. J. Inf. Secur.* **2022**, *21*, 115–158. [CrossRef] [PubMed]
12. Karwowski, J.; Szynkiewicz, W. Quantitative metrics for benchmarking human-aware robot navigation. *IEEE Access* **2023**, *11*, 79941–79953. [CrossRef]
13. Dudek, W.; Szynkiewicz, W. Cyber-security for mobile service robots–challenges for cyber-physical system safety. *J. Telecommun. Inf. Technol.* **2019**, *2*, 29–36. [CrossRef]
14. Sabaliauskaite, G.; Ng, G.S.; Ruths, J.; Mathur, A. A Comprehensive Approach, and a Case Study, for Conducting Attack Detection Experiments in CyberPhysical Systems. *Robot. Auton. Syst.* **2017**, *98*, 174–191. [CrossRef]
15. Loukas, G.; Karapistoli, E.; Panaousis, E.; Sarigiannidis, P.; Bezemskij, A.; Vuong, T. A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles. *Ad Hoc Netw.* **2019**, *84*, 124–147. [CrossRef]
16. Vuong, T.; Filippoupolitis, A.; Loukas, G.; Gan, D. Physical indicators of cyber attacks against a rescue robot. In Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS), Budapest, Hungary, 24–28 March 2014; pp. 338–343. [CrossRef]
17. Morante, S.; Victores, J.G.; Balaguer, C. Cryptobotics: Why robots need cyber safety. *Front. Robot. AI* **2015**, *2*, 23. [CrossRef]
18. Finnicum, M.; King, S. Building secure robot applications. In Proceedings of the 6th USENIX Workshop on Hot Topics in Security, San Francisco, CA, USA, 9 August 2011; USENIX Association: Berkeley, CA, USA, 2011; HotSec'11, p. 1.
19. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A. ROS: An Open-Source Robot Operating System. In Proceedings of the IEEE ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, pp. 1–6.
20. Bezemskij, A.; Loukas, G.; Anthony, R.; Gan, D. Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle. In Proceedings of the 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS), Granada, Spain, 14–16 December 2016; pp. 61–68.
21. Guo, P.; Kim, H.; Virani, N.; Xu, J.; Zhu, M.; Liu, P. Exploiting physical dynamics to detect actuator and sensor attacks in mobile robots. *arXiv* **2017**, arXiv:1708.01834. https://doi.org/10.48550/arXiv.1708.01834.
22. Ahmad Yousef, K.M.; AlMajali, A.; Ghalyon, S.A.; Dweik, W.; Mohd, B.J. Analyzing Cyber-Physical Threats on Robotic Platforms. *Sensors* **2018**, *18*, 1643. [CrossRef]
23. Zapata-Impata, B.; Gil, P.; Torres, F. Learning Spatio Temporal Tactile Features with a ConvLSTM for the Direction Of Slip Detection. *Sensors* **2019**, *19*, 523. [CrossRef] [PubMed]
24. Wang, C.; Tok, Y.C.; Poolat, R.; Chattopadhyay, S.; Elara, M.R. How to secure autonomous mobile robots? An approach with fuzzing, detection and mitigation. *J. Syst. Archit.* **2021**, *112*, 101838. [CrossRef]
25. Tang, Y.; Zhang, D.; Ho, D.W.; Yang, W.; Wang, B. Event-Based Tracking Control of Mobile Robot With Denial-of-Service Attacks. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 3300–3310. [CrossRef]
26. Guerrero-Higueras, A.; DeCastro-García, N.; Matellan, V. Detection of Cyber-attacks to indoor real time localization systems for autonomous robots. *Robot. Auton. Syst.* **2018**, *99*, 75–83. [CrossRef]
27. Dash, P.; Karimibiuki, M.; Pattabiraman, K. Stealthy attacks against robotic vehicles protected by control-based intrusion detection techniques. *Digit. Threat. Res. Pract.* **2021**, *2*, 7. [CrossRef]

28. Olivato, M.; Cotugno, O.; Brigato, L.; Bloisi, D.; Farinelli, A.; Iocchi, L. A Comparative Analysis on the use of Autoencoders for Robot Security Anomaly Detection. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 984–989. [CrossRef]
29. Azzalini, D.; Bonali, L.; Amigoni, F. A minimally supervised approach based on variational autoencoders for anomaly detection in autonomous robots. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2985–2992. [CrossRef]
30. Malhotra, P.; Vig, L.; Shroff, G.; Agarwal, P. Long short term memory networks for anomaly detection in time series. In Proceedings of the European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 22–23 April 2015; Volume 89, pp. 89–94.
31. Malhotra, P.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; Shroff, G. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv* **2016**, arXiv:1607.00148. https://doi.org/10.48550/arXiv.1607.00148.
32. Zhang, C.; Song, D.; Chen, Y.; Feng, X.; Lumezanu, C.; Cheng, W.; Ni, J.; Zong, B.; Chen, H.; Chawla, N. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. *CoRR* **2018**, abs/1811.08055. Available online: https://arxiv.org/abs/1811.08055 (accessed on 4 October 2023). [CrossRef]
33. Park, D.; Hoshi, Y.; Ch.C., K. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1544–1551. [CrossRef]
34. Wu, H.; Yan, W.; Xu, Z.; et al.. Multimodal Prediction-Based Robot Abnormal Movement Identification Under Variable Time-length Experiences. *J. Intell. Robot. Syst.* **2022**, *104*, 8. [CrossRef]
35. Alshehri, A.; Owais, M.; Gyani, J.; Aljarbou, M.H.; Alsulamy, S. Residual Neural Networks for Origin–Destination Trip Matrix Estimation from Traffic Sensor Information. *Sustainability* **2023**, *15*, 9881. [CrossRef]
36. Man, J.; Sun, G. A Residual Learning-Based Network Intrusion Detection System. *Secur. Commun. Netw.* **2021**, *2021*, 9. [CrossRef]
37. Ji, T.; Sivakumar, A.; Chowdhary, G.; Driggs-Campbell, K. Proactive Anomaly Detection for Robot Navigation With Multi-Sensor Fusion. *arXiv* **2022**, arXiv:2204.01146. https://doi.org/10.48550/arXiv.2204.01146.
38. Wellhausen, L.; Ranftl, R.; Hutter, M. Safe Robot Navigation Via Multi-Modal Anomaly Detection. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1326–1333. [CrossRef]
39. Mantegazza, D.; Giusti, A.; Gambardella, L.; Guzzi, J. An Outlier Exposure Approach to Improve Visual Anomaly Detection Performance for Mobile Robots. *IEEE Robot. Autom. Lett.* **2022**, *7*, 11354–11361. [CrossRef]
40. Xia, X.; Pan, X.; Li, N.; He, X.; Ma, L.; Zhang, X.; Ding, N. GAN-Based Anomaly Detection: A Review. *Neurocomputing* **2022**, *493*, 497–535. [CrossRef]
41. Inaam, I.; Usama, M.; Qadir, J.; Janjua, M.; Al-Fuqaha, A.; Hoang, D.; Niyato, D. Challenges and Countermeasures for Adversarial Attacks on Deep Reinforcement Learning. *IEEE Trans. Artif. Intell.* **2022**, *3*, 90–109. [CrossRef]
42. Khan, M.A.; Karim, M.; Kim, Y. A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network. *Symmetry* **2019**, *11*, 583. [CrossRef]
43. Lis, K.; Niewiadomska-Szynkiewicz, E.; Dziewulska, K. Siamese Neural Network for Keystroke Dynamics-Based Authentication on Partial Passwords. *Sensors* **2023**, *23*, 6685. [CrossRef] [PubMed]
44. Olah, C. Understanding LSTM Networks. Available online: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 1 July 2019).
45. Gers, F.A.; Schraudolph, N.N.; Schmidhuber, J. Learning Precise Timing with Lstm Recurrent Networks. *J. Mach. Learn. Res.* **2003**, *3*, 115–143. [CrossRef]
46. Graves, A. Generating Sequences With Recurrent Neural Networks. *arXiv* **2014**, arXiv:cs.NE/1308.0850.
47. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *CoRR* **2015**, abs/1506.04214. Available online: https://papers.nips.cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf (accessed on 4 October 2023).
48. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 20 September 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.