applied sciences

MDPI

Review

Trustworthy Artificial Intelligence Methods for Users' Physical and Environmental Security: A Comprehensive Review

Sabina Szymoniak *, Filip Depta, Łukasz Karbowiak and Mariusz Kubanek

Department of Computer Science, Częstochowa University of Technology, 42-201 Czestochowa, Poland; filip.depta@pcz.pl (F.D.); lukasz.karbowiak@icis.pcz.pl (Ł.K.); mariusz.kubanek@icis.pcz.pl (M.K.)
* Correspondence: sabina.szymoniak@icis.pcz.pl

Abstract: Artificial Intelligence is an indispensable element of the modern world, constantly evolving and contributing to the emergence of new technologies. We meet it in everyday applications, primarily using intelligent systems that aim to improve our lives. Artificial Intelligence techniques must inspire users' trust because they significantly impact virtually every industry and person. For this reason, systems using Artificial Intelligence are subject to many requirements to verify their trustworthiness in various aspects. This review focused on users' physical and environmental security, considering the safety and robustness dimensions of Trustworthy Artificial Intelligence. We examined these Trustworthy Artificial Intelligence solutions and dimensions because security is one of the most-critical aspects of human life and can be considered in many different contexts. We examined the trustworthiness of Artificial Intelligence techniques in systems supporting road safety and securing computer network users. Also, we analyzed the challenges and requirements of the newly designed solutions using Trustworthy Artificial Intelligence methods. Verifying Trustworthy Artificial Intelligence solutions and their practical use will increase users' physical and environmental security.

Keywords: Trustworthy Artificial Intelligence; safety and robustness; physical and environmental security; traffic and pedestrian safety; Intrusion-Detection Systems

check for updates

Citation: Szymoniak, S.; Depta, F.; Karbowiak, Ł.; Kubanek, M. Trustworthy Artificial Intelligence Methods for Users' Physical and Environmental Security: A Comprehensive Review. *Appl. Sci.* **2023**, *13*, 12068. https://doi.org/10.3390/app132112068

Academic Editor: Krzysztof Koszela

Received: 29 September 2023
Revised: 30 October 2023
Accepted: 3 November 2023
Published: 6 November 2023

1. Introduction

We encounter various facilities in everyday life provided by different networks' systems and architectures. Most of such solutions belong to Cyber–Physical Systems (CPSs) or the Internet of Things (IoT). These systems use sensors and various Artificial Intelligence (AI) algorithms to improve our lives [1]. Moreover, they work in different areas of our lives [2–4]. For example, we can find such systems in medicine [5], sports [6], or security [7]. Each solution is characteristic of a specific area and problem. We find systems that use sensors to monitor patients' health in medicine. The implemented AI methods, combined with sensors, can help control the vital functions of chronically ill people and signal the need to deliver medications to the patient when necessary. People with eyesight problems can also be helped by using AI [8,9]. In sports, we can find AI methods that enable the analysis of the athlete's movement and performance. Additionally, these systems can analyze the athlete's vital functions and react in life-threatening situations [10].

On the other hand, the security aspect affects many different planes of human life. We can consider physical and environmental security here. Physical and environmental security cover many issues. Physical security primarily refers to protecting people and property against various factors (for example, fire, flood, theft, vandalism, and terrorism). Environmental security, in turn, refers to the protection of specific infrastructure. Both issues are reflected in various standards prepared by agencies such as NIST, ISO, COBIT, and GDR [11]. In particular, by physical security, we mean protecting human life against factors that may contribute to human death. For example, physical security may include preventing car accidents. Based on other drivers' behavior, we can predict a possible collision. Also, physical security is related to pedestrians at crossings and on the shoulders of the road.

In contrast, environmental security concerns data security and computer network users' identity. Especially characteristic are IoT systems, which consist of many communicating wireless devices (mobile devices and sensors). IoT systems can also use Wireless Sensors Network (WSN) technology, the task of which is to monitor and collect data from a defined area. The characteristics of the data sent by users of devices located in IoT systems can be varied. Regardless of the data characteristics, the data collected by these devices should be appropriately secured against unauthorized access and attacks by rogue users [12,13].

As mentioned, CPSs or IoT systems use Artificial Intelligence algorithms. In short, these algorithms tell the system how to learn to operate on its own. AI algorithms use techniques like Deep or Machine Learning, Cloud Computing, or Spiking Neural Networks (SNNs), depending on the problem to be solved. As AI algorithms' users, we can set some requirements for them. From the physical and environmental security point of view, these requirements will concentrate on dimensions like safety, robustness or privacy, and data governance. Therefore, we can set the following requirements: the ability to make informed decisions by system users, the security of users, their privacy and data, system resistance to attacks, traceability, the transparency of system components, and responsibility [14–16].

Artificial Intelligence algorithms must meet the requirements of being trustworthy. The concept of TAI emerged in response to the rapid pace of technological change. Moreover, the Trustworthy Artificial Intelligence (TAI) systems have become a priority for the European Union. AI systems must be human-centric and serve humanity and the good of society at large. These systems offer enormous opportunities, but also pose certain risks, which can impact society. As a result, trust in technology has become a key goal for developers of AI-based systems [14]. A Trustworthy Artificial Intelligence system must have three characteristics throughout its life cycle. The first is legal compliance, which means that the system must comply with applicable legal provisions at the international or national level. The second feature is ethics, which requires the system to comply with ethical principles and values that TAI must ensure. The most-important ethical principles include respect for human autonomy, justice, damage prevention, and the possibility of explanation. Also, it is necessary to consider specific values for specific groups of people (e.g., children and people with disabilities). The third feature is robustness, which relates to both a technical and a social point of view. All these features should work together to be a Trustworthy AI [14–16]. The TAI-equipped systems must be assessed against the mentioned earlier requirements and dimensions that describe their attributes or characteristics.

Moreover, the safety and robustness dimensions of TAI are strictly connected to TAI's ethical and explainable aspects. They are essential for building trust in AI systems. TAI requires ethical data processing, so internal procedures and policies to ensure compliance with data protection laws can also help facilitate ethical data processing and, thus, complement existing legal processes. In turn, explainability is crucial to building and maintaining user trust in AI systems. This principle means that processes must be transparent, the capabilities and goals of AI systems openly communicated, and decisions as explainable as possible to those directly and indirectly influenced by them. Without this information, the decision cannot be adequately challenged. It is not always possible to explain why a particular model produced a particular result or decision (and what combination of inputs contributed to it).

Artificial Intelligence is an indispensable element of the modern world, and its proper use will allow for good cooperation with humans in the future. AI is still evolving as it is the primary driver of emerging technologies such as big data, autonomy, and robotics, and it will continue to act as a technological innovator. Learning about AI's characteristics and effective use will change work and workplace dynamics forever. AI will impact virtually every industry and every person. For this reason, we should focus on each challenge and problem that can be solved using Artificial Intelligence methods.

1.1. Motivations and Contributions

In the era of the development of computer technologies and increased user mobility, the demand for intelligent systems (CPSs) that introduce amenities in our lives using various methods and techniques of AI is also growing. Considering that these systems use devices and sensors that process user data, it should be stated that CPSs must inspire users' trust and adapt to their needs. A special need and, simultaneously, a requirement for CPSs is security, which can be considered on many levels. People want to be safe on the way to work or the shop and do not want to become victims of car accidents. On the other hand, people expect technological security as well. As computer network users, people expect their data to remain intact or lost as a result of data transmission security errors.

New solutions in the physical and environmental security field should contribute to increasing their users' safety levels. However, using AI methods may raise concerns about the effectiveness of these solutions and possible malfunctions. For this reason, regularly reviewing these solutions, their efficacy, and problematic issues is necessary, especially in TAI algorithms' safety and robustness dimensions. Thanks to this, it will be possible to improve solutions that require it.

This review focused on TAI-based users' physical and environmental security solutions. We used these security types to describe the realizations and characteristics of the safety and robustness of TAI dimensions. We believe that studying the safety and robustness dimensions of TAI in users' physical and environmental security solutions can help readers understand the state-of-the-art theory and practice in this regard.

First, we present the levels of vehicle automation. Next, we consider the work on detecting cars in traffic by different neural network approaches and how this affects autonomous vehicles. Then, we discuss pedestrian safety and address the issue of pedestrian detection using different methods. Next, we consider some works related to adversarial attacks on SNNs, DVS cameras, and bioinspired solutions to enhance adversarial attack resilience and TAI regarding Spiking Neural Networks' opportunities and risks. Also, we explain the safety problems and necessities in the IoT and sensor networks. We discuss the safety levels implemented by intrusion- and attack-detection systems. We analyze the challenges and requirements of the newly designed solutions connected with TAI methods.

Also, we looked for other reviews related to Trustworthy Artificial Intelligence methods for users' physical and environmental security. We found reviews focused only on physical (for example, [17–21]) or environmental security (for example, [22–26]). These works mainly considered some subset of Artificial Intelligence methods and did not consider the trustworthy dimensions of AI.

1.2. Methodology

We mainly collected the papers using various search engines, including Google Scholar and DBLP. Moreover, we analyzed the references from the found articles and citations to these articles. Our main goal was the most-complete and up-to-date overview of Artificial Intelligence methods for users' physical and environmental security. Considering these two security areas in CPSs and IoT systems, we focused on the systems responsible for the security of network users, traffic, and pedestrian crosses. Also, we considered cases for the safety and robustness of more-strictly neuro-inspired AI, particularly the trustfulness of rarely exploited Spiking Neural Networks.

1.3. Organization

The organization of the rest of this paper is as follows. Section 2 briefly presents trustworthy methods in Artificial Intelligence. We describe AI techniques and their applications. We focused on techniques like Deep Learning, Fuzzy Systems, Quantum Computing, Cloud Computing, and Edge Computing. Also, we explain how neural networks work. We briefly explain SNNs, how they differ from traditional ANNs, what Dynamic Vision Sensor (DVS) cameras are and what the differences are compared to traditional cameras. In Section 3, we present physical safety in traffic. This is realized by monitoring the vehicle environment.

We present the categories of automobile automation. Next, we show the methods of vehicle detection. Mainly, these are methods based on neural networks. In the next step, we discuss pedestrian safety. Also, we present accident statistics and discuss different approaches for pedestrian detection. Then, we review some works about adversarial attacks on SNNs, DVS cameras, and bioinspired solutions to enhance adversarial attack resilience. We also consider TAI regarding SNNs' opportunities and risks and introduce Dynamic Vision Sensors as an emerging, promising technology to enhance road safety. In Section 4, we discuss issues connected to the safety and security of computer network users. We describe security requirements and threats. Afterwards, we provide an overview of the TAI-based method used for intrusion and attack detection in CPSs (including specific solutions like the IoT). Next, we summarize and conclude our analysis. In the last section, we present an overview of the entire article, conclusions from the research, and challenges and future directions in users' physical and environmental security.

Figure 1 shows a visualization of the processed topics related to the rest of the article's structure.

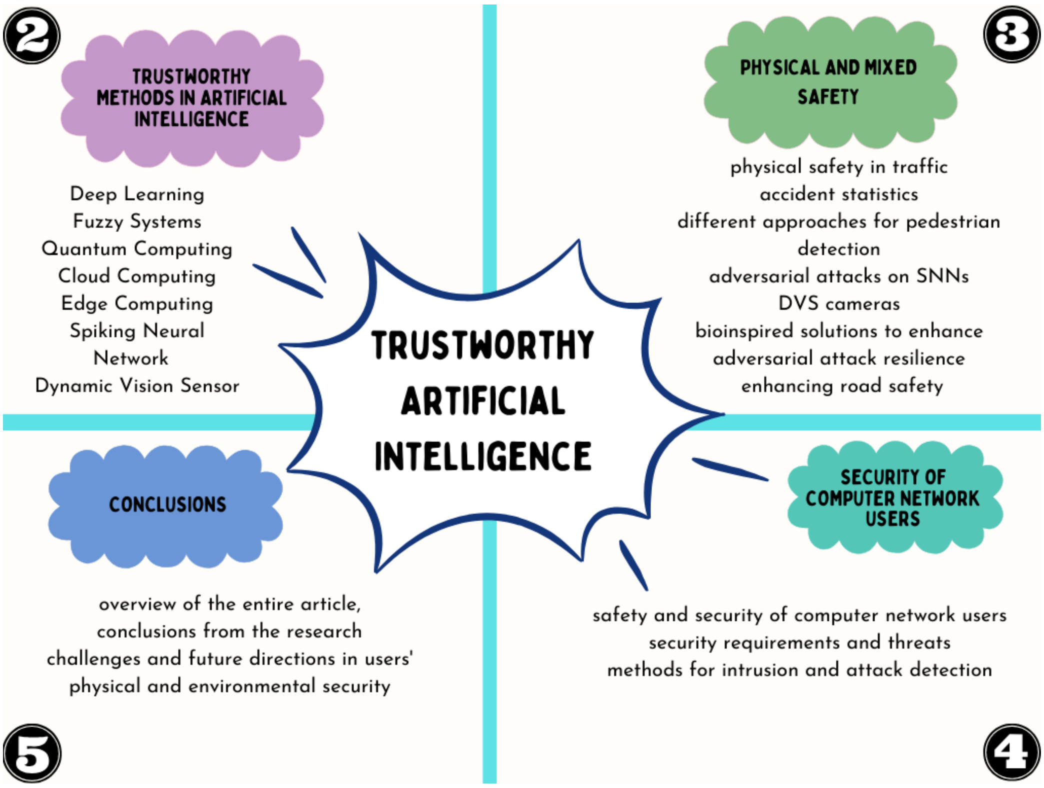
2
TRUSTWORTHY METHODS IN ARTIFICIAL INTELLIGENCE
Deep Learning
Fuzzy Systems
Quantum Computing
Cloud Computing
Edge Computing
Spiking Neural Network
Dynamic Vision Sensor
3
PHYSICAL AND MIXED SAFETY
physical safety in traffic
accident statistics
different approaches for pedestrian detection
adversarial attacks on SNNs
DVS cameras
bioinspired solutions to enhance adversarial attack resilience
enhancing road safety
TRUSTWORTHY ARTIFICIAL INTELLIGENCE
CONCLUSIONS
overview of the entire article,
conclusions from the research
challenges and future directions in users' physical and environmental security
5
SECURITY OF COMPUTER NETWORK USERS
safety and security of computer network users
security requirements and threats
methods for intrusion and attack detection
4

Figure 1. Visualization of the processed topics.

1.4. Acronyms

Table 1 summarizes this paper's acronyms and their explanation.

Table 1. List of acronyms used in this paper.

Acronym	Explanation	Acronym	Explanation
ANN	Artificial Neural Network	KNN	K-Nearest Neighbors algorithm
BL	Bayesian Learning	KPCA	Kernel Principal Component Analysis
BSM	Basic Safety Messages	LSB	Least-Significant Bit
(D)CNN	(Deep) Convolutional Neural Network	LSTM	Long Short-Term Memory
CPS	Cyber–Physical System	MITM	Man In The Middle attack
(D)DoS	(Distributed) Denial of Service attack	ML	Machine Learning
DL	Deep Learning	NB	Naive Bayes
DNN	Deep Neural Network	PSO	Particle Swarm Optimization
DRNN	Dense Random Neural Network	RNN	Recurrent Neural Network
DSU	Driving Scene Understanding	R-CNN	Region-based Convolutional Neural Network
DVS	Dynamic Vision Sensor	R-STDP	Reward-modulated Spike-Timing-Dependent Plasticity
(FF)CNN	(Feed-Forward) Convolutional Neural Network	SNDAE	Stacked Non-symmetric Deep Auto-Encoder
FL	Federated Learning	SNN	Spiking Neural Network
FPGA	Field Programmable Gate Array	SOTA	State-Of-The-Art
GA	Genetic Algorithm	SPP	Spatial Pyramid Pooling
GAN	Generative Adversarial Network	SSD	Single-Shot Detector
GNN	Graph Neural Network	SVM	Support Vector Machine
GRU	Gated Recurrent Unit	(T)AI	(Trustworthy) Artificial Intelligence
HOG	Histogram of Oriented Gradients	TA-SNN	Temporalwise Attention SNN
IDS	Intrusion-Detection System	ToM	Theory of Mind
IMU	Inertial Measurement Unit	UAV	Unnamed Aerial Vehicle
IoT	Internet of Things	VANET	Vehicular Ad Hoc Network
(I)RF	(Improved) Random Forest	WSN	Wireless Sensor Network

2. Trustworthy Methods in Artificial Intelligence

In this section, we present trustworthy methods in Artificial Intelligence. We briefly describe AI techniques like Deep Learning or Fuzzy Systems with their applications. Also, we explain how neural networks work. Next, we mention newer AI techniques like Quantum Computing, Cloud Computing, Edge Computing, and Spiking Neural Networks.

2.1. AI Methods and Their Applications

Artificial Intelligence is a tool that enables machines to learn from experience, adapt to new inputs, and perform human-like tasks. In recent years, AI has reached a significant level to ensure the practical functioning of many issues of collecting and analyzing helpful information.

When characterizing AI, first, one should start with Deep Learning (DL) [27]. DL is a Machine Learning (ML) technique that teaches computers to do what comes naturally to people, to learn by example. Countless developers use the latest innovative Deep Learning technologies to take their businesses to a new level. There are many areas of AI technology, such as autonomous vehicles, computer vision, automatic text generation, etc., where the scope and use of deep learning are growing.

A typical example of AI is neural networks with the ability to recognize objects, such as facial recognition [28]. These networks make it possible to recognize individual faces using biometric mapping. Such use has led to breakthrough advances in surveillance technologies, but has also been met with much criticism for breaching privacy. Offering legal agencies surveillance technology to monitor entire cities through a network of CCTV cameras and accurately assigning each citizen his/her real-time social credit score is not something that will be acceptable to the public.

This is different in the case of the use of AI in control and automation. AI can perform the same type of work repeatedly without fatigue. By the way, it is an ideal tool in the form of Fuzzy Systems [29], where relying on typically real values does not allow for correct control of machines. Automation increases productivity and results in lower overall costs and, in some cases, a safer working environment. One should also mention the appropriate organization of tasks. First of all, genetic and evolutionary algorithms complement the elements of AI, where, often in combination with neural networks, they are perfect for optimization issues.

Neural networks work well for obtaining various data. With each passing day, the data everyone produces grow exponentially. Rather than manually entering these data, networks allow you to collect and analyze them based on your past experiences [30]. Data acquisition is the transfer of knowledge from various sources to a data storage medium, where it is often accessed, used, and analyzed by various organizations. Often, data collection is preceded by edge processing. AI uses neural networks to analyze a large amount of such data and helps draw logical conclusions.

An example of the intensive use of AI algorithms in the form of neural networks (recursive and not only) for speech and text analysis can be found in Chatbot software v1 [31]. This is software that provides communication when solving customer problems by inputting audio or text data. Earlier bots only responded to specific commands, and the bot knew what the user meant if the user said the wrong thing. The bot had the capabilities that were implemented for it. The real change came when Chatbots were enhanced with AI algorithms that make it possible to understand the language, not just the commands themselves.

Another type of AI is hybrid solutions that form the basis of Quantum Computing. AI helps solve complex quantum physics problems with supercomputers' accuracy using Quantum Neural Networks [32]. This could lead to groundbreaking changes in the near future. It is an interdisciplinary field that focuses on building quantum algorithms to improve computational tasks within AI, including sub-domains such as Machine Learning. The entire concept of quantum-assisted AI algorithms remains in the domain of conceptual research.

Cloud Computing is another element of AI [33]. With so much data being transferred each day, storing the data in physical form would be a serious problem. ML functions operating in the Cloud Computing environment increase the efficiency of data organization, and in combination with Edge Computing, they significantly reduce the need for space for the crucial data stored.

2.2. Spiking Neural Networks

A Spiking Neural Network (SNN) is a more-biologically plausible version of an Artificial Neural Network (ANN). Neurons communicate through synapses using electrical pulses (spikes, action potentials)—instead of scalar values [34]—as almost all biological neurons do. Spikes are binary events that encode information in time and quantity (spike train). Hence, continuous time flow is necessary to process data with SNNs. There is no time step concept in SNNs as opposed to ANNs. However, the digital simulation of SNNs with general-purpose accelerators requires using a time step (as a time quantum for simulation), but this is a term coming from the numerical simulation domain.

One of the main goals of SNNs is to tremendously reduce the amount of energy required for training and inference. The human brain requires about 20 W of power to perform extremely complex computations [35]. Such low magnitudes of power consumption would not be possible when using CPU or General-Purpose GPU (GPGPU) calculations. Hence, the specialized neuromorphic devices are the native platforms for SNNs to execute. There have been many approaches for creating neuromorphic hardware, for example SpiNNaker, BrainScaleS, IBM TrueNorth, Intel Loihi, and Intel Loihi 2 [36,37].

Nevertheless, as mentioned earlier, SNNs can be simulated using general-purpose accelerators, which are not as energy efficient as specialized neuromorphic hardware, yet allow for inexpensive and elastic research on SNNs. There are a few frameworks and tools to perform such simulations—Intel LAVA [38], Nengo [39], Sandia Fugu [40], snnTorch [41], SpykeTorch [42], NEURON [43], NEST [44], BRIAN [45], CARLsim [46], and others [47].

Here, we reference some papers about SNN fundamentals, as furthermore, we focused mainly on TAI: [34,35,48–50].

3. Physical and Mixed Safety

This section considers the safety TAI dimension on a physical and mixed safety basis. Firstly, we focus on traffic and pedestrian safety and consider TAI-assessed systems for drivers. Next, we evaluate AI systems' robustness considering hardware robustness and software stability. Also, we examine adversarial attacks on AI methods.

3.1. Traffic Safety

Traffic safety is an increasingly important aspect of our lives. The number of automobiles in 2015 was 1.1-billion, resulting in heavy traffic. Vehicles travel at high speeds, which, combined with heavy traffic, results in more accidents, which result in injury and even death. New cars are installing systems that assist drivers and, sometimes, even replace them (full autonomy). We distinguish between five levels of autonomy in vehicles [51]. Level 0 means no auto automation. Such vehicles are still the largest number on the world's roads. With Level 1 driver assistance, the system can assist the driver. The vehicle can perform steering and acceleration in the Level 2 advanced driver-assistance system. Up to this level, it is considered that a human is responsible for monitoring the environment. In subsequent levels, the system monitors the vehicle's surroundings. The vehicle can perform most of the driver's tasks in Level 3 conditional automation. Level 4, high automation, the vehicle can perform all the driver's tasks under specific circumstances. Level 5 is full automation. The vehicle can perform all of the driver's tasks under any conditions. The installed and designed systems are based on multiple sensors to improve safety on the road. AI also plays a role here [52,53].

Traffic safety systems based on AI primarily rely on images and vehicle detection [54]. There are a number of preprocessing methods that increase the effectiveness of such detection. Vehicle detection can fall into the previously mentioned Level 1, and such a system can inform the driver about his/her surroundings in a limited way. For example, systems operating in difficult conditions related to visibility are a significant advantage here, serving to detect people or animals at night or in fog.

When vehicle detection is at a high level, we can extend the safety system to determine the trajectories of vehicles in the environment. If the system has such knowledge, it can determine potential collisions [55] with other traffic participants [56,57]. With such a system in place, we are already higher in the previously discussed automation model. A system that recognizes and analyzes the vehicle's surroundings can be assigned to Level 4, as it can monitor and analyze the environment. It can also determine the real danger to the vehicle being driven.

Systems that detect people at pedestrian crossings or intersections are also essential in vehicles.

3.2. Pedestrian Safety

Pedestrians are the least-protected road users. They do not have any safety measures in the case of a possible accident. Because of this, any traffic incident involving them is very dangerous. Therefore, this is a very important topic for road safety research. According to data from the World Health Organization (2021), there have been about 20–50-million fatalities and injuries in road accidents worldwide. As many as 19% involve fatal accidents involving pedestrians [58]. In Poland, 28,660 people were injured in 2021. Pedestrians accounted for 16.9% of all those injured. Based on the data [59], only 24.8% of all accidents involving pedestrians were their fault. This means that systems that detect people and their potential behavior can help prevent as many as 72.7% of accidents. If we look at the statistics for the location of accidents, it turns out that most, 71.9%, involve intersections and pedestrian crossings. This confirms the validity of the research and ANNs' continued improvements in detecting people. However, in this case, we have stricter requirements. They mainly rely on accuracy and real-time performance, as this directly impacts the field of autonomous vehicles and smart cities. The constant pursuit of the perfect detection of pedestrians and other objects has resulted in the development of different approaches to this topic [60].

The Convolutional Neural Network (CNN) approach will be discussed first. CNNs use image recognition and classification to detect objects, recognize faces, etc. The proposed improvement involves [61] automatically optimizing the feature representation to the detection task and regularizing the neural network. The accuracy of the Support Vector Machine (SVM) classifier using the features learned by the CNN is equivalent to the accuracy of the CNN, confirming the importance of automatically optimized features. However, the computational demand of the CNN classifier is more than an order of magnitude lower than that of the SVM, irrespective of the type of features used.

The next group discussed is the R-CNN, Fast R-CNN, and Faster R-CNN. The Region-based Convolutional Neural Network is based on the proposal of regions that are used to locate objects within the input image. Ross Girshick et al. [62,63] proposed extracting 2000 region proposals from images and processing them. The problem with such a solution is the long processing time of 2000 region proposals. It is impossible to work in real-time because processing a single image takes a few to tens of seconds. In addition, the selective search algorithm is a fixed algorithm. Therefore, we cannot improve the quality of the candidate region proposals. Based on this knowledge, the proposal to improve [64] the R-CNN was to modify this algorithm. The new approach of the region proposal method was based on the algorithm in [65]. Such a modification makes it possible to significantly improve the operation of the entire method. The next step for the R-CNN method was for the author to build a faster object-detection algorithm, and the new method was named Fast R-CNN. Here, we do not determine 2000 region proposals. Instead, the operation feeds the input image to the CNN to generate a convolutional feature map. Based on such a map, proposal regions are determined and are wrapped into squares. After transforming them to a fixed size, we feed them to the fully connected layer. We can successfully use this model to detect pedestrians [66]. One modification that improves the method’s pedestrian-detection capability is using the EdgeBoxes algorithm to generate effective region proposals from the image [67]. The quality of these regions significantly affects detection performance. Also added is a batch normalization layer between the convolutional layer and the activation function layer. Due to the wide variety of features in pedestrian detection in natural scenes, it was noted that a divide-and-conquer philosophy could mitigate them.

Based on this, the Scale-Aware Fast R-CNN (SAF R-CNN) method was proposed [68]. The model introduces multiple built-in subnetworks, which detect pedestrians with scales from disjoint ranges. Outputs from all of the subnetworks are then adaptively combined to generate the final detection results, which are robust to the large variance in instance scales via a gate function defined over the sizes of object proposals. The latest improvement to the R-CNN and Fast R-CNN algorithms eliminates the selective search algorithm for region proposals. Shaoqing Ren et al. developed an object-detection algorithm called Faster R-CNN [69], allowing the network to learn region proposals. The Faster R-CNN algorithm takes the entire image as the input to the convolutional network, exactly like the Fast R-CNN. Still, a separate network was implemented to predict region proposals instead of a selective algorithm.

An analysis of the Faster R-CNN algorithm for pedestrian detection was performed in the article [70]. The study showed the validity of using the Region Proposal Network. Another proposal to improve [71] Faster R-CNN was about improving the quality of the network and using K-means cluster analysis. The faster R-CNN method can detect people with a drone [72]. Based on this algorithm, a new FCF R-CNN model was developed [73]. It is based on feature fusion and context analysis. The proposed method has better results for pedestrians that are small in size and obscured, and it is also robust to difficult scenes.

In 2015, Redmon et al. proposed the first single-stage detector YOLO [74]. The first version had reduced detection accuracy relative to the two-stage detector, but significantly increased the detection speed. High speed and precision are crucial for pedestrians and unusual situations on the road. Subsequent versions of the YOLO network were prepared to improve detection quality. Variations containing specific improvements were also introduced for newer versions. The first of these concerns the YOLOv2 network [75,76]. The improvement of this network began with a modification of the DataNet53 model, in which feature creation was strengthened. In addition, three inception depth convolution modules were added and integrated at different levels. All these changes lead to a more-comprehensive characterization of the object in the image. The next version of the YOLOv3 network [77,78] uses the HOG method, which was implemented as preprocessing. This method makes it possible to highlight pedestrian contour features, especially small pedestrian target features, and reduce the interference caused by background information on the detection results. Another attempt to improve the YOLOv4 [79,80] network was based on implementing a modified detection model. The proposed model combines a new type of Spatial Pyramid Pooling (SPP) network and K-means clustering algorithm with the YOLOv4 model for easier feature extraction. In addition, the Mish activation function is applied to the neck of the detection model, replacing the Leaky ReLU activation function to improve detection performance.

Another improved pedestrian detection performance solution is an updated version of MobileNet [81] combined with the Single-Shot Detector (SSD) [82,83]. This method's four components are important for pedestrian detection, feature extraction, deformation, occlusion handling, and classification. The solution proposed in the paper allows the coordination of the components to increase their strength with a reduced number of parameters. The model achieved better results when tested on low-end edge devices than different versions of the YOLO network.

3.3. Trustworthy and Explainable SNNs

We decided to temporarily divide AI system robustness into three categories—from the low level to the highest: hardware robustness, software stability, and inference reliability. Often, AI robustness is investigated only for the last category—inference reliability. The general-purpose hardware robustness and software stability are tested and assured earlier in the software stack, by much broader use cases, with low-level tests performed not only for the AI domain. Hence, traditional AI, which uses general-purpose hardware and primarily general-purpose software, can be investigated mainly for inference robustness. However, SNNs are targeted to utilize custom, specialized hardware—neuromorphic devices. Therefore, hardware robustness and software stability can be considered in the context of TAI as a whole, together with inference reliability. As adversarial attacks on classic ANNs/CNNs are the often addressed issue, we discuss them separately in Section 3.4.

El-Sayed et al. [84] designed the Fault Simulation Framework for testing faulty behaviors of spiking Integrate-and-Fire (I&F) neuron analog model implementation. They considered standard transistor defect models, including stuck-on and stuck-off behaviors. For the passive elements (resistors and capacitors), the defect model includes variations of the parameters up to 50%. In summary, their defect model for a single analogue neuron took 46 different defects into account. They identified many defective behaviors, ranging from catastrophic states (non-functional neurons) to parametric defects (variations in spike timing compared to the nominal response). They defined a taxonomy for different neuron defects. Finally, the authors stated that these hardware-level errors can be readily reproduced on the network's behavioral level by direct changes to the model parameters.

Spyrou et al. [85] investigated neuron defects identified in El-Sayed et al.'s [84] experiments. They designed two SNNs for the classification task. The first was designed to classify digits from the N-MNIST [86] dataset, and the second was for gesture classification from IBM's DVS128 gesture dataset [87]. The first network performed with 98.08% classification accuracy and the second with 82.2% classification accuracy (with shortened samples). After the experiments, consisting of fault injection and implementing a fault repair mechanism, they proposed a neuron fault-tolerant strategy for SNNs and a fault-tolerant SNN architecture. The proposed techniques improved the overall classification accuracies. Their strategy consisted of a few mechanisms, including passive fault tolerance using dropout, active fault tolerance in hidden layers (offline self-test, an online self-test, a recovery mechanism), and active fault tolerance in the output layer.

In the next research, Spyrou et al. [88], instead of simulating an SNN, implemented it utilizing an FPGA as a neuromorphic device. Their SNN performs poker card symbol recognition tasks. The baseline for the recognition task (without fault injection) was 85 $\pm$ 2.5%. The fault injection engine flips the bits of different memory segments (splitter parameters, router parameters, neuron parameters, kernel parameters, kernel weights). After that, the accuracy results were compared to the baseline performance. Based on the comparison results, the authors pointed out the list of memory segments that should be especially protected against memory faults. These are splitter parameters, router parameters, and kernel parameters. Other parameters indicate some degree of fault tolerance for bit flips in Least-Significant Bits (LSBs). However, we suppose that these parameters' LSBs' resilience indicates that some variables could be represented with fewer bits.

Han and Han [89] in 2014 focused on a robust human detection system in all weather conditions. They proposed their neuromorphic vision solution, an alternative to already existing algorithms. Although they did not use the Dynamic Vision Sensor (DVS) (an explanation is given further in this subsection), their system was inspired by the primary visual cortex (V1), and its neurons respond specifically to differently oriented lines. The system provides robustness for detecting humans mostly occluded with clothing, people wearing helmets, where the rest of the body is not visible, and human detection in rainy weather or in low-light conditions. They also proposed a solution for neuromorphic stereovision processing. Finally, they proved their system performance by achieving a 95% pedestrian detection rate and showing robustness in detecting bike riders against dark and wet conditions. The next year, in 2015, Han and Han [90] proposed a system with a 99% successful detection rate for its new use case.

Zooming our considerations out from a particular vehicle or agent, Prez et al. [91] utilized the SVM and SNN as two alternatives for predicting crowd movement trajectory, which is critical for many physical safety applications. They simulated crowd movement at the microscopic level (interaction between individual agents). After comparing the two alternatives, it became clear that the SNN performed better than the SVM in predicting crowd behavior. Although the authors assumed that using methodologies developed from SVM could provide better results than the currently obtained, they also stated that it should still be possible to achieve better performance with SNNs in future work.

To a certain extent, crowd movement is related to Driving Scene Understanding (DSU), as people's movements can be a crucial part of DSU. DSU aims to understand ongoing on-road driving scenarios [92]. Gaurav, in his thesis [93], focused on DSU, using an SNN (by learning the ANN and converting it to the SNN). To train his 3D-CNN network, he used visual data only from the Honda Research Institute Driving Dataset (HDD) [92]. The best mean result for the goal-oriented action layer was achieved by the SNN using the "True Max U" method as a counterpart for the MaxPooling layer. The converted network was deployed on the GPU SNN simulator. The best mean result for the cause-oriented layer—which heads up towards explainable AI—was achieved by using a nonspiking ReLU ANN. He also proposed two new methods for performing a neuromorphic-friendly implementation of the MaxPooling layer. He proved their efficacies on the Intel Loihi neuromorphic chip using the Nengo, NengoDL, and NengoLoihi libraries.

Zhao et al. [94] remarked that AI systems often interact with multiple agents, which leads to safety risks (physical safety in that context) arising from such interactions. They proposed a solution for AI agents interfacing with others, a use case for the Theory-of-Mind-based SNN (ToM-SNN). Their ToM-SNN was trained using Reward-modulated Spike-Timing-Dependent Plasticity (R-STDP) [95,96]. Their SNN aims to recognize other agents' risky mental states and help them when necessary. In a series of experiments performed in a simple grid world, their agent helped others avoid safety risks (rescue behavior chosen). It recognizes others' behavior policies and behaves differently for different agents. Hence, the mentioned paper can be crucial for DSU systems and AI development towards TAI.

However, the authors stated that much work is needed to scale their ToM-SNN model. They targeted creating a model inspired by the mirror neuron system, the model that understands others' actions and beyond that.

Guarav's and Zhao's solutions, mentioned above, could be beneficial also for enhancing environmental safety at mass events, e.g., as the agents enhancing the functionality of mass event security systems. After scaling up the mentioned solutions, they could predict massive crowd movements at events and detect riskily behaving individuals. Hence, our main concerns about implementing the solutions in such applications are the low availability of neuromorphic processors, which are necessary for the low energy demand of such systems, and the need for scaling up the proposed solutions, which are quite promising.

We found few papers focusing on the explainable SNN field, which signalizes that it is still a poorly developed area in the SNNs domain. Nguyen, in her Master's thesis [97], proposed the Temporal Spike Attribution (TSA) method, which allows for a better understanding of the SNN's outcome. Next, Kim and Panda [98] basically proposed a tool that produces a heatmap of the input shape, showing an attention map for the input data, named Spike Activation Map (SAM). The authors claimed that this is the beginning (in 2021) of the "explainable neuromorphic computing" area that is leading to increased trust in SNNs' outcomes. Finally, Seras et al. (2022) [99] focused on detecting abnormal input data by applying the Out-of-Distribution (OoD) data detector. The OoD input is the input data that the model was never trained with. Hence, it can produce untrustworthy outcomes. Their detector helps to expose the OoD input by observing hidden layers of the SNN. Moreover, the detector is able to indicate which part of the input is the most-atypical, yielding the map of the input abnormality. The authors compared their work to other OoD detection schemes, claiming that their method is competitive with them.

Further, we explored DVS safety and robustness solutions. A DVS is a device that detects scene illumination changes. A DVS is also called a silicon retina or event camera. Each pixel of a device's matrix reacts to changes in its field of view. The sensor produces a stream of asynchronous events: it elicits an event at the moment of brightness change detection. A particular event encodes a few pieces of information, i.e., time, pixel location, and the sign of the brightness change.

In contrast to traditional cameras, DVSs have some attractive properties—instantaneous reaction for the changes in the scene (temporal resolution in the order of µs instead of the order of many ms in traditional cameras), high dynamic ranges (up to 140 dB instead of ∼60 dB), and low energy consumption. More details on DVSs were given in the survey by Gallego et al. [100]. Event sensors are a kind of native source of information for SNNs, due to their event asynchronous nature, similar to neuronal spikes.

Zhang et al. [101] explored three aspects of processing events from the DVS cameras with SNNs. First was a conversion of a large-scale CNN to the SNN without losing too much precision. The solutions they proposed for conversion between network types can be generalized to convert other DNNs into SNNs. Secondly, they transformed the Cityscapes dataset [102] into two different representations: event-processing mode and contrast-detection mode. Lastly, they constructed a 3D-structured-light-acquisition system and 3D-image-recognition algorithm, using a DVS camera from Prophesee, Model PSEE300EVK. The authors stated that their algorithm achieved good generalization results.

Yao et al. [103] proposed the Temporalwise Attention SNN (TA-SNN) to classify event streams coming from DVS cameras with higher accuracy. The authors observed that event streams include many redundant events in the temporal dimension. Hence, their TA module estimates how much attention the SNN should pay to the current, temporarily created time frame. The time frame was constructed from events belonging to a particular time window. In the inference mode, the attention value must surpass a certain threshold for further event processing (utilizing the SNN). Otherwise, the processing is skipped. In the training mode, the attention mechanism weights inflowing time frames. The authors tested their TA-SNN with three different tasks, i.e., gesture recognition (using the DVS128 Gesture dataset [87], achieving 98.61% accuracy), image classification (using the CIFAR10-

DVS dataset [104], achieving 72.00% accuracy), and spoken digit recognition (using the Spiking Heidelberg Digits dataset [105], achieving 91.08% accuracy). The authors stated that their attention mechanism did not impact accuracy significantly, but the reduction in the number of events significantly reduced the amount of computation.

Liu et al. [106] remarked that most current (in 2021) SNNs for processing event streams focus on object-recognition tasks. The authors proposed a hierarchical SNN to recognize actions using motion information. Their SNN was equipped with a motion-perception layer and a motion-pooling layer, consisting of motion-sensitive neurons. Neurons from the motion-pooling layer have a larger receptive field than neurons from the motion-perception layer. However, both layers were characterized by the same neural dynamics. The SNN consisted of five spiking layers, including the input and classifier layers. The solution was evaluated with three datasets, i.e., the DailyAction-DVS dataset [106], the DVS128 Gesture dataset [87], and the Action Recognition dataset [107]. For the DailyAction-DVS dataset—which could help recognize human action for safety applications—the proposed solution achieved 90.3% accuracy. For the DVS128 Gesture and Action Recognition datasets, the achieved accuracies were 92.7% and 78.1%, respectively. The accuracies achieved for the datasets were better than the accuracies from other papers quoted by the authors.

Instead of detecting or classifying objects or actions, Salah et al. [108] focused on a robust relative localization system, utilizing a DVS camera, Inertial Measurement Unit (IMU) sensor, and stationary flickering LEDs as landmarks. However, they did not utilize SNNs in this work. The authors tested and evaluated their system by mounting it to a small Unnamed Aerial Vehicle (UAV) and comparing the localization results with the Optitrack Prime 13 motion-capture system, a ground truth for localization. They achieved a maximum positioning error equal to 0.0137 m, a mean positioning error of 0.0052 m, a maximum orientation error of 2.16°, a mean orientation error of 0.567°, and a relative positioning error of 0.074% at a 7 m range. The test was performed indoors. The authors stated that the proposed system outperformed SOTA methods in terms of localization accuracy and execution time.

Table 2 summarizes all datasets mentioned in this section related to SNNs and DVS processing. The first column references the dataset; the second column contains the dataset name; the third column points to papers described in this subsection and utilizing the dataset; the fourth column briefly describes the dataset; the fifth column references the dataset download page.

Table 2. Summary of the datasets related to SNNs and spike processing.

Ref.	Name	Papers Utilizing	Description	Total Samples	Download
[86]	N-MNIST	[85,109]	The spiking version of the MNIST dataset. It consists of handwritten digits. The dataset is generated with a DVS camera and consists of 70,000 samples.	70 k	https://www.garrickorchard.com/datasets/n-mnist (accessed on 16 July 2023)
[87]	DVS128 Gesture	[85,103,106,109]	A spiking dataset was recorded with a DVS camera, comprising 11 hand gesture categories, under 3 different illumination conditions, with 29 subjects. The camera resolution is 128 × 128 spiking pixels. The recordings are 6 s long on average [103]. There are 1342 recordings in total.	1342	https://research.ibm.com/interactive/dvsgesture/ (accessed on 17 July 2023)
[92]	Honda Research Institute Driving Dataset (HDD)	[93]	The dataset was created for the Driving Scene Understanding task realization. Consists of 104 h of driving data, comprising three video cameras (1920 × 1200 px, 30 fps), Velodyne HDL-64E S2 3D LiDAR, GeneSys Elektronik GmbH Automotive Dynamic Motion Analyzer, and the vehicle's CAN data (throttle angle, brake pressure, steering angle, yaw rate, and speed).	104 h	https://usa.honda-ri.com/HDD (accessed on 18 July 2023)

Table 2. *Cont.*

Ref.	Name	Papers Utilizing	Description	Total Samples	Download
[102]	Cityscapes	[101]	The dataset consists of 25,000 annotated (segmentation) frames coming from driving scenes. There are 5000 frames annotated on the pixel level, and 20,000 are weakly annotated (with polygons). The frames are annotated with 30 classes, recorded in different months in 50 cities (mainly Germany). Each annotated frame is preceded and followed by non-annotated frames, supplied with stereo frames, global coordinates, data from odometry, and an outside thermometer.	25 k	https://www.cityscapes-dataset.com/downloads/ (accessed on 20 July 2023)
[104]	CIFAR10-DVS	[103]	The dataset is the CIFAR10 dataset converted by the DVS camera (at a resolution 128 × 128 spiking px) to event representation. There are 10 classes (animals and vehicles), 1000 recordings per class, and 10,000 recordings overall. The images were upscaled, displayed on the LCD monitor with a circular movement, and recorded with the DVS.	10 k	https://figshare.com/articles/dataset/CIFAR10-DVS_New/4724671/2 (accessed on 21 July 2023)
[105]	Spiking Heidelberg Digits (SHDs)	[103]	An audio spiking dataset. It consists of ∼10,000 high-quality audio recordings. The words are pronounced by 12 distinct speakers and converted to spiking representation to 700 spiking channels. There are 20 classes—digits from 0 to 9 spoken in English and German.	∼10 k	https://zenkelab.org/resources/spiking-heidelberg-datasets-shd/ (accessed on 22 July 2023)
[106]	DailyAction-DVS	[106]	The dataset comprises 1440 DVS recordings of 12 different activities (12 classes). There are 2 illuminations, 15 actors, and 128 × 128 spiking px, and each recording is about 6 s long.	1440	https://github.com/qianhuiliu/SNN-action-recognition (accessed on 23 July 2023)
[107]	Action Recognition	[106]	The dataset consists of 450 DVS recordings of 10 different human actions, acted by 15 subjects with an average recording length of 5 s. Recorded at different positions and distances from the subjects. The sensor resolution is 346 × 260 spiking px.	450	https://github.com/CrystalMiaoshu/PAFBenchmark (accessed on 25 July 2023)

3.4. Adversarial Attacks on CNNs, SNNs, and Neuro-Inspired Solutions Against Them

The adversarial attack occurs when the attacker corrupts the samples on which the ML algorithm is trained or corrupts the inferred samples. The first type—corrupting training samples—is called the causative or poisoning attack. It targets perturbing the original data distribution. The second type—corrupting inferred samples—is called the evasive or exploratory attack. Usually, this kind of attack targets the misclassification (often with a high confidence level) or exploration of the model properties and behaviors [110].

Dapello et al. [111] remarked that current (2020) SOTA CNNs are loosely inspired by the primate visual system. With the adversarial attack, these CNNs can be fooled by adding a low noise to the image or sample, more generally. The perturbations are modest—imperceptible or almost imperceptible for the naked human eye—but cause misclassifications in the CNNs. As a solution for that problem, the authors proposed a new class of models—VOneNet—a hybrid CNN vision model. Each VOneNet model consists of two blocks, i.e., VOneBlock, and a classic adapted CNN. The VOneBlock is heavily inspired by the primate visual cortex and serves as an input of VOneNet. It is a hardwired set of Gabor filters, then a non-linear layer and a stochastic layer. No learning occurs in that block. After the VOneBlock, there is a standard adapted and trainable CNN. The authors created a VOneResNet50 model (a specific model of the VOneNet class) and tested it against the classic ResNet50 CNN to test the performance of their solution. The VOneResNet50 achieved better overall results than the standard ResNet50 in classifying perturbed images (∼54.3% for VOneResNet50 against ∼43.6%). The results of the clean images were slightly worse than what ResNet50 achieved (∼71.7% for VOneResNet50 against ∼75.6%).

Branytskyi et al. [112], inspired by VOneNet (Dapello et al. [111]), proposed a VOneGAN. Similarly, as in the VOneNet, the architecture of the VOneGAN consists of the bioinspired input block and standard trainable model. The input block, as VOneNet's, also comprises Gabor filters, a non-linear and stochastic layer. The authors stated that the proposed solution improved the training stability and quality of the produced visual content.

Shi et al. [113] proposed a different solution than VOneNet for resilience to adversarial attacks. The authors propose the Visual Attention from Recurrent Sparse reconstruction (VARS). The solution is inspired by the human visual system and its attention mechanism, which groups the areas of the field of view into objects, neglects nonsignificant objects (or noises), and selects the most-significant objects. In addition, VARS improves its attention maps during recurrent updates. VARS can be plugged into neural networks as an attention mechanism. The researchers tested VARS against five robustness benchmarks. In each benchmark, VARS (when based on the RVT [114] network design) performed better or similar to previous methods. The authors released their code at https://github.com/bfshi/VARS (accessed on 10 June 2023).

Heading up toward SNNs and DVS cameras, some works propose solutions for resilience to adversarial attacks. Marchisio et al. [109] focused on designing adversarial attacks for the DVS cameras. They proposed five types of attack, i.e., sparse, frame, corner, dash, and MF-Aware dash attacks, and successfully performed the attacks, causing accuracy drops in the classification task performed by the SNN on two datasets, i.e., DVS128 Gesture [87] and N-MNIST [86]. They proposed applying the activity filter and the mask filter on the outcoming event stream. The authors remarked that the applied filters cannot completely defend against proposed adversarial attacks, especially against the MF-Aware dash attack. Despite the applied filters, the attack caused at least a 20% accuracy drop for the DVS128 Gesture spiking classifier and at least a 65% drop for the N-MNIST spiking classifier.

Krithivasan et al. [115] revealed the adversarial Native SpikeAttack and Proxy SpikeAttack on SNNs. The attacks target bumping up the energy consumption and increasing the latency instead of fooling the SNN. Using the proposed attack methods, which increased the overall spike number to 2.5×, the authors bumped the energy consumption up to 2.3× and increased the latency to 2.2×. Eventually, the authors proposed three defense strategies, i.e., input quantization (lowering the precision of chosen states and parameters), spike dropout (dropping random spikes), and threshold and leak modulation (raising the spike threshold and membrane leaking rate, which leads to potentially improved sparsity). They allowed only for less than a 0.5% accuracy drop while evaluating the effectiveness of the defense methods, achieving promising results. For almost each tested case, the best was the spike dropout strategy, which often significantly reduced the excessive spike activity.

EL-Allami et al. [116] considered the impact of the structural parameters of spiking neurons on the resilience against white-box adversarial attacks. They proposed a methodology for testing the robustness of an SNN and remarked that security studies show that SNNs can be robust in their design by properly tuning the neurons' structural parameters. Finally, they designed trustworthy SNNs by scanning the space of the chosen parameters and finding sweet spots for achieving the best robustness. The baseline accuracy for a properly tuned SNN was around 95–98% for a broad space of structural parameter values for the MNIST dataset. After performing the attacks, the performance drastically dropped in a wide space of SNN parameters. However, the authors found a parametric sweet spot, where the accuracy dropped only from 98% to 84% after the attack. This is a splendid result compared to the rest of the parameter space, where the achieved accuracies were predominantly miserable, reaching even as low as ∼0.0%. The authors published their research source code at https://github.com/rda-ela/SNN-Adversarial-Attacks (accessed on 10 June 2023).

Nomura et al. [117] considered Time-To-First-Spike (TTFS) encoding as a resilience method to adversarial attacks on SNNs. They trained their SNN using backpropagation and explored its resistance to adversarial attack. They tested different temporal penalty values to achieve the best result. Then, the results were compared to the standard ANN adversarial attack resistance. The achieved accuracy was significantly higher for the properly tweaked SNN than for the classic ANN. The authors observed that the achieved accuracies for clean samples (non-attacking ones) were slightly better for the ANN than the SNN. The authors concluded that the results suggest that SNNs mirror some aspects of human vision/recognition.

Kim et al. [118] analyzed different SNN encodings to defend against adversarial attacks. They compared rate coding and direct coding, paying attention to the accuracy, the adversarial attacks' robustness, and energy efficiency. They evaluated three different network architectures on three datasets, using backpropagation to train the networks. Using two different types of adversarial attacks, they remarked that rate coding had up to 20% higher resilience to adversarial attacks than direct coding. In terms of accuracy, the directly coded SNNs achieved better results. However, the accuracy gap between the two varied, depending on a few parameters, i.e., the complexity of the SNN and the number of training time steps. The authors used the 16 bit Eyeriss platform to estimate the energy efficiency for both encodings. Overall, the rate coding was less energy intensive on the used platform than direct coding (roughly 50% less energy). Finally, the authors suggested choosing a coding method according to a specific application.

4. Security of Computer Network Users

The security of IoT users is a comprehensive and constant issue. Users of such networks can use banking or training platforms, Internet messengers (Messenger, Signal, WhatsApp), social networks (Facebook, Twitter), intelligent devices, or networks of vehicles. Additionally, during communication, users use various devices, and each communication involves transferring information between the nodes that make up the network. Messages may contain sensitive user data. Hence, there is a need to secure these data using cryptographic techniques such as encryption, hashing, security protocols, or authentication and key management algorithms [119–121].

While implementing security measures in the IoT systems, the so-called Cybersecurity triad (CIA triad) allows for managing the security policy in a computer network. The CIA triad oscillates around three aspects essential to securing communications. The first aspect is confidentiality, which ensures that there will be no unauthorized attempts to access confidential information. Integrity is the second aspect, providing data consistency, accuracy, and reliability. The last aspect is availability, which guarantees consistency and easy access to the data for authorized parties [122].

Thus, we should secure data transmission between devices using encryption protocols such as SSL/TLS to prepare an appropriately secured IoT network. Also, we should require strong authentication mechanisms for all devices to prevent unauthorized access, which means that only authorized devices will have access to the network. Next, we should use the principle of the least-privilege management model to mitigate potential risks. Thanks to this, we ensure that users have the appropriate permissions to access individual devices and data in the IoT network [123,124].

IoT systems and WSNs are vulnerable to cyber attacks inside and outside the network. There are many classifications (e.g., routing-based, characteristic-based, based on the action layer of the communication protocol) and types of cyber attacks. A cyber attack may result in data loss, interception, or modification. The most-common attacks in these solutions are the Man In The Middle attack (MITM), stolen or guessing attacks, capture or impersonation attacks [125], zero-day attacks, malware [25], or (Distributed) Denial of Service attacks [126].

During the Man In The Middle attack, the aggressor tries to disrupt communication between two end systems by injecting a malicious node between legitimate nodes or attacking communication protocols in IoT systems. The duplicated packets can be sent multiple times to the recipient [125,127]. In stolen or guessing attacks, the aggressor can guess the password when the smart card, device, or other verifier is lost or stolen from the server. After that, the aggressor uses the so-called offline password guessing technique due to the fixed verifier output value for the same input [125]. A capture attack refers to a sensor node or device. The aggressor hijacks such an element to take over the network. Next, he/she removes it from the network. During the last step of the capture attack, the aggressor redeploys the sensor node or device as a malicious node [125,128]. Impersonation attacks refer to situations when the aggressor uses the identity of a different user, device, or another network element during communication. Also, the aggressor can install the client's certificate on such a device and next impersonate them [129]. The Denial of Service attack (DoS) consists of overloading the application serving specific data or handling customer data, i.e., exhausting its resources. If the aggressor uses computers in many places (called botnets [130]) during such an attack, he/she performs a Distributed Denial of Service attack (DDoS) [126,131]. In zero-day attacks, the aggressor tries to use the vulnerability before software developers can find and fix it [132].

One of the security elements of IoT or WSN solutions is systems that detect intruders and their intrusions into the network. Intrusion-Detection Systems (IDSs) are designed to scan the network and the devices that make it up. Then, these systems analyze and evaluate the collected information about the network activity. On this basis, they generate alarms and react if they encounter suspicious activity. IDSs can be both hardware and software systems. IDSs can be implemented using a variety of techniques. The effectiveness of such systems depends on the implemented decision engine. To make the system more versatile and effective, it applies DL and ML algorithms, which can adapt to the constantly evolving types of network attacks and methods of carrying them out [133–136]. Figure 2 shows the typical architecture of a CPS with an AI-based Intrusion-Detection System. The border router forwards network traffic to an Intrusion-Detection System. This system analyzes received data using an implemented AI model.

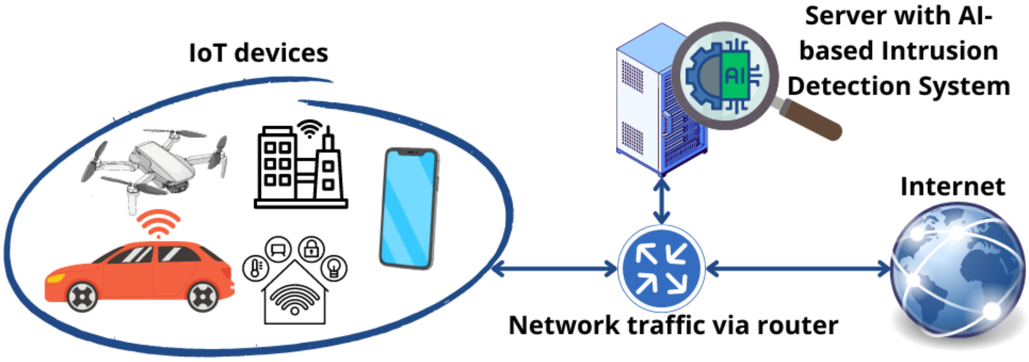
Server with AI-based Intrusion Detection System
AI
IoT devices
Internet
Network traffic via router

Figure 2. Typical architecture of a CPS with an AI-based Intrusion-Detection System.

4.1. TAI-Based Securing Against Intrusion Attacks

In this subsection, we present an overview of research connected with Trustworthy Artificial Intelligence methods for the security of computer network users. We considered AI-based Intrusion-Detection Systems and AI-based methods for specific attack detection. Also, we discuss these methods and conclude our insights about the mentioned research. The security of network users is connected with the safety dimension of TAI.

In [137], Kumar et al. proposed a deep blockchain-based trustworthy privacy-preserving secured framework for a cyber–physical system network called DBTP2SF. The proposed system was designed to meet the requirements of security, privacy, centralization, trust, and integrity in interconnected device environments. DBTP2SF has a modular structure. The first module is responsible for credibility and trust management, where observational sensor data are assessed to generate a reputation score of a given network node. The privacy-protection module maintains data integrity and transforms and generates data models. In turn, the anomaly-detection module uses Deep Neural Networks (DNNs) consisting of a inputlayer, output layer, and hidden layer. The implemented network processes the data collected by the sensors and analyzed by the first two modules. The DNN network then evaluates the behavior of the data and divides them into two categories, regular activities and attacks. When an attack is detected, data are sent to the system administrator for him/her to take appropriate action. The authors used publicly available datasets to validate the proposed system. The data contained events related to various attacks that may occur in WSNs (for example, MITM and guessing attacks). The authors measured the performance of the implemented DNN and anomaly-detection system using the following metrics: accuracy, precision, detection rate (recall), and F1-score [134,138]. While analyzing the efficiency of the intrusion-detection process, the authors obtained an accuracy of 98.97% and a detection rate of 93.87%. Kumar et al. concluded that intrusion detection using the proposed DNN ensures the security of network users because it protects them against intrusion into the network.

Gu et al. in [139] noticed that data quality also influences the effectiveness of IDS intrusion-detection methods. High-quality training data will increase detection performance and reduce training complexity and response time. They proposed classifying intrusions and regular instances in their solution, which uses the SVM and Naive Bayes (NB) feature embedding. The authors used the NB algorithm to transform the original features into new data to obtain high-quality data. Then, the obtained data trained the SVM classification model. The authors used publicly available datasets for intrusion detection to research and evaluate the proposed model. The authors achieved an accuracy that oscillated between 93.75% and 99.35% for different datasets. Gu et al. summarized that such features characterize their proposed method as having high precision, a high detection rate, a low false alarm rate, and a rapid training speed.

Ravi et al. in [140] proposed a comprehensive model of network attack detection and classification of network attacks. The authors developed a system containing several modules based on DL. To extract the features of a Recursive Deep Learning model, the authors used Recurrent Neural Networks, Long Short-Term Memory (LSTM) networks, and the Gated Recurrent Unit (GRU) to extract the optimal functions necessary to detect and classify attacks. The authors used Kernel Principal Component Analysis (KPCA) to reduce dimensionality without data loss. The authors then passed the combined features to the collective meta-classifier, which used the Random Forest (RF) and the SVM model for prediction. The last stage of the system's operation was detecting and classifying network attacks, which were performed using logistic regression. The authors tested the proposed method using publicly available test datasets, and better results were obtained during the tests than existing solutions for the same test data. The authors concluded that the proposed solution could be used to monitor network traffic effectively and warn about possible attacks in real-time.

Wang et al. in [141] adapted the LightGBM [142] algorithm to an intruder detection system called AI@NTDS. The LightGBM uses the following characteristics: a histogram algorithm, Gradient-based One-Side Sampling, Exclusive Feature Bundling, and Leafwise Tree Growth. Gradient-based One-Side Sampling uses low-gradient random sampling and introduces a constant low-gradient weight. Such an assumption allows for an increase in the efficiency and scalability of the operation of the entire algorithm. The authors compared the proposed method with similar solutions, obtaining better results and demonstrating the effectiveness of the LightGBM algorithm in detecting and classifying attacks by intruders. They concluded that the proposed system effectively detects threats related to remote network connections. Also, Leevy et al. in [143,144] considered the LightGBM algorithm in their intruder detection system.

Latif et al. in [145] proposed a lightweight, Dense Random Neural Network (DRNN) for intrusion detection dedicated to IoT solutions. The network model consists of dense clusters that can communicate and perform interactions similar to the interactions between brain cells. The model assumes one input layer, four hidden cluster layers, and one output layer. The communication between hidden layers is based on a Multi-Layer Feed-Forward architecture. The proposed network was tested for effectiveness and efficiency. The results showed that the proposed model correctly classified nine typical IoT attacks with high detection accuracy.

Cai et al. in [146] proposed a Hybrid Parallel Deep Learning model for intrusion detection based on Metric Learning. The proposed system consists of several elements. The first is the feature-extraction module, which extracts network traffic features and reduces redundant features. The system then creates two parallel CNN architectures to combine their spatial features further. In the next step, two parallel LSTM architectures analyze the temporal information of the combined features. In the last stage of the system operation, the CosMargin classifier classifies the distinguished space–time features. The authors compared their model with similar solutions and achieved high classification accuracy.

In [147], Balyan et al. proposed a hybrid-network-based IDS model that uses an enhanced Genetic Algorithm (GA), Particle Swarm Optimization, and Improved Random Forest methods (IRF). In the proposed model, the Particle Swarm Optimization method improves the decision vectors, while the GA is used to reconfigure them using evolutionary operators. In addition, the algorithm selects the best features, thereby minimizing the number of dimensions, increasing the true positive rate and lowering the false positive rate. In turn, the IRF eliminates fewer essential attributes, activates the list of decision trees in each iteration process, and oversees the operation of the RF classifier. The authors achieved high performance of the proposed system by testing its operation using a well-known dataset for IDSs.

Alsarhan et al. in [148] considered ensuring safe travel with intelligent vehicles and roads that create the so-called Vehicular Ad Hoc Networks (VANETs). These networks are exposed to attacks in which intercepted nodes modify or reject security messages. The detection of this type of intrusion is important in VANETs because the lack of such protection may affect people's health and lives. The authors divided the intrusion-detection scheme into four stages. The first is the security-risk filter based on rules, which is responsible for determining whether the observed event of the node deviates from the normal profile. The authors used Density-based Spatial Clustering of Applications with Noise in this stage. In the second step, an adder used the rules of Dempster–Shafer theory [149]. The adder computes an overall belief value by fusing all sources of evidence for each activity. Each activity is recorded in the node's event history database, which considers both the node's good and destructive activities. Saving to the database is the fourth stage of the proposed system. The final step is related to the use of Bayesian Learning. The authors used Bayesian Learning (BL) to update the probability of a belief with new activities. Alsarhan et al. determined the suspicion level of all incoming data based on the degree to which they deviated from the data reported from trustworthy nodes. The authors simulated the system using the Markov Modulated Poisson Process and two Gaussian distribution functions. The results showed that the proposed solution was characterized by high precision and accuracy in detecting intrusion into the VANET. Thus, implementing such a system will protect the network and the devices that compose it against attacks, thus protecting human life.

The problem of detecting attacks in vehicle networks was also dealt with by Ullah et al. [150,151]. The authors observed that any attack on intelligent vehicle networks could seriously disrupt vehicle communication and become dangerous to human life. Ullah et al. proposed a hybrid model of an intruder-detection system based on DL to improve the security of such systems. This model combines LSTM and the GRU. The LSTM processes the input data. Between the LSTM and GRU, a DENSE layer ensures quick system responses. The GRU, on the other hand, generates an exit probability. The results of the tests of the proposed model showed its better performance, including a shorter response time than other models. The authors confirmed that the model could detect different types of cyber attacks.

Also, Driss et al. focused on attacks occurring in vehicle networks. In [152], a Federated Learning-based (FL) technique was proposed to detect cyber attacks in the mentioned networks. The model uses the GRU to observe network traffic, considering the temporal relationship between traffic samples, regulating the learning process, and detecting unknown attack patterns. Learning happens asynchronously. Each network node independently performs the training algorithm with its copy of the dataset and shares the weights of the locally learned model with the aggregating instance. The authors tested the proposed system with publicly available test data and obtained promising results.

Hu et al. in [153] considered the security and intrusion-detection problems in autonomous vehicle communication using a vehicle-mounted network. The authors proposed a new CNN model based on two-dimensional Mosaic pattern coding to detect intrusion in these networks. The method proposed by Hu et al. consists of two processes. The first is coding based on 2D Mosaic patterns. During this process, one-dimensional data are converted to two-dimensional form and then packed to be joined into a Mosaic pattern. The second process is related to the operation of the CNN, which consists of the input, convolutional, pooling, fully connected, and output layers. The lower layers process the network traffic. However, the output layer contains only two neurons connected to the neurons of the previous layer. This layer classifies incoming network data into normal and attacking network traffic. The authors simulated their method and showed that the proposed method effectively distinguishes attacking traffic from ordinary information flowing in the network of vehicles. Additionally, the technique enables real-time intrusion detection and improves the reliability of system differentiation.

Roy et al. in [154,155] proposed a B-Stacking Intrusion-Detection System for IoT networks. Their method used the K-Nearest Neighbors (KNN), RF, and XGBoost algorithms to detect intrusions and anomalies in the network. The authors tested their system using publicly available datasets. Thanks to the obtained results, the authors confirmed a high detection rate for less popular cyber attacks with infrequent observations, for example the user to root attack, during which attackers gain access to the root by exploiting the system's vulnerabilities by logging as a regular user. The authors obtained lower false positive and false negative rates than other intrusion-detection methods.

Qazi et al. in [156] proposed an automatic IDS using a DL approach. They considered a new unsupervised technique connected with SVM classification: Stacked Non-symmetric Deep Auto-Encoder (SNDAE). This connection increased the detection reliability and decreased the computational requirements and the training and computing costs. The authors performed a complex analysis of their approach using publicly available datasets. The authors obtained an overall accuracy of 99.65% and a precision of 99.99%. Both results were better during the comparison with similar techniques. The authors concluded that the following research would consider a system's ability to manage zero-day attacks.

In [157], Nguyen et al. proposed Realguard, an Intrusion-Detection System based on a Deep Neural Network (DNN). The proposed approach can operate in the edge gateway of a network of connected devices to distinguish between normal and attacking network traffic and attacks. The proposed DNN model includes five hidden layers, where each neuron from one layer connects with neurons from the next layer. In addition, the network considers over 34,000 parameters, which translates into a high efficiency of intrusion detection. The network extracts the characteristics of the network traffic. Thus, it detects ten types of attacks, such as MITM or denial of service attacks. The authors tested the operation of the proposed system on a publicly available dataset. The obtained results showed the high precision and accuracy of the system with a simultaneous low rate of false alarms.

Yang et al. in [158] proposed a lightweight and effective intrusion-detection method based on cloud–edge collaboration. The authors used an auto-encoder, a temporal convolutional network, and a federated learning framework in this method. It enabled the dimensionality reduction of raw network traffic data's high-dimensional features. In comparing the proposed model with Long Short-Term Memory and the Gated Recurrent Unit, the authors achieved better results (accuracy of 98.62%). The proposed method successfully detected botnets in the network, brute force attacks, or DoS attacks. They concluded that their method could reduce the computation and storage requirements. Also, it deals with the issue of edge device resource limitations.

In [159], Dina et al. used a focal loss function to train DL-based models to overcome the data imbalance issue for intrusion detection in the IoT. The authors optimized the model by enabling dynamically scaled gradient updatesand down-weighing easy instances. Also, the focal loss function caused the model to focus on the hard misclassified examples. The authors implemented this function in Feed-Forward Neural Networks and Convolutional Neural Networks. They used well-known datasets to analyze the correctness of the proposed intrusion-detection model. The authors received better results for the model based on Feed-Forward Neural Networks.

Thus, in [160], Altaf et al. proposed a Graph Neural Network-based (GNN) Intrusion-Detection System. In their method, the network graph processed multiple edges with multi-dimensional edge features. Thanks to this, the network graph can capture the complete exchange of information between any pair of nodes. In this structure, nodes mean the source and destination IP addresses. Multiple adjacent edges and the multi-dimensional edge feature matrix describe the device communication. The authors evaluated their model on four benchmark datasets for typical performance metrics. Also, they compared their method with other GNN models and achieved better effectiveness of their model than the other models.

In [161–163], Blaise et al. developed a botnet-detection method called BotFP. The proposed method considers two variants: the network traffic is defined according to the source IP address, and the frequency distribution signatures of the IP protocol attributes distinguish the host activity. The first variant groups similar traffic instances using clustering algorithms, while the second variant is used to classify new bots and uses the SVM and Multi-layer Perceptron algorithms. The research showed that the proposed method detects bots with high accuracy in both variants. Nevertheless, the method proposed by Blaise et al. filters out hosts with the number of packets below a certain threshold value, which means that bots must be active on the network and can help bypass hidden bots.

Ilango et al. in [164] focused on detecting Low-Rate Denial of Service (LR DoS) attacks. These are attacks where the attacker acts with extreme care and sends a burst of traffic that forces the data flow using the TCP protocol to enter the retransmission timeout state. These attacks are difficult to detect because they resemble real network traffic. The authors proposed an LR-DoS-attack-detection system based on the Feed-Forward Convolutional Neural Network (FFCNN). The system transforms the data into a two-dimensional grid for the CNN, which it then uses to extract the input features deeply. The system processes the output to judge whether the traffic is malicious or benign. The system was tested and compared with other IDS solutions using publicly available datasets.

Gupta et al. in [165] focused on MITM attacks in medical sensor networks. The authors proposed an attack-detection model based on the combined RF and Grid Search CV techniques. Gupta et al. showed that the proposed model could protect sensor data against cyber attacks. Thus, the proposed system can help protect the privacy and security of patients in real-time. Using a publicly available dataset for MITM attacks, the authors lost the system and obtained an average accuracy of 94.23%. Additionally, the authors tested the model for its performance and achieved a shorter classification time than other ML techniques.

In [166], Sohi et al. also focused on applying DL techniques to IDSs. This time, the authors focused primarily on zero-day attacks related to newly discovered vulnerabilities in various systems. Sohi et al. found that signature-based IDSs may not detect zero-day attacks. For this reason, they used Recursive Neural Networks (RNNs) to find complex attack patterns and generate similar signatures. The generated signatures were also assessed to improve the IDS detection rate. The performance evaluation of the proposed RNN showed that the accuracy of this system increased by 16.67% compared to other detection methods. In conclusion, the authors emphasized that RNNs can be successfully used in IDS systems due to the possibility of extracting features from existing attack patterns and generating new variants to anticipate new and unknown attacks.

It is also worth mentioning verifying the correctness of the operation and the effectiveness of IDSs' intrusion detection and that the types of attacks and the methods of carrying them out are constantly evolving. Hence, there is also a need to model attacks using AI methods to strengthen systems protecting [167] network users. The main problem during AI usage can be adversarial attacks. During this attack, the attacker tries to fool the AI model by applying perturbations to some data. Adversarial attacks can affect AI algorithms' training phase, resulting in false detection, favoring the attacker.

The works by Han et al. [168,169] proposed a new adversarial attack structure that assesses the resistance of IDSs based on ML algorithms. The created structure considered the automatic mutation of malicious network traffic while maintaining its functionality. Using this structure's Generative Adversarial Network (GAN), the authors generated opposing features. Then, using Particle Swarm Optimization (PSO), they searched for mutants of the mysterious movement. Also, they proposed a system for IDSs' verification.

Also, Jiang et al. in [170] considered an adversarial attack problem. Firstly, the authors proposed an algorithm for generating adversarial samples. Secondly, they offered an IDS for an IoT network called the Feature Grouping and Multi-model Fusion Detector. The authors tested their model using publicly available datasets for the IoT. Also, the authors obtained better performance of the proposed system than another method against adversarial attacks. Jiang et al. concluded that their approach allowed for resisting adversarial attacks with high precision.

Another problem related to users' security is malicious software, for example malware in intelligent devices. Usually, the attacker prepares malware to steal the data, but also, it can contain code that will harm the operating system [171]. Imtiaz et al. in [172] considered the malware-detection problem. The authors proposed the DeepAMD system for Android-based devices. The proposed method used Deep Artificial Neural Networks to identify and detect malware applications on the Android system. The test results confirmed the high precision of malicious software detection.

In turn, Rey et al. in [173] used FL for malware detection in IoT devices. The authors employed Multi-Layer Perceptron and auto-encoder-based models. The authors used a dataset that models malware-affected network traffic for tests. They compared the federated approach with a non-privacy-preserving setup and a local setup. They concluded that federated methods positively impact malware detection in IoT devices.

4.2. Discussion and Findings

This subsection summarizes the overviewed papers on TAI-based security for users against intrusion attacks. Tables 3 and 4 have an identical structure. In the column Refs., we assigned bibliographic references to the overviewed papers. The second column contains information about the AI techniques used during the research. In the third column, we summarize the characteristics of the proposed method. The fourth column contains information about the application of the proposed methods. The fifth column (Acc.) contains the average accuracy value obtained during the research. The last column (Prec.) contains the average precision value obtained during the research. The designation NM in the last two columns means that these values were Not Mentioned in the cited papers.

Table 3 summarizes the overviewed proposed AI techniques used in IDSs, including their application. We overviewed IDSs for each CPS and specific solutions like the IoT or vehicular networks. The most-popular AI techniques in the overviewed systems were the RF, GRU, and LSTM. The features of the overviewed methods depended on AI techniques and IDS solutions.

In Table 4, we summarize the overviewed proposed AI techniques for specific attacks and malware detection. We overviewed methods for each CPS and methods dedicated to the IoT. Also, there were methods dedicated to devices with the Android operating system. These studies focused on malware detection and dangerous attacks like zero-day, DoS, MITM, and adversarial attacks. The most-popular AI techniques in the overviewed methods were the SVM, DNN, and RNN.

Table 3. Summary of overviewed proposed AI techniques used in IDSs.

Refs.	AI Techniques	Characteristics	Application	Acc.	Prec.
[137]	DNN	• complies with security, privacy, centralization, trust, and integrity requirements in connected device environments • uses blockchain technology	IoT	98.97	97.71
[139]	SVM, NB	• converts low-quality data to high-quality data • uses the idea of feature embedding to classify the features of normal and abnormal data	CPSs	NM	NM
[140]	RNN, LSTM, GRU, RF	• effective monitoring of network traffic and alerting about possible attacks in real-time • uses KPCA to reduce dimensionality without losing data	CPS	98	96
[141]	LightGBM	• effectively detects threats related to remote network connections	CPS	99.2	98.78
[145]	DRNN	• the used dense clusters can communicate and perform interactions similar to the interactions between brain cells	IoT	99.14	99.13
[146]	Metric Learning	• tested using two specially prepared datasets • a feature classifier may use a subtractive angular margin loss in a cosine space	CPS	99.94	99
[147]	GA, PSO, IRF	• selects the best features, thus minimizing the number of dimensions, increasing the true positive rate and lowering the false positive rate	CPS	98.98	99.85
[148]	ML, BL	• ensures safe travel with intelligent vehicles and roads • protects the network and the devices against attacks, thus protecting human life	VANET	NM	NM
[150,151]	DL, LSTM, GRU	• detects attacks in vehicle networks • reduces training and response times and improves the accuracy of attack detection	VANET	99.7	97.26
[152]	FL, GRU, RF	• learning is asynchronous • each network node independently executes the training algorithm with its copy of the dataset • each node shares the weights of the local learned model with the aggregate instance	VANET	99.52	99.77
[153]	CNN	• it effectively distinguishes attack traffic from ordinary information flowing in a network of vehicles • enables detection of intrusions in real-time	VANET	NM	NM
[154,155]	KNN, RF, XGBoost	• high detection rate of less-popular cyber attacks	IoT	99.11	98.56
[158]	FL	• high accuracy • high detection rate of popular cyber attacks	IoT	98.62	98.9
[159]	FFCNN	• adaptation of focal loss function intrusion detection	IoT	93.26	95.24
[160]	GNN	• processing multiple edges with multi-dimensional edge features in the graph structure	IoT	99.45	98.89

Some overviewed studies did not adapt existing AI algorithms for intrusion or attack detection. The authors of these studies proposed their own solution, so they marked an AI method family (ML or DL) only. Each proposed method was verified according to the accuracy, precision, and detection rate (recall) metrics and the F1-score. The most-significant parameters were the accuracy and precision. Accuracy is a measure of the efficiency or effectiveness in the context of classification in Machine Learning. It measures how accurately a model classifies data. Precision is a widely utilized performance indicator in the fields of Machine Learning and statistics, particularly within the domain of binary classification. The metric evaluates the precision of a model's positive predictions by calculating the proportion of true positive predictions to the overall number of positive predictions, encompassing both true positives and false positives. The precision metric is employed to assess the efficacy of a model in accurately predicting the classification of an example as belonging to the positive class. We included the average values of the accuracy and precision mentioned by the authors in Tables 3 and 4. The authors used publicly available datasets to prove their methods. Only Cai et al. in [146] prepared their datasets.

Table 4. Summary of overviewed proposed AI techniques for specific attacks and malware detection.

Refs.	AI Techniques	Characteristics	Application	Acc.	Prec.
[156]	DL, SNDAE, SVM	• ability to detect zero-day attacks	IoT	99.65	99.99
[157]	DNN	• extracts the characteristics of network traffic • detects MITM and DoS attacks	IoT	99.57	98.45
[161–163]	SVM, Multilayer Perceptron	• defines network traffic based on the source IP address • identifies host activity using the frequency distribution signature of IP protocol attributes • detects botnets	CPS	NM	NM
[164]	FFCNN	• detects Low-Rate Denial of Service attacks	IoT	99	99.45
[165]	RF, Grid Search CV	• protects the privacy and security of patients in real-time • detection of MITM attacks in medical sensor networks	IoT	94.23	93.45
[166]	RNN	• detects zero-day attacks • used in IDSs due to the possibility of extracting functions from existing attack patterns and generating new variants of attacks to predict new and unknown attacks	CPS	NM	NM
[168,169]	GAN, PSO	• adversarial attack detection	CPS	NM	NM
[170]	LSTM, RNN	• protects against adversarial attacks	IoT	99.82	99.59
[172]	DNN	• identifies and detects malware applications on the Android system	Android system, IoT	NM	NM
[173]	FL	• identifies and detects malware applications in the IoT connected to WiFi, 5G, or B5G networks	IoT	99.87	99.98

The analyzed research showed a strong need to provide intelligent intrusion-and attack-detection methods. This need is also related to the ever-growing demand for communication via electronic links between users and devices. Users communicate with each other, communicate with devices, as well as their devices communicate with other devices. Users, therefore, require that information transmitted during such communication remains safe and intact by rogue computer network users. Determining which attack is the most-dangerous for users and devices operating on computer networks is impossible. All kinds of inappropriate user activity can devastate users and their data.

Hence, trust in AI algorithms is needed to assess network traffic in terms of possible intrusions. IDSs should meet data privacy and security requirements by detecting common and less-popular attacks for a specific solution. Users should feel safe knowing that a trustworthy system will respond quickly and block a bad connection. At this same time, TAI methods in IDSs successfully deal with technical problems related to data quality and efficiency and reduce learning and response times. The prediction of an attack or inappropriate user behavior should be prepared in real-time so that the system and its methods become more trustworthy.

Our insights and findings from the described studies are as follows. Firstly, almost all mentioned methods used publicly available datasets to train and verify the proposed systems. These datasets may not imitate real network conditions. The network conditions are constantly changing, and devices connect to and disconnect from the network, while these devices have different software and hardware configurations. Also, we can indicate many types of malware software that are still evolving. Thus, methods for Intrusion-Detection Systems should also be trained and verified using real network environments and devices. Thanks to this, the analysis and evaluations will show the proposed methods' effectiveness. The issue of preparing realistic datasets should be approached in several steps. First, it should be determined whether the set is to be prepared for one specific attack, for several, or for general network traffic, which will enable the detection of various irregularities in the network. Each dataset version will be equally valuable due to the possibility of testing IDSs. Then, we need to prepare a controlled test environment against which we can test attacks. Attacks should be conducted based on real scenarios, including various types of network traffic, devices, regular user traffic, and environmental attacks. This ensures that the test data generated will be both realistic and diverse. Operational scenarios should also consider different levels of attack complexity, different network protocols and attack styles, and different times of device connections and activity in the network, such as infinite attacks or data transformation attempts, depending on the assumptions made before preparing the test environment.

Secondly, the set of cyber attack types is extensive. Probably, no system will detect all possible attacks and intrusions. So, these detection methods focus mainly on the most-typical attacks on CPSs, and identifying the most-prevalent threats is a good approach. However, detecting the least-popular, niche attacks should also be considered to prevent more-complex attacks using many types of attacks. Some CPSs must be secured against one or two attack types, but most of them must be secured against a broad spectrum of attacks, mainly in the case of users' data. Indicating specific threats or attack vectors that need to be considered is difficult. Both methods of defending against cyber attacks and conducting cyber attacks are constantly evolving. This all depends on the specific environment. In the context of IoT security, in addition to the typical attacks mentioned earlier and attacks related to the poor security of IoT devices (due to a lack of software updates or the use of weak passwords), you should also pay attention to the following:

- Attacks on custom network configurations;
- Attacks on unique systems (for example, those using specialized software or niche technologies);
- Attacks on custom ports or services unique to a particular organization or industry;
- Attacks on protocols: old generation, new generation, or less popular.

These attack vectors primarily target the technical parameters of IoT environments and CPSs and can be very dangerous for users and the entire infrastructure. So, CPSs need Intrusion-Detection Systems and methods for specific attack detection.

At last, the accuracy and precision values obtained and included in Tables 3 and 4 are promising. They allowed us to conclude that the discussed methods meet TAI's safety, robustness, privacy and data governance assumptions. High accuracy and precision values (over 90%) suggest that these systems detect network intrusions and threats with high efficiency and precision. This means that they provide the appropriate level of network user safety. Moreover, if network users are secure, their data also are secure. So, the privacy and data governance dimension requirements can also be considered fulfilled. The robustness dimension is the system's ability to respond even during disturbances. Network traffic has various characteristics, so disturbances in it are a natural phenomenon that IDSs detect. This means that each mentioned system can adapt to changing network conditions and correctly detect intrusions and threats in network traffic.

5. Conclusions

This manuscript explored the use of Trustworthy Artificial Intelligence (TAI) methods to enhance users' physical and environmental security. It focused on traffic and pedestrian safety systems, detecting and predicting behavior, and Intrusion-Detection Systems for IoT systems. The study highlighted TAI's safety and robustness dimensions, considering users' security in physical and environmental safety. It also explored less-exploited TAI domains like Spiking Neural Networks, neuromorphic accelerators, and Dynamic Vision Sensors. As an indispensable element of the modern world, AI is still evolving, and new AI techniques are appearing. Trustworthiness should be a key feature of such solutions, which make people aware of how much they can trust them. We must constantly verify these methods to check their trustworthiness in many dimensions.

We noted the following insights and conclusions while analyzing the described methods and systems selected for this manuscript.

When it comes to safety regarding traffic, the analysis of the vehicle's environment is very important. As presented in numerous articles, vehicle detection is more straightforward due to the smaller variety of features of such objects. Systems in this area work very well, resulting in the existence of autonomous vehicles. In the case of pedestrian detection, the problem is more complex. Under all conditions, an ideal system to recognize them can improve their safety. Several proposals have been made for such systems, which achieve good results, but it is still worth creating an infallible system.

The challenge for neural networks is the perfect detection of objects, preserving high resilience to attacks. Further research should focus on improving current solutions to enhance security. Increasing city safety can also be realized by implementing a "smart city" [174,175]. We think the cooperation of multiple neural networks at different levels can greatly improve our security, which will also be an interesting research direction. Fusion, which involves the seamless integration of information from various sensors, such as cameras and LiDARs, shows promising results [176–179]. These sensors operate using different technologies, and analyzing environmental conditions for each of them poses a challenge. Moreover, this advanced technique allows for a substantial improvement in detection effectiveness, also in challenging environmental conditions, and it enhances precision and decision reliability in favorable conditions.

We identified a small gap worth fulfilling as we found only a few papers concerning explainable SNNs. Fulfilling this gap can go along with explainable AI methods on traditional ANNs. However, some SNNs, based on the ToM, are already explicitly designed, which is a step toward explainable SNNs.

Some of the mentioned papers point out that SNNs exhibit a natural ability to resist some adversarial attacks to some extent (probably due to their stochastic nature) under the condition that the parameters of a network are tuned. Hence, it is worth further exploring the parameter space of SNNs to develop a framework for easier parameter tuning against adversarial attacks.

It is also worth mentioning that we still do not have a stable and strongly neurobiologically inspired framework for SNNs' training or continuous learning. Hence, designing this kind of framework would be appreciated in the field.

In the case of environmental security, which is connected with users' activities in the network (for example, as IoT systems users), we observed the extreme need to provide intelligent intrusion- and attack-detection methods. Also, security is highly joined with the safety dimension of TAI. Users use a network during many activities, and many users and devices connect with the network. They can be potential victims of cyber attacks with different sources, courses, and effects. Also, technological development entails the development of cyber attack methods.

Nevertheless, users cannot always defend themselves against attacks on their own, and users need security and protection techniques that will protect their private data and can rely on it. Therefore, we believe that creating intelligent methods of detecting intrusions and attacks using TAI algorithms will undoubtedly increase users' environmental security.

Based on our insights, future research directions in Intrusion-Detection Systems should focus on methods of detecting all possible attacks, especially less-frequently used ones. Especially, the research should consider attacks on custom network configurations, unique systems (with specialized software or niche technologies), custom ports or services unique to a particular organization or industry, and on protocols (old generation, new generation, or less popular). Also, the important point in such research should be preparing datasets corresponding to the real working environment of the devices and testing IDSs in a real network. To prepare realistic datasets for IDSs, it is necessary to determine if the set is for specific attacks or general network traffic. In the next step, researchers should create a controlled test environment with real scenarios, including network traffic, devices, user traffic, and environmental attacks. Also, it is necessary to consider different levels of attack complexity, network protocols, attack styles, and device connections to ensure diverse and realistic data. The existing datasets for testing are well-known and widely used. Therefore, they may be unable to detect many security-threatening situations for network users on a real network, especially the less-frequently used or the newest attacking techniques. Finally, IDSs require continuous development. They should be continuously developed because the attacking methods still evolve, and IDSs should consider these changes to improve their operation.

Author Contributions: Conceptualization, S.S., F.D., Ł.K. and M.K.; methodology, S.S., F.D., Ł.K. and M.K.; validation, S.S., F.D., Ł.K. and M.K.; investigation, S.S., F.D., Ł.K. and M.K.; data curation, S.S., F.D., Ł.K. and M.K.; writing—original draft preparation, S.S., F.D., Ł.K. and M.K.; writing—review and editing, S.S., F.D., Ł.K. and M.K.; visualization, S.S., F.D., Ł.K. and M.K.; supervision, S.S., F.D., Ł.K. and M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Liu, J.; Tang, Y.; Zhao, H.; Wang, X.; Li, F.; Zhang, J. CPS Attack Detection under Limited Local Information in Cyber Security: An Ensemble Multi-node Multi-class Classification Approach. *ACM Trans. Sens. Netw.* **2023**. [CrossRef]
2. Thiebes, S.; Lins, S.; Sunyaev, A. Trustworthy artificial intelligence. *Electron. Mark.* **2021**, *31*, 447–464. [CrossRef]
3. Kaur, D.; Uslu, S.; Rittichier, K.J.; Durresi, A. Trustworthy Artificial Intelligence: A Review. *ACM Comput. Surv.* **2022**, *55*, 1–38. [CrossRef]
4. Hasan, M.K.; Habib, A.A.; Shukur, Z.; Ibrahim, F.; Islam, S.; Razzaque, M.A. Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations. *J. Netw. Comput. Appl.* **2023**, *209*, 103540. [CrossRef]
5. Mittal, A.; Dhalla, S.; Gupta, S.; Gupta, A. Automated Analysis of Blood Smear Images for Leukemia Detection: A Comprehensive Review. *ACM Comput. Surv.* **2022**, *54*, 1–37. [CrossRef]
6. Alzamer, H.; Abuhmed, T.; Hamad, K. A short review on the Machine Learning-guided oxygen uptake prediction for sport science applications. *Electronics* **2021**, *10*, 1956. [CrossRef]
7. Yan, C.; Ji, X.; Wang, K.; Jiang, Q.; Jin, Z.; Xu, W. A Survey on Voice Assistant Security: Attacks and Countermeasures. *ACM Comput. Surv.* **2022**, *55*, 1–36. [CrossRef]
8. Raut, R.; Jadhav, A.; Jaiswal, S.; Pathak, P. IoT-Assisted Smart Device for Blind People. In *Intelligent Systems for Rehabilitation Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2022; pp. 129–150.
9. Kubanek, M.; Bobulski, J. Device for Acoustic Support of Orientation in the Surroundings for Blind People. *Sensors* **2018**, *18*, 4309. [CrossRef]
10. Nait Aicha, A.; Englebienne, G.; Van Schooten, K.S.; Pijnappels, M.; Kröse, B. Deep Learning to Predict Falls in Older Adults Based on Daily-Life Trunk Accelerometry. *Sensors* **2018**, *18*, 1654. [CrossRef]
11. Platt, F. Physical Threats to the Information Infrastructure. In *Computer Security Handbook*; John Wiley & Sons: New York, NY, USA, 2012; pp. 1–22.
12. Szymoniak, S. Amelia—A new security protocol for protection against false links. *Comput. Commun.* **2021**, *179*, 73–81. [CrossRef]
13. Guembe, B.; Azeta, A.; Misra, S.; Ahuja, R. Trustworthy Machine Learning Approaches for Cyberattack Detection: A Review. In Proceedings of the International Conference on Computational Science and Its Applications, Malaga, Spain, 4–7 July 2022; pp. 265–278. [CrossRef]
14. Smuha, N.A. The EU Approach to Ethics Guidelines for Trustworthy Artificial Intelligence. *Comput. Law Rev. Int.* **2019**, *20*, 97–106. [CrossRef]
15. Hickman, E.; Petrin, M. Trustworthy AI and corporate governance: The EU's ethics guidelines for trustworthy artificial intelligence from a company law perspective. *Eur. Bus. Organ. Law Rev.* **2021**, *22*, 593–625. [CrossRef]
16. Liu, H.; Wang, Y.; Fan, W.; Liu, X.; Li, Y.; Jain, S.; Liu, Y.; Jain, A.K.; Tang, J. Trustworthy AI: A Computational Perspective. *ACM Trans. Intell. Syst. Technol.* **2022**, *14*, 1–59. [CrossRef]
17. Hasan, R.; Hasan, R. Pedestrian safety using the Internet of Things and sensors: Issues, challenges, and open problems. *Future Gener. Comput. Syst.* **2022**, *134*, 187–203. [CrossRef]
18. Bharadiya, J. Artificial Intelligence in Transportation Systems A Critical Review. *Am. J. Comput. Eng.* **2023**, *6*, 34–45. [CrossRef]
19. Bhattacharya, S.; Jha, H.; Nanda, R.P. Application of IoT and Artificial Intelligence in Road Safety. In Proceedings of the 2022 Interdisciplinary Research in Technology and Management (IRTM), Kolkata, India, 26–28 February 2021; pp. 1–6.
20. Olugbade, S.; Ojo, S.; Imoize, A.L.; Isabona, J.; Alaba, M.O. A Review of Artificial Intelligence and Machine Learning for Incident Detectors in Road Transport Systems. *Math. Comput. Appl.* **2022**, *27*, 77. [CrossRef]
21. Mchergui, A.; Moulahi, T.; Zeadally, S. Survey on artificial intelligence (AI) techniques for vehicular ad-hoc networks (VANETs). *Veh. Commun.* **2022**, *34*, 100403. [CrossRef]

22. Abdullahi, M.; Baashar, Y.; Alhussian, H.; Alwadain, A.; Aziz, N.; Capretz, L.F.; Abdulkadir, S.J. Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. *Electronics* **2022**, *11*, 198. [CrossRef]
23. de Souza, C.A.; Westphall, C.B.; Machado, R.B.; Loffi, L.; Westphall, C.M.; Geronimo, G.A. Intrusion detection and prevention in fog based IoT environments: A systematic literature review. *Comput. Netw.* **2022**, *214*, 109154. [CrossRef]
24. Ahmad, R.; Alsmadi, I.; Alhamdani, W.; Tawalbeh, L. Zero-day attack detection: A systematic literature review. *Artif. Intell. Rev.* **2023**, *56*, 10733–10811. [CrossRef]
25. Victor, P.; Lashkari, A.H.; Lu, R.; Sasi, T.; Xiong, P.; Iqbal, S. IoT malware: An attribute-based taxonomy, detection mechanisms and challenges. *Peer-to-Peer Netw. Appl.* **2023**, *16*, 1380–1431. [CrossRef] [PubMed]
26. Barnawi, A.; Gaba, S.; Alphy, A.; Jabbari, A.; Budhiraja, I.; Kumar, V.; Kumar, N. A systematic analysis of deep learning methods and potential attacks in internet-of-things surfaces. *Neural Comput. Appl.* **2023**, *35*, 18293–18308. [CrossRef]
27. Ben-Bright, B.; Yongzhao, Z.; Benjamin, G.; Keddy, W.D.; Banaseka, K.F. A Review of Deep Machine Learning. *Int. J. Eng. Res. Afr.* **2016**, *24*, 124–136. [CrossRef]
28. Hammadi, O.I.; Abas, A.D.; Ayed, K.H. Face recognition using deep learning methods a review. *Int. J. Eng. Technol.* **2018**, *7*, 6181–6188. [CrossRef]
29. Contreras-Valenzuela, M.R.; Seuret-Jiménez, D.; Hdz-Jasso, A.M.; León Hernández, V.A.; Abundes-Recilla, A.N.; Trutié-Carrero, E. Design of a Fuzzy Logic Evaluation to Determine the Ergonomic Risk Level of Manual Material Handling Tasks. *Int. J. Environ. Res. Public Health* **2022**, *19*, 6511. [CrossRef] [PubMed]
30. Serrano, W. Neural networks in big data and Web search. *Data* **2018**, *4*, 7. [CrossRef]
31. Senthilkumar, M.; Chowdhary, C.L. An AI-Based Chatbot Using Deep Learning. In *Intelligent Systems*; Apple Academic Press: Ontario, CA, USA, 2019; pp. 231–242. [CrossRef]
32. Abel, S.; Criado, J.C.; Spannowsky, M. Completely quantum neural networks. *Phys. Rev. A* **2022**, *106*, 022601. [CrossRef]
33. Agavanakis, K.N.; Karpetas, G.E.; Taylor, M.; Pappa, E.; Michail, C.M.; Filos, J.; Trachana, V.; Kontopoulou, L. Practical Machine Learning based on cloud computing resources. In Proceedings of the AIP Conference Proceedings, Beirut, Lebanon, 10–12 April 2019; Volume 2123, p. 020096. [CrossRef]
34. Jeong, D.S. Tutorial: Neuromorphic spiking neural networks for temporal learning. *J. Appl. Phys.* **2018**, *124*, 152002. [CrossRef]
35. Schuman, C.D.; Potok, T.E.; Patton, R.M.; Birdwell, J.D.; Dean, M.E.; Rose, G.S.; Plank, J.S. A Survey of Neuromorphic Computing and Neural Networks in Hardware. *arXiv* **2017**, arXiv:1705.06963.
36. Davies, M.; Wild, A.; Orchard, G.; Sandamirskaya, Y.; Guerra, G.A.F.; Joshi, P.; Plank, P.; Risbud, S.R. Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook. *Proc. IEEE* **2021**, *109*, 911–934. [CrossRef]
37. Orchard, G.; Frady, E.P.; Rubin, D.B.D.; Sanborn, S.; Shrestha, S.B.; Sommer, F.T.; Davies, M. Efficient neuromorphic signal processing with loihi 2. In Proceedings of the 2021 IEEE Workshop on Signal Processing Systems (SiPS), Coimbra, Portugal, 20–22 October 2021; pp. 254–259.
38. Cachi, P.G.; Ventura, S.; Cios, K.J. MT-SNN: Spiking Neural Network that Enables Single-Tasking of Multiple Tasks. *arXiv* **2022**, arXiv:2208.01522.
39. Bekolay, T.; Bergstra, J.; Hunsberger, E.; Dewolf, T.; Stewart, T.; Rasmussen, D.; Choo, X.; Voelker, A.; Eliasmith, C. Nengo: A Python tool for building large-scale functional brain models. *Front. Neuroinform.* **2014**, *7*, 48. [CrossRef] [PubMed]
40. Aimone, J.B. Neural algorithms and computing beyond Moore's law. *Commun. ACM* **2019**, *62*, 110. [CrossRef]
41. Eshraghian, J.K.; Ward, M.; Neftci, E.; Wang, X.; Lenz, G.; Dwivedi, G.; Bennamoun, M.; Jeong, D.S.; Lu, W.D. Training Spiking Neural Networks Using Lessons From Deep Learning. *Proc. IEEE* **2022**, *111*, 1016–1054. [CrossRef]
42. Mozafari, M.; Ganjtabesh, M.; Nowzari-Dalini, A.; Masquelier, T. SpykeTorch: Efficient Simulation of Convolutional Spiking Neural Networks With at Most One Spike per Neuron. *Front. Neurosci.* **2019**, *13*, 625. [CrossRef] [PubMed]
43. Lytton, W.W.; Seidenstein, A.H.; Dura-Bernal, S.; McDougal, R.A.; Schürmann, F.; Hines, M.L. Simulation Neurotechnologies for Advancing Brain Research: Parallelizing Large Networks in NEURON. *Neural Comput.* **2016**, *28*, 2063–2090. [CrossRef]
44. Tiddia, G.; Golosio, B.; Albers, J.; Senk, J.; Simula, F.; Pronold, J.; Fanti, V.; Pastorelli, E.; Paolucci, P.S.; van Albada, S.J. Fast Simulation of a Multi-Area Spiking Network Model of Macaque Cortex on an MPI-GPU Cluster. *Front. Neuroinform.* **2022**, *16*, 883333. [CrossRef]
45. Stimberg, M.; Brette, R.; Goodman, D.F. Brian 2, an intuitive and efficient neural simulator. *eLife* **2019**, *8*, e47314. [CrossRef]
46. Niedermeier, L.; Chen, K.; Xing, J.; Das, A.; Kopsick, J.; Scott, E.; Sutton, N.; Weber, K.; Dutt, N.; Krichmar, J.L. CARLsim 6: An Open Source Library for Large-Scale, Biologically Detailed Spiking Neural Network Simulation. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–10. [CrossRef]
47. Rothganger, F.; Warrender, C.; Trumbo, D.; Aimone, J. N2A: A computational tool for modeling from neurons to algorithms. *Front. Neural Circuits* **2014**, *8*, 1. [CrossRef]
48. Gerstner, W.; Kistler, W.M.; Naud, R. *Neuronal Dynamics*; Cambridge University Press: Cambridge, UK, 2014.
49. Zenke, F.; Neftci, E.O. Brain-inspired learning on neuromorphic substrates. *Proc. IEEE* **2021**, *109*, 935–950. [CrossRef]
50. Kempter, R.; Gerstner, W.; van Hemmen, J.L. Hebbian learning and spiking neurons. *Phys. Rev. E* **1999**, *59*, 4498–4514. [CrossRef]
51. The 6 Levels of Vehicle Autonomy Explained. 2022. Available online: https://www.synopsys.com/automotive/autonomous-driving-levels.html (accessed on 12 June 2023).
52. Silva, P.B.; Andrade, M.; Ferreira, S. Machine Learning applied to road safety modeling: A systematic literature review. *J. Traffic Transp. Eng.* **2020**, *7*, 775–790. [CrossRef]

53. Eskandari Torbaghan, M.; Sasidharan, M.; Reardon, L.; Muchanga-Hvelplund, L.C. Understanding the potential of emerging digital technologies for improving road safety. *Accid. Anal. Prev.* **2022**, *166*, 106543. [CrossRef] [PubMed]
54. Chen, L. Road vehicle recognition algorithm in safety assistant driving based on artificial intelligence. *Soft Comput.* **2021**, *27*, 1153–1162. [CrossRef]
55. Fu, Y.; Li, C.; Yu, F.R.; Luan, T.H.; Zhang, Y. A Survey of Driving Safety With Sensing, Vehicular Communications, and Artificial Intelligence-Based Collision Avoidance. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6142–6163. [CrossRef]
56. Benterki, A.; Boukhnifer, M.; Judalet, V.; Maaoui, C. Artificial Intelligence for Vehicle Behavior Anticipation: Hybrid Approach Based on Maneuver Classification and Trajectory Prediction. *IEEE Access* **2020**, *8*, 56992–57002. [CrossRef]
57. Ahmad, S.e.a. Accident Risk Prediction and Avoidance in Intelligent Semi-autonomous Vehicles Based on Road Safety Data and Driver Biological Behaviours. *J. Intell. Fuzzy Syst.* **2020**, *38*, 4591–4601. [CrossRef]
58. Kuşkapan, E.; Sahraei, M.A.; Çodur, M.K.; Çodur, M.Y. Pedestrian safety at signalized intersections: Spatial and machine learning approaches. *J. Transp. Health* **2022**, *24*, 101322. [CrossRef]
59. Police Report 2021. 2021. Available online: https://statystyka.policja.pl/download/20/381967/Wypadkidrogowe2021.pdf (accessed on 1 June 2023).
60. Tian, D.; Han, Y.; Wang, B.; Guan, T.; Wei, W. A Review of Intelligent Driving Pedestrian Detection Based on Deep Learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 5410049. [CrossRef]
61. Szarvas, M.; Yoshizawa, A.; Yamamoto, M.; Ogata, J. Pedestrian detection with convolutional neural networks. In Proceedings of the IEEE Proceedings. Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005; pp. 224–229. [CrossRef]
62. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014. [CrossRef]
63. R-CNN, Fast R-CNN, Faster R-CNN, YOLO—Object Detection Algorithms. Available online: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e (accessed on 20 September 2023).
64. Dong, P.; Wang, W. Better region proposals for pedestrian detection with R-CNN. In Proceedings of the 2016 Visual Communications and Image Processing (VCIP), Chengdu, China, 27–30 November 2016; pp. 1–4. [CrossRef]
65. Dollár, P.; Appel, R.; Belongie, S.; Perona, P. Fast Feature Pyramids for Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1532–1545. [CrossRef]
66. Wang, K.; Zhou, W. Pedestrian and cyclist detection based on deep neural network fast R-CNN. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419829651. [CrossRef]
67. Zhao, Z.Q.; Bian, H.; Hu, D.; Cheng, W.; Glotin, H. Pedestrian Detection Based on Fast R-CNN and Batch Normalization. In Proceedings of the Intelligent Computing Theories and Application, Liverpool, UK, 7–10 August 2017; Huang, D.S., Bevilacqua, V., Premaratne, P., Gupta, P., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 735–746.
68. Li, J.; Liang, X.; Shen, S.; Xu, T.; Feng, J.; Yan, S. Scale-Aware Fast R-CNN for Pedestrian Detection. *IEEE Trans. Multimed.* **2018**, *20*, 985–996. [CrossRef]
69. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:1506.01497.
70. Zhang, L.; Lin, L.; Liang, X.; He, K. Is Faster R-CNN Doing Well for Pedestrian Detection? In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 443–457.
71. Zhang, H.; Du, Y.; Ning, S.; Zhang, Y.; Yang, S.; Du, C. Pedestrian Detection Method Based on Faster R-CNN. In Proceedings of the 2017 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, China, 15–18 December 2017; pp. 427–430. [CrossRef]
72. Hung, G.L.; Sahimi, M.S.B.; Samma, H.; Almohamad, T.A.; Lahasan, B. *Faster R-CNN Deep Learning Model for Pedestrian Detection from Drone Images*; Springer: Cham, Switzerland, 2020; Volume 1, pp. 1–9.
73. Zhai, S.; Dong, S.; Shang, D.; Wang, S. An Improved Faster R-CNN Pedestrian Detection Algorithm Based on Feature Fusion and Context Analysis. *IEEE Access* **2020**, *8*, 138117–138128. [CrossRef]
74. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
75. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
76. Sweta Panigrahi, U.S.N.R. InceptionDepth-wiseYOLOv2: Improved implementation of YOLO framework for pedestrian detection. *Int. J. Multimed. Inf. Retr.* **2022**, *11*, 409–430. [CrossRef]
77. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
78. Li, A.; Gao, X.; Qu, C. Pedestrian Detection Based on Improved YOLOv3 Algorithm. In *Intelligent Life System Modelling, Image Processing and Analysis, Proceedings of the 7th International Conference on Life System Modeling and Simulation, LSMS 2021 and 7th International Conference on Intelligent Computing for Sustainable Energy and Environment, ICSEE 2021, Hangzhou, China, 30 October–1 November 2021*; Springer: Singapore, 2021. [CrossRef]
79. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.

80. Wen, B.; Wu, M. Study on Pedestrian Detection Based on an Improved YOLOv4 Algorithm. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 1198–1202. [CrossRef]
81. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017. Available online: http://xxx.lanl.gov/abs/1704.04861 (accessed on 20 September 2023).
82. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37. [CrossRef]
83. Murthy, C.B.; Hashmi, M.F.; Keskar, A.G Optimized MobileNet + SSD: A real-time pedestrian detection on a low-end edge device. *Int. J. Multimed. Inf. Retr.* **2021**, *10*, 171–184. [CrossRef]
84. El-Sayed, S.A.; Spyrou, T.; Pavlidis, A.; Afacan, E.; Camuñas-Mesa, L.A.; Linares-Barranco, B.; Stratigopoulos, H.G. Spiking Neuron Hardware-Level Fault Modeling. In Proceedings of the 2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS), Napoli, Italy, 13–15 July 2020; pp. 1–4. [CrossRef]
85. Spyrou, T.; El-Sayed, S.A.; Afacan, E.; Camunas-Mesa, L.A.; Linares-Barranco, B.; Stratigopoulos, H.G. Neuron Fault Tolerance in Spiking Neural Networks. In Proceedings of the 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), Virtual, 1–5 February 2021; pp. 743–748. [CrossRef]
86. Orchard, G.; Jayawant, A.; Cohen, G.K.; Thakor, N. Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades. *Front. Neurosci.* **2015**, *9*, 437. [CrossRef]
87. Amir, A.; Taba, B.; Berg, D.; Melano, T.; McKinstry, J.; Di Nolfo, C.; Nayak, T.; Andreopoulos, A.; Garreau, G.; Mendoza, M.; et al. A Low Power, Fully Event-Based Gesture Recognition System. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7388–7397. [CrossRef]
88. Spyrou, T.; El-Sayed, S.A.; Afacan, E.; Camuñas-Mesa, L.A.; Linares-Barranco, B.; Stratigopoulos, H.G. Reliability Analysis of a Spiking Neural Network Hardware Accelerator. In Proceedings of the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), Valencia, Spain, 25–27 March 2022; pp. 370–375. [CrossRef]
89. Han, W.S.; Han, I.S. All Weather Human Detection Using Neuromorphic Visual Processing. In *Intelligent Systems for Science and Information: Extended and Selected Results from the Science and Information Conference 2013*; Chen, L., Kapoor, S., Bhatia, R., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 25–44. [CrossRef]
90. Han, W.S.; Han, I.S. Neuromorphic visual object detection for enhanced driving safety. In Proceedings of the 2015 Science and Information Conference (SAI), London, UK, 28–30 July 2015; pp. 721–726. [CrossRef]
91. Prez, H.; Rudomin, I.; Tabarez-Paz, I. *Support Vector Machine and Spiking Neural Networks for Data Driven Prediction of Crowd Character Movement*; MIT Press: Cambridge, MA, USA, 2016; pp. 638–645. [CrossRef]
92. Ramanishka, V.; Chen, Y.T.; Misu, T.; Saenko, K. Toward Driving Scene Understanding: A Dataset for Learning Driver Behavior and Causal Reasoning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
93. Gaurav, R. Driving Scene Understanding using Spiking Neural Networks. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2022.
94. Zhao, Z.; Lu, E.; Zhao, F.; Zeng, Y.; Zhao, Y. A Brain-Inspired Theory of Mind Spiking Neural Network for Reducing Safety Risks of Other Agents. *Front. Neurosci.* **2022**, *16*, 753900. [CrossRef]
95. Farries, M.A.; Fairhall, A.L. Reinforcement Learning With Modulated Spike Timing–Dependent Synaptic Plasticity. *J. Neurophysiol.* **2007**, *98*, 3648–3665. [CrossRef] [PubMed]
96. Frémaux, N.; Gerstner, W. Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules. *Front. Neural Circuits* **2016**, *9*, 85. [CrossRef] [PubMed]
97. Nguyen, E. Temporal Spike Attribution: A Local Feature-Based Explanation for Temporally Coded Spiking Neural Networks. Master's Thesis, University of Twente, Enschede, The Netherlands, 2021.
98. Kim, Y.; Panda, P. Visual explanations from spiking neural networks using inter-spike intervals. *Sci. Rep.* **2021**, *11*, 19037. [CrossRef]
99. Seras, A.M.; Ser, J.D.; Lobo, J.L.; Garcia-Bringas, P.; Kasabov, N. A Novel Explainable Out-of-Distribution Detection Approach for Spiking Neural Networks. *arXiv* **2022**, arXiv:2210.00894.
100. Gallego, G.; Delbrück, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-Based Vision: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 154–180. [CrossRef] [PubMed]
101. Zhang, X.; Xiao, G.; Gui, S.; Ren, Q. Research on Brain-inspired Vision Based on Dynamic Vision Sensor Cameras. In Proceedings of the 2020 International Conference on Aviation Safety and Information Technology, Weihai, China, 14–16 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 725–733. [CrossRef]
102. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
103. Yao, M.; Gao, H.; Zhao, G.; Wang, D.; Lin, Y.; Yang, Z.; Li, G. Temporal-Wise Attention Spiking Neural Networks for Event Streams Classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 10221–10230. Available online: http://xxx.lanl.gov/abs/2107.11711 (accessed on 19 September 2023).

104. Li, H.; Liu, H.; Ji, X.; Li, G.; Shi, L. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Front. Neurosci.* **2017**, *11*, 309. [CrossRef] [PubMed]
105. Cramer, B.; Stradmann, Y.; Schemmel, J.; Zenke, F. The Heidelberg Spiking Data Sets for the Systematic Evaluation of Spiking Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 2744–2757. [CrossRef]
106. Liu, Q.; Xing, D.; Tang, H.; Ma, D.; Pan, G. Event-based Action Recognition Using Motion Information and Spiking Neural Networks. *SSRN* **2021**, *2*, 1743–1749. [CrossRef]
107. Miao, S.; Chen, G.; Ning, X.; Zi, Y.; Ren, K.; Bing, Z.; Knoll, A. Neuromorphic Vision Datasets for Pedestrian Detection, Action Recognition, and Fall Detection. *Front. Neurorobot.* **2019**, *13*, 38. [CrossRef]
108. Salah, M.; Chehadah, M.; Humais, M.; Wahbah, M.; Ayyad, A.; Azzam, R.; Seneviratne, L.; Zweiri, Y. A Neuromorphic Vision-Based Measurement for Robust Relative Localization in Future Space Exploration Missions. *arXiv* **2022**, arXiv:2206.11541.
109. Marchisio, A.; Pira, G.; Martina, M.; Masera, G.; Shafique, M. DVS-Attacks: Adversarial Attacks on Dynamic Vision Sensors for Spiking Neural Networks. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–9. [CrossRef]
110. Machado, G.R.; Silva, E.; Goldschmidt, R.R. Adversarial Machine Learning in Image Classification: A Survey Toward the Defender's Perspective. *ACM Comput. Surv.* **2021**, *55*, 8:1–8:38. [CrossRef]
111. Dapello, J.; Marques, T.; Schrimpf, M.; Geiger, F.; Cox, D.D.; DiCarlo, J.J. Simulating a Primary Visual Cortex at the Front of CNNs Improves Robustness to Image Perturbations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 13073–13087. [CrossRef]
112. Branytskyi, V.; Golovianko, M.; Malyk, D.; Terziyan, V. Generative adversarial networks with bio-inspired primary visual cortex for Industry 4.0. *Procedia Comput. Sci.* **2022**, *200*, 418–427. [CrossRef]
113. Shi, B.; Song, Y.; Joshi, N.; Darrell, T.; Wang, X. Visual Attention Emerges from Recurrent Sparse Reconstruction. *arXiv* **2022**, arXiv:2204.10962.
114. Mao, X.; Qi, G.; Chen, Y.; Li, X.; Duan, R.; Ye, S.; He, Y.; Xue, H. Towards Robust Vision Transformer. *arXiv* **2022**, arXiv:2105.07926.
115. Krithivasan, S.; Sen, S.; Rathi, N.; Roy, K.; Raghunathan, A. Efficiency Attacks on Spiking Neural Networks. In Proceedings of the 59th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 10–14 July 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 373–378. [CrossRef]
116. El-Allami, R.; Marchisio, A.; Shafique, M.; Alouani, I. Securing Deep Spiking Neural Networks against Adversarial Attacks through Inherent Structural Parameters. In Proceedings of the 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 1–5 February 2021; pp. 774–779. [CrossRef]
117. Nomura, O.; Sakemi, Y.; Hosomi, T.; Morie, T. Robustness of Spiking Neural Networks Based on Time-to-First-Spike Encoding Against Adversarial Attacks. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 3640–3644. [CrossRef]
118. Kim, Y.; Park, H.; Moitra, A.; Bhattacharjee, A.; Venkatesha, Y.; Panda, P. Rate Coding Or Direct Coding: Which One Is Better For Accurate, Robust, And Energy-Efficient Spiking Neural Networks? In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 71–75. [CrossRef]
119. Yang, W.; Liu, L.; Liu, Y.; Fan, L.; Lu, W. Secure and efficient multi-dimensional range query algorithm over TMWSNs. *Ad Hoc Netw.* **2022**, *130*, 102820. [CrossRef]
120. Temene, N.; Sergiou, C.; Georgiou, C.; Vassiliou, V. A Survey on Mobility in Wireless Sensor Networks. *Ad Hoc Netw.* **2022**, *125*, 102726. [CrossRef]
121. Szymoniak, S.; Siedlecka-Lamch, O. Securing Meetings in D2D IoT Systems. In Proceedings of the Ethicomp, 20th International Conference on the Ethical and Social issues in Information and Communication Technologies, Turku, Finland, 26–28 July 2022; pp. 30–41.
122. Sarker, I.H.; Furhad, M.H.; Nowrozy, R. Ai-driven cybersecurity: An overview, security intelligence modeling and research directions. *SN Comput. Sci.* **2021**, *2*, 1–18. [CrossRef]
123. Kubanek, M.; Bobulski, J.; Karbowiak, L.u. Intelligent Identity Authentication, Using Face and Behavior Analysis. In Proceedings of the Ethicomp, 20th International Conference on the Ethical and Social issues in Information and Communication Technologies, Turku, Finland, 26–28 July 2022; pp. 42–51.
124. Szymoniak, S.; Kesar, S. Key Agreement and Authentication Protocols in the Internet of Things: A Survey. *Appl. Sci.* **2023**, *13*, 404. [CrossRef]
125. Attkan, A.; Ranga, V. Cyber-physical security for IoT networks: A comprehensive review on traditional, blockchain and artificial intelligence based key-security. *Complex Intell. Syst.* **2022**, *8*, 3559–3591. [CrossRef]
126. Amma, N.; Selvakumar, S. Optimization of vector convolutional deep neural network using binary real cumulative incarnation for detection of distributed denial of service attacks. *Neural Comput. Appl.* **2022**, *34*, 2869–2882. [CrossRef] [PubMed]
127. Sivasankari, N.; Kamalakkannan, S. Detection and prevention of man-in-the-middle attack in iot network using regression modeling. *Adv. Eng. Softw.* **2022**, *169*, 103126. [CrossRef]
128. Liu, J.; Liu, L.; Liu, Z.; Lai, Y.; Qin, H.; Luo, S. WSN node access authentication protocol based on trusted computing. *Simul. Model. Pract. Theory* **2022**, *117*, 102522. [CrossRef]
129. Vinoth, R.; Deborah, L.J.; Vijayakumar, P.; Kumar, N. Secure Multifactor Authenticated Key Agreement Scheme for Industrial IoT. *IEEE Internet Things J.* **2021**, *8*, 3801–3811. [CrossRef]

130. Catillo, M.; Pecchia, A.; Villano, U. Botnet Detection in the Internet of Things through All-in-One Deep Autoencoding. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; Association for Computing Machinery: New York, NY, USA, 2022. [CrossRef]
131. Rao, P.M.; Deebak, B. Security and privacy issues in smart cities/industries: Technologies, applications, and challenges. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *14*, 10517–10553. [CrossRef]
132. He, Z.; Rezaei, A.; Homayoun, H.; Sayadi, H. Deep Neural Network and Transfer Learning for Accurate Hardware-Based Zero-Day Malware Detection. In Proceedings of the Great Lakes Symposium on VLSI 2022, Irvine, CA, USA, 6–8 June 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 27–32. [CrossRef]
133. Amanoul, S.V.; Abdulazeez, A.M.; Zeebare, D.Q.; Ahmed, F.Y. Intrusion-Detection Systems Based on Machine Learning Algorithms. In Proceedings of the 2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS), Shah Alam, Malaysia, 26 June 2021; pp. 282–287.
134. Kocher, G.; Kumar, G. Machine Learning and deep learning methods for intrusion detection systems: Recent developments and challenges. *Soft Comput.* **2021**, *25*, 9731–9763. [CrossRef]
135. Ahmad, R.; Wazirali, R.; Abu-Ain, T. Machine Learning for Wireless Sensor Networks Security: An Overview of Challenges and Issues. *Sensors* **2022**, *22*, 4730. [CrossRef]
136. Apruzzese, G.; Laskov, P.; Montes de Oca, E.; Mallouli, W.; Brdalo Rapa, L.; Grammatopoulos, A.V.; Di Franco, F. The role of Machine Learning in cybersecurity. *Digit. Threat. Res. Pract.* **2023**, *4*, 1–38. [CrossRef]
137. Kumar, R.; Tripathi, R. DBTP2SF: A deep blockchain-based trustworthy privacy-preserving secured framework in industrial internet of things systems. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4222. [CrossRef]
138. Kumar, N.; Makkar, A. *Machine Learning in Cognitive IoT*; CRC Press: Boca Raton, FL, USA, 2020.
139. Gu, J.; Lu, S. An effective intrusion detection approach using SVM with naïve Bayes feature embedding. *Comput. Secur.* **2021**, *103*, 102158. [CrossRef]
140. Ravi, V.; Chaganti, R.; Alazab, M. Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network Intrusion-Detection System. *Comput. Electr. Eng.* **2022**, *102*, 108156. [CrossRef]
141. Wang, B.X.; Chen, J.L.; Yu, C.L. An AI-Powered Network Threat Detection System. *IEEE Access* **2022**, *10*, 54029–54037. [CrossRef]
142. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146–3154.
143. Leevy, J.L.; Hancock, J.; Zuech, R.; Khoshgoftaar, T.M. Detecting cybersecurity attacks using different network features with lightgbm and xgboost learners. In Proceedings of the 2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI), Atlanta, GA, USA, 21–38 October 2020; pp. 190–197.
144. Leevy, J.L.; Hancock, J.; Zuech, R.; Khoshgoftaar, T.M. Detecting cybersecurity attacks across different network features and learners. *J. Big Data* **2021**, *8*, 1–29. [CrossRef]
145. Latif, S.; Huma, Z.e.; Jamal, S.S.; Ahmed, F.; Ahmad, J.; Zahid, A.; Dashtipour, K.; Aftab, M.U.; Ahmad, M.; Abbasi, Q.H. Intrusion Detection Framework for the Internet of Things Using a Dense Random Neural Network. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6435–6444. [CrossRef]
146. Cai, S.; Han, D.; Yin, X.; Li, D.; Chang, C.C. A Hybrid parallel deep learning model for efficient intrusion detection based on metric learning. *Connect. Sci.* **2022**, *34*, 551–577. [CrossRef]
147. Balyan, A.K.; Ahuja, S.; Lilhore, U.K.; Sharma, S.K.; Manoharan, P.; Algarni, A.D.; Elmannai, H.; Raahemifar, K. A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method. *Sensors* **2022**, *22*, 5986. [CrossRef]
148. Alsarhan, A.; Al-Ghuwairi, A.R.; Almalkawi, I.T.; Alauthman, M.; Al-Dubai, A. Machine Learning-driven optimization for intrusion detection in smart vehicular networks. *Wirel. Pers. Commun.* **2021**, *117*, 3129–3152. [CrossRef]
149. Du, Y.W.; Zhong, J.J. Generalized combination rule for evidential reasoning approach and Dempster–Shafer theory of evidence. *Inf. Sci.* **2021**, *547*, 1201–1232. [CrossRef]
150. Ullah, S.; Khan, M.A.; Ahmad, J.; Jamal, S.S.; e Huma, Z.; Hassan, M.T.; Pitropakis, N.; Buchanan, W.J. HDL-IDS: A hybrid deep learning architecture for intrusion detection in the Internet of Vehicles. *Sensors* **2022**, *22*, 1340. [CrossRef] [PubMed]
151. Ullah, S.; Ahmad, J.; Khan, M.A.; Alkhammash, E.H.; Hadjouni, M.; Ghadi, Y.Y.; Saeed, F.; Pitropakis, N. A New Intrusion-Detection System for the Internet of Things via Deep Convolutional Neural Network and Feature Engineering. *Sensors* **2022**, *22*, 3607. [CrossRef] [PubMed]
152. Driss, M.; Almomani, I.; Ahmad, J. A federated learning framework for cyber attack detection in vehicular sensor networks. *Complex Intell. Syst.* **2022**, *8*, 4221–4235. [CrossRef]
153. Hu, R.; Wu, Z.; Xu, Y.; Lai, T. Vehicular-Network-Intrusion Detection Based on a Mosaic-Coded Convolutional Neural Network. *Mathematics* **2022**, *10*, 2030. [CrossRef]
154. Roy, S.; Li, J.; Choi, B.J.; Bai, Y. A lightweight supervised intrusion detection mechanism for IoT networks. *Future Gener. Comput. Syst.* **2022**, *127*, 276–285. [CrossRef]
155. Roy, S.; Li, J.; Bai, Y. A Two-layer Fog-Cloud Intrusion Detection Model for IoT Networks. *Internet Things* **2022**, *19*, 100557. [CrossRef]
156. ul Haq Qazi, E.; Imran, M.; Haider, N.; Shoaib, M.; Razzak, I. An intelligent and efficient network Intrusion-Detection System using deep learning. *Comput. Electr. Eng.* **2022**, *99*, 107764. [CrossRef]

157. Nguyen, X.H.; Nguyen, X.D.; Huynh, H.H.; Le, K.H. Realguard: A Lightweight Network Intrusion-Detection System for IoT Gateways. *Sensors* **2022**, *22*, 432. [CrossRef]
158. Yang, R.; He, H.; Xu, Y.; Xin, B.; Wang, Y.; Qu, Y.; Zhang, W. Efficient intrusion detection toward IoT networks using cloud–edge collaboration. *Comput. Netw.* **2023**, *228*, 109724. [CrossRef]
159. Dina, A.S.; Siddique, A.; Manivannan, D. A deep learning approach for intrusion detection in Internet of Things using focal loss function. *Internet Things* **2023**, *22*, 100699. [CrossRef]
160. Altaf, T.; Wang, X.; Ni, W.; Yu, G.; Liu, R.P.; Braun, R. A new concatenated Multigraph Neural Network for IoT intrusion detection. *Internet Things* **2023**, *22*, 100818. [CrossRef]
161. Blaise, A.; Bouet, M.; Conan, V.; Secci, S. Botnet fingerprinting: A frequency distributions scheme for lightweight bot detection. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1701–1714. [CrossRef]
162. Blaise, A.; Bouet, M.; Conan, V.; Secci, S. Botfp: Fingerprints clustering for bot detection. In Proceedings of the NOMS 2020–2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–7.
163. Blaise, A.; Mihailescu, E.; Vidalenc, B.; Aufrechter, L.; Mihai, D.; Carabas, M. Learning Model Generalisation for Bot Detection. In Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference, Barcelona, Spain, 15–16 June 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 57–63. [CrossRef]
164. Ilango, H.S.; Ma, M.; Su, R. A FeedForward–Convolutional Neural Network to Detect Low-Rate DoS in IoT. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105059. [CrossRef]
165. Gupta, K.; Sharma, D.K.; Gupta, K.D.; Kumar, A. A tree classifier based network intrusion detection model for Internet of Medical Things. *Comput. Electr. Eng.* **2022**, *102*, 108158. [CrossRef]
166. Sohi, S.M.; Seifert, J.P.; Ganji, F. RNNIDS: Enhancing network Intrusion-Detection Systems through deep learning. *Comput. Secur.* **2021**, *102*, 102151. [CrossRef]
167. Apruzzese, G.; Andreolini, M.; Ferretti, L.; Marchetti, M.; Colajanni, M. Modeling realistic adversarial attacks against network intrusion detection systems. *Digit. Threat. Res. Pract.* **2021**, *3*, 1–19. [CrossRef]
168. Han, D.; Wang, Z.; Zhong, Y.; Chen, W.; Yang, J.; Lu, S.; Shi, X.; Yin, X. Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2632–2647. [CrossRef]
169. Han, D.; Wang, Z.; Chen, W.; Zhong, Y.; Wang, S.; Zhang, H.; Yang, J.; Shi, X.; Yin, X. DeepAID: Interpreting and improving deep learning-based anomaly detection in security applications. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 15–19 November 2021; pp. 3197–3217.
170. Jiang, H.; Lin, J.; Kang, H. FGMD: A robust detector against adversarial attacks in the IoT network. *Future Gener. Comput. Syst.* **2022**, *132*, 194–210. [CrossRef]
171. Liu, Y.; Tantithamthavorn, C.; Li, L.; Liu, Y. Deep Learning for Android Malware Defenses: A Systematic Literature Review. *ACM Comput. Surv.* **2022**, *55*, 1–36. [CrossRef]
172. Imtiaz, S.I.; ur Rehman, S.; Javed, A.R.; Jalil, Z.; Liu, X.; Alnumay, W.S. DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network. *Future Gener. Comput. Syst.* **2021**, *115*, 844–856. [CrossRef]
173. Rey, V.; Sánchez Sánchez, P.M.; Huertas Celdrán, A.; Bovet, G. Federated learning for malware detection in IoT devices. *Comput. Netw.* **2022**, *204*, 108693. [CrossRef]
174. Eremia, M.; Toma, L.; Sanduleac, M. The Smart City Concept in the 21st Century. In Proceedings of the 10th International Conference Interdisciplinarity in Engineering, INTER-ENG 2016, Tirgu Mures, Romania, 6–7 October 2016; Volume 181, pp. 12–19. [CrossRef]
175. Dameri, R.P. Searching for smart city definition: A comprehensive proposal. *Int. J. Comput. Technol.* **2013**, *11*, 2544–2551. [CrossRef]
176. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum PointNets for 3D Object Detection from RGB-D Data. *arXiv* **2017**, arXiv:1711.08488.
177. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-Task Multi-Sensor Fusion for 3D Object Detection. 2020. Available online: http://xxx.lanl.gov/abs/2012.12397 (accessed on 25 October 2023).
178. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-View 3D Object Detection Network for Autonomous Driving. 2017. Available online: http://xxx.lanl.gov/abs/1611.07759 (accessed on 25 October 2023).
179. Chen, H.; Li, Y.; Su, D. Multi-modal fusion network with multi-scale multi-path and cross-modal interactions for RGB-D salient object detection. *Pattern Recognit.* **2019**, *86*, 376–385. [CrossRef]

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.