electronics

MDPI

Article

Autonomous Threat Response at the Edge Processing Level in the Industrial Internet of Things

Grzegorz Czeczot, Izabela Rojek * and Dariusz Mikołajewski

Faculty of Computer Science, Kazimierz Wielki University, 85-064 Bydgoszcz, Poland; czeczot@ukw.edu.pl (G.C.); dariusz.mikolajewski@ukw.edu.pl (D.M.)

* Correspondence: izabela.rojek@ukw.edu.pl

Abstract: Industrial Internet of Things (IIoT) technology, as a subset of the Internet of Things (IoT) in the concept of Industry 4.0 and, in the future, 5.0, will face the challenge of streamlining the way huge amounts of data are processed by the modules that collect the data and those that analyse the data. Given the key features of these analytics, such as reducing the cost of building massive data centres and finding the most efficient way to process data flowing from hundreds of nodes simultaneously, intermediary devices are increasingly being used in this process. Fog and edge devices are hardware devices designed to pre-analyse terabytes of data in a stream and decide in realtime which data to send for final analysis, without having to send the data to a central processing unit in huge local data centres or to an expensive cloud. As the number of nodes sending data for analysis via collection and processing devices increases, so does the risk of data streams being intercepted. There is also an increased risk of attacks on this sensitive infrastructure. Maintaining the integrity of this infrastructure is important, and the ability to analyse all data is a resource that must be protected. The aim of this paper is to address the problem of autonomous threat detection and response at the interface of sensors, edge devices, cloud devices with historical data, and finally during the data collection process in data centres. Ultimately, we would like to present a machine learning algorithm with reinforcements adapted to detect threats and immediately isolate infected nests.

Keywords: Markov game; swarm; penetration testing; edge computing; computational fog; IIoT; IoT

Citation: Czeczot, G.; Rojek, I.; Mikołajewski, D. Autonomous Threat Response at the Edge Processing Level in the Industrial Internet of Things. *Electronics* **2024**, *13*, 1161. https://doi.org/10.3390/electronics13061161

Academic Editors: Wei Cui, Yaoming Zhuang and Wei Zhou

Received: 13 January 2024
Revised: 19 March 2024
Accepted: 19 March 2024
Published: 21 March 2024

1. Introduction

Industry 4.0 is advancing swiftly towards Industry 5.0. Many aspects of Industry 5.0 are already under consideration. It is no longer surprising that both households and large industrial complexes are equipped with sensors, automation control devices, and computing units, which employ data to enhance life and production. The increasing number of end devices and the growing volume of data that require processing and interpretation indicate the necessity to shift the focus of performing complex executive operations. It is crucial to process some of the gathered data in close proximity to the signal-emitting devices and to act upon it in realtime, while considering relevant historical data. Fog and edge computing are recognised as the appropriate means of data exchange and processing. This constitutes one aspect of the Industry 5.0 framework. The need to guarantee the cybersecurity of data transmitted through IoT networks has been widely discussed in numerous publications. Of course, with the ever-changing landscape of this concept, we take great care to uphold the essential protocols for safeguarding both the infrastructure and network traffic.

The IIoT and IoT networks predominantly rely on universal Internet access, though alternative channels dedicated solely to OT are utilised when Internet access is unavailable. This approach offers both merits and demerits. The benefits include the absence of the requirement to search for a medium capable of transferring immense amounts of data

between multiple end devices, as civil broadband connections, which are comparably inexpensive to upkeep, are exploited. Unfortunately, the ubiquity of this platform and its vulnerability to cyberattacks necessitate the creation of increasingly sophisticated and intelligent security measures against hacking and data breaches.

Therefore, it is crucial to contemplate security during the conceptual phase in designing IIoT and IoT infrastructure. Passive and partially active security measures, such as firewalls, constructing isolated subnets, and monitoring traffic with tools and teams analysing network traffic, exist. These solutions are based on principles that humans have developed. However, they are still vulnerable to deception and compromise by humans.

It is crucial to remain ahead of potential intruders by examining future developments theoretically. System security engineers are progressively utilising artificial intelligence to enhance their work. The security of smaller IT networks is commonly monitored by large teams of individuals supported by specialized tools. As cyber intruders and attackers are using more sophisticated infiltration methods, it is imperative to utilise all available strategies to combat and counteract their activities.

The aim of the defender is to safeguard the infrastructure against intrusions by monitoring the network and patching vulnerabilities. Conversely, the attacker is focused on undermining the infrastructure and gaining entry to a critical component. To attain this objective, the attacker must conduct reconnaissance to explore the infrastructure and attack components along the path leading to the critical component.

In this article, the focus is on providing methods for enhancing network traffic security and safeguarding IoT and IIoT infrastructure against vulnerabilities and exploitation. The article highlights the significance of network security and the need to detect and prevent vulnerabilities. Additionally, technical jargon will be explained when first used, and the language used will be passive, objective, and value neutral. Consistent adherence to style guides, citations, and footnote formatting is also maintained throughout the article. The aim is to identify methods for autonomous response procedures to intrusion attempts in IoT and IIoT networks. If security measures are implemented at the communication level between end devices and the fog and edge computing layers, it can safeguard most of the attack's vulnerable traffic. After analysing the topic of the Markov game, it is efficacious to start the conceptual process of developing autonomous neural networks.

1.1. Fog Computing in IoT

The concept of the Internet of Things (IoT) involves receiving data from sensors and transmitting the datathrough various levels of signals to a central processing unit. This unit continuously analyses the data and sends signals to controlling devices or procedures in almost real time.Increasingly, due to the limited hardware capabilities of on-premises devices, much of the processing is shifted to the cloud. This is due to its easier scalability for computing and data storage requirements. Unfortunately, the utilization of cloud solutions has constraints concerning access time. However, there is a lengthy delay in transmitting signals from the sensor to the computing unit and then returning the signals with a decision to the actuating devices. The notion of Industry 4.0 and its succeeding versions has shifted the focus towards the components facilitating the exchange process between the central computing unit and the sensor/actuator devices. Cloud-based, smaller instances were utilized to maintain quicker data access, albeit at the cost of computing power. This approach redirects specific signals from a limited number of IoT devices, resulting in a more efficient solution. It is important to note that the main computing instance is still responsible for managing Fog computing.

Cyberattacks commonly target both private and government entities that utilise cloud, edge, and fog computing. Developing a Threat Intelligence Platform (TIP) is crucial in providing protection for such architecture. In the industrial sector, safeguarding data is given utmost priority due to its sensitive nature. Extensive measures such as deploying intelligent equipment and sensor devices are implemented to minimize threats and potential security breaches. The characteristic of heterogeneity and geographic distribution has

an impact on the integration of cloud security frameworks into fog computing systems. Some of the security challenges that are considered include confidentiality, authentication, availability, and information privacy. These frameworks aid in creating and supervising access for individuals and organisations.

1.2. Edge Computing in IoT

Edge computing is a cutting-edge paradigm that brings services and applications to the closest locations to the data source, instead of relying on centralised cloud infrastructure. This approach provides computational power to process data and enhances the connection between the cloud and end-user devices.One effective method to address cloud computing problems is to augment the presence of edge nodes in a specific location. As the IoT is commonly used to process data at the edge of networks, edge computing is becoming increasingly popular [1]. A number of approaches, such as fog computing and edge computing, are providing complementary solutions to cloud computing by reducing the amount of data processing at the edge of the network. In order to increase availability, reduce latency, and ultimately overcome the problems of cloud computing, storage, computing, and power are being placed at the edge of the network [2]. This strategy will simultaneously alleviate the number of devices linked to a single cloud server and mitigate potential errors. Delay-sensitive and bandwidth-intensive applications can be processed close to the data source using edge computing [3]. Figure 1 shows a multi-layer model to deliver cloud-based IoT services at the edge.

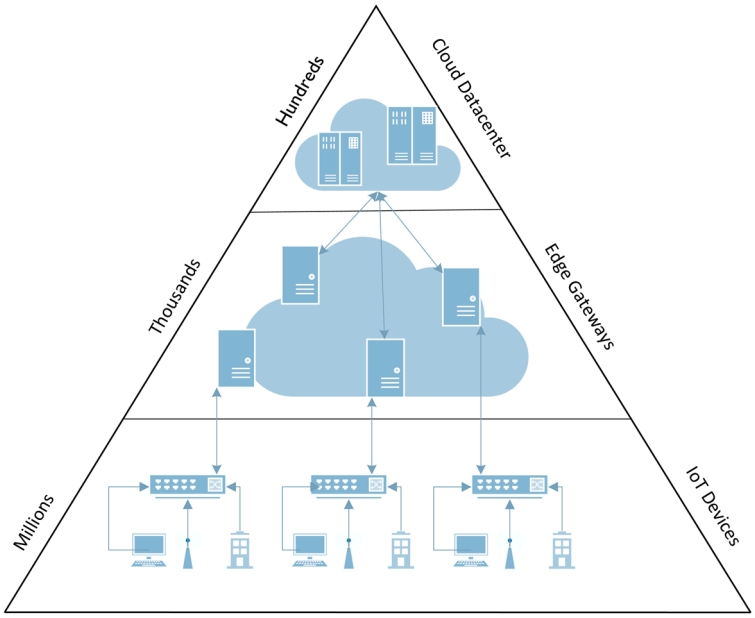

Figure 1. Layered model for cloud edge.

The rise in the number of mobile devices has made it difficult for traditional centralised cloud computing to meet the quality-of-service requirements of many applications. Edge computing is expected to be the solution to this problem, especially with the advent of 5G network technology [4,5]. The radio access network (RAN) is one of the major challenges associated with 5G technology. In the RAN, real-time RAN information can be provided through mobile edge computing. This information can be used by network operators to improve end-user quality of experience (QoE), as the real-time RAN provides context-aware services [6].

Edge computing improves infrastructure for IoT devices with limited data transfer capabilities, computing nodes, and paradigms. It solves limitations of connecting nodes with the cloud and enables countless benefits [2].

- Network management—involves eliminating connectivity issues and improving congestion control to enhance overall network performance, which is crucial in the IoT;
- Latency minimization—edge computing significantly improves the very important quality of service (QoS) in terms of minimizing delays in transmitting data to IIoT applications that require high responsiveness, i.e., autonomous vehicles;
- Cost optimisation—effective infrastructure planning using edge computing can significantly reduce costs compared to omitting the intermediary layer;
- Energy management—IoT-based edge computing aims to manage energy efficiently. Participants have a need for tight control over power management. IoT devices and applications that are energy efficient are desirable in edge computing. Studies suggest that 1 trillion IoT nodes will require sensor platforms that use power harvesting to provide scalability, reduce costs, and avoid frequent battery changes;
- Data management—efficient and effective data management mechanisms are desired in edge computing to handle the large amounts of data generated by IoT devices in a timely manner. Transmitting and aggregating these data are important considerations in data management;
- Resource management—to meet service-level objectives, the effective management of computing resources is critical. This involves coordinating resources, estimating availability, and allocating workloads appropriately.

The most important requirements of edge computing in IoT are as follows: dependability, latency, security, interaction in real time, and compatibility. Edge computing is used to reduce the amount of data sent to the cloud and to reduce the latency of accessing services.

1.3. Cybersecurity Role in Edge and Fog Computing

Cybersecurity plays a critical role in the context of fog and edge computing, offering several contributions. Fog and edge computing entail processing and storing data in proximity to edge devices, which pose new security challenges. Cybersecurity guarantees the confidentiality, integrity, and availability of sensitive data transmitted and stored within fog and edge environments. It encompasses measures such as encryption, access controls, secure communication protocols, and secure data storage. The communication between edge devices and fog nodes is relied upon in distributed networks. Cybersecurity deals with network-level threats, such as unauthorized access, network spoofing, man-in-the-middle attacks, and denial-of-service (DoS) attacks. Robust network security protocols and intrusion detection systems should be implemented to protect the communication channels and maintain secure connectivity. Physical attacks, tampering, or unauthorised access to edge devices in fog computing are possible. Cybersecurity techniques such as secure bootstrapping, device authentication, and secure firmware updates are used to ensure the security of these devices. By safeguarding the edge devices, the overall security of the fog computing system is enhanced. They generate large amounts of data that can be analysed in real time to detect anomalies and identify threats. Cybersecurity techniques, such as machine learning and behaviour analysis, can be applied to detect malicious activities, intrusions, or abnormal behaviour within the fog and edge network. Rapid threat detection enables timely responses and mitigates potential damages. Privacy concerns arise because fog and edge computing involve the processing and storage of sensitive data at the edge. To address these concerns, cybersecurity measures such as data anonymisation, access controls, and privacy-preserving techniques are employed to ensure that user data arehandled with care and in compliance with privacy regulations.

In fog and edge computing, cybersecurity involves monitoring the system for security events, collecting logs, and performing security analysis. Security management practices, such as vulnerability scanning, patch management, and security audits, help identify and address potential security weaknesses in the fog and edge infrastructure.By implementing backup and disaster recovery mechanisms, cybersecurity improves the resilience and continuity of fog and edge computing systems. This ensures that in the event

of a security incident or system failure, the system can recover quickly and continue to operate effectively.

Cybersecurity helps establish trust, protects data, and mitigates risks in fog and edge computing environments. This enables organisations to confidently leverage the benefits of these technologies.

2. Approach

The literature and professional experience suggest that IoT and IIoT networks have limited solutions to counteract intruder attacks. To achieve autonomous problemsolving with minimal human influence on decisions, we need to improve machine learning algorithms. It is important to refine these methods with each attack, with a focus on reinforcement learning. The cooperation between cloud computing and machine management has led to an increase in cyberattacks. This is a significant concern.

There are imperfect solutions available from individual vendors. However, using them often requires purchasing expensive licenses and support. In this case, the focus will be on vendor-agnostic solutions.

2.1. Comparison of Existing Cybersecurity Solutions in IoT

The main types of IoT cybersecurity threats are as follows: DoS attacks, taking control of devices left on default settings, exploiting security vulnerabilities on devices that are not regularly updated, taking control of devices whose configuration was incorrectly matched to the needs, the use of unencrypted connections, and data transfer.

When applying this form of counteracting cyber threats, we use solutions proposed by several leading vendors such as Nozomi, FortiNet, Claroty, and Armiwhich.These solutions involve monitoring the infrastructure at individual levels using sniffing devices, which use specific rules to classify anomalies according to the level of threat advancement.When highly dangerous events occur, the group of SIEM and SOC engineers supervising the infrastructure takes quick actions to eliminate the threat or performs preventive actions (patching vulnerabilities) based on analyses of reports generated by mechanisms implemented in the vendor's environment. Automated scripts triggered by specific correlation rules are also used.

Threat intelligence involvesthe collection of information about potential and existing cyberthreats, the analysis of data, and the application of the insights gained to anticipate, detect, and mitigate threats before they impact an organisation. Threat intelligence is more than just the installation of firewalls or anti-virus software; it is about understanding the threat landscape, the sources of threats, the methods used, and the potential targets.

Threat Intelligence means having a dedicated team of security professionals constantly on the lookout for signs of danger. It helps an organisation defend itself effectively by providing insight into the tactics, techniques, and procedures (TTPs) used by cybercriminals. Threat intelligence gives organisations an understanding of where their assets might be compromised, who might be doing it, and how they might do it.

Intrusion prevention systems (IPSs), also known as IDPSs, are security frameworks that monitor network activity and system operations for malicious activity. IPSs have four primary elements: detecting malicious activity, logging data about the activity, attempting to stop it, and reporting the incident [6]. There are various types of intrusion prevention systems (IPS), including the following:

- Wireless intrusion prevention system (WIPS)—monitors the wireless network and traffic for suspicious communication between nodes by analysing wireless network protocols;
- Network behaviour analysis (NBA)—identifies potential threats that result in atypical traffic, such as DDoS attacks, malware, or security breaches, which is achieved by analysing network traffic;
- The network-based intrusion prevention system (NIPS)—employs a comprehensive approach by monitoring network traffic and analysing activity on individual protocols;

- Host-based intrusion prevention system (HIPS)—constitutes a software designed to examine events from a particular host exhibiting suspicious behaviour.

An effective strategy to enhance cybersecurity in IoT-based networks could be the implementation of autonomous techniques like penetration testing. The literature suggests the use of particle swarm optimisation (PSO) analysis in penetration testing. This approach is prompted by the projection that households will possess an average of 30 IoT devices by 2030. The devised methodology assists in identifying susceptibilities in ever-evolving home IoT networks. Autonomous processes and the concept of swarm intelligence are being considered for effective vulnerability detection. This approach employs multiple agents concurrently testing the same environment, assuming that many parallel and similar processes will improve the vulnerability search better than a cyclical check of the entire network by one process. Success relies on the autonomy and self-control of these agents. The detection of vulnerabilities is crucial in low-power video network devices. It is essential to predict and counteract attacks on complex networks, which may deprive components of their ability to operate through DDoS attacks, benefiting from the lack of power supply and energy-saving advantages.

Counteracting cyberattacks also involves keeping the infrastructure updated on an ongoing basis. The use of periodic security update activities allows for the ongoing patching of detected vulnerabilities. Additionally, it is recommended to build infrastructures according to standards set by institutions such as the National Institute of Standards and Technology (NIST) or according to the recommendations of IoT device manufacturers.

2.2. Framework Based on Markov Game Model

In the game model, there are typically three objects: the attacker, the defender, and the user. The attacker's goal is to exploit system vulnerabilities to cause malfunctions or even the failure of the IoT system. The defender is concerned with improving the security of the system and mitigating potential threats by implementing reinforcement schemes. The user is concerned with the status of the IoT system. First, we build an SVM classifier for the automatic classification of the security state. Then, during the simulation game, we record the state of the IoT software chain at different times, calculating the probability of the actions of the participants based on the state changes caused by the different behaviours. At the end, we calculate the benefits of the three parties in the game under the different strategies. Once these steps have been completed, the defender can develop the most effective reinforcement scheme by constructing a Markov game model to identify threats to the IoT system.

Figure 2 shows the method, which comprises two main procedures: the classification of IoT security situations based on a support vector machine, and the detection of security situations based on a Markov game model.

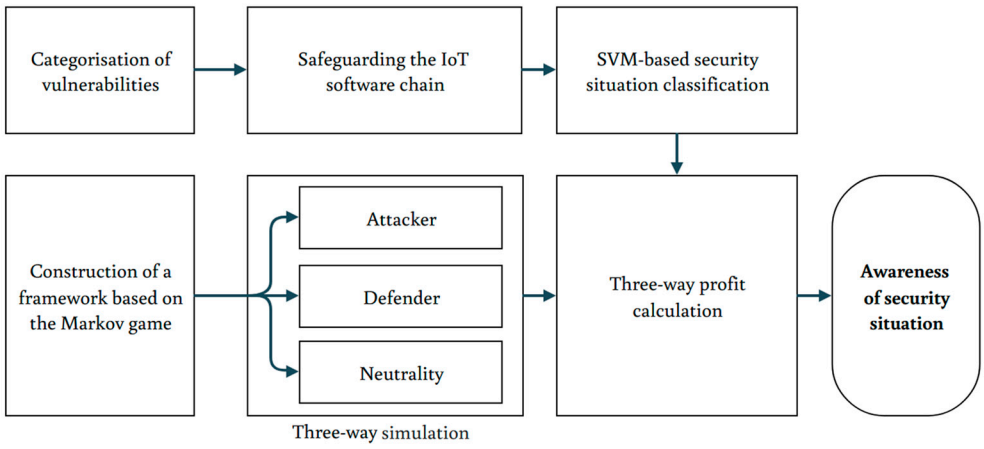

Figure 2. Framework of Markov game base method.

2.3. MGT in Cybersecurity

Due to the dynamic and adversarial nature of cyber threats, Markov game theory (MGT) has several potential applications in cybersecurity:

(1) Adaptive intrusion detection systems (IDSs)

In cybersecurity, IDSs play a key role by continuously monitoring network activity to detect and respond to potential security breaches. Markov game theory and reinforcement learning enhance IDSs, enabling them to adaptively respond to evolving cyber threats. The essential context for using IDSs with MGT includes the following:

- Interaction modelling: strategic interactions between attackers and defenders in a network environment are modelled using Markov games. The environment represents the network, while the players are the attackers, and the IDS agents are the defenders. Players can implement new detection rules, update firewall configurations, quarantine suspicious devices, and take other security-related actions.
- State representation: information about the network topology, current system configuration, traffic patterns, and the presence of known vulnerabilities or attack signatures are incorporated into the state space of a Markov game. To capture the dynamic nature of the cyber environment, states evolve over time based on network activity and defender actions.
- Selection of actions: AIDS agents are trained to select actions that minimise the risk of successful attacks while minimising disruption to legitimate network traffic using reinforcement learning algorithms. These actions may include implementing new detection algorithms, updating access control policies, or activating incident response procedures.
- Rewards structure: rewards provide IDS agents with feedback based on their actions and their impact on the security state of the network. Rewards can be based on factors such as identifying suspicious activity, preventing successful attacks, or minimising false alarms. As cumulative rewards are optimised over time, IDS agents learn to make decisions that improve overall security.
- Exploration vs. exploitation: in order to adapt to changing threats, IDS agents try to strike a balance between exploration and exploitation. Reinforcement learning (RL) algorithms allow IDS agents to explore new defence strategies while leveraging existing knowledge of attack patterns and network vulnerabilities.
- Learning and adaptation: IDS agents learn to recognise patterns of malicious behaviour, adapt their detection strategies, and proactively respond to emerging threats by interacting with the environment and receiving feedback from reward functions. IDS agents can continuously improve their performance over time using reinforcement learning algorithms such as Q learning or deep learning with reinforcement.
- Real-time decision making: intrusion detection systems (IDS) based on Markov games operate in realtime, constantly monitoring network traffic and making quick decisions to mitigate security threats. Based on the current threat landscape and evolving tactics of attackers, these systems use reinforcement learning to dynamically adjust their behaviour.
- By integrating Markov game theory and reinforcement learning, IDSs can become more adaptive, resilient, and effective in defending against sophisticated cyber threats in dynamic network environments.

(2) Vulnerability patching and configuration management

Using Markov game theory and reinforcement learning, it is possible to plan and prioritise security updates and configuration changes to reduce the risk of exploitation and minimise the impact on the system. Below is a detailed overview of how they can be used in this regard:

- Interaction modelling: strategic interactions between system administrators responsible for patching and configuration management and potential attackers looking to exploit security vulnerabilities can be modelled using Markov games. The players are the administrators and the attackers, while the environment represents the OT infrastructure. Administrators can take actions such as planning patch deployments, configuring security settings, and updating software versions, while attackers choose which vulnerabilities they want to exploit and when to attack.
- State representation: Markov state spaces contain information about the current state of the system, installed software versions, known security vulnerabilities, patching status, network traffic patterns, and historical intrusion information. Reflecting the dynamic nature of the cyber environment, states evolve over time based on administrator actions, system updates, and attacker actions.
- Selection of actions: administrators are trained to select actions that effectively mitigate security risks while minimising disruption to system operations using reinforcement learning algorithms. These actions may include prioritising patches based on severity, criticality, and potential impact on operations; scheduling the deployment of patches during maintenance windows; and balancing security needs with business continuity requirements.
- Rewards structure: the rewards function provides feedback to administrators based on their actions and their impact on the security status of the IT infrastructure. Rewards are based on factors such as successfully deploying patches, preventing successful attacks, reducing security vulnerabilities, and minimising downtime. Administrators learn to make decisions that improve overall security while minimising operational impact by optimising cumulative rewards over time.
- Exploration vs. exploitation: to effectively manage security threats, administrators balance exploration and exploitation. Reinforcement learning enables administrators to explore new patch policies and configuration management practices while leveraging existing knowledge of best practices for vulnerability management and system maintenance.
- Learning and adaptation: administrators can learn to identify patterns of vulnerabilities, effectively prioritise patches, and proactively respond to security threats through interactions with the environment and feedback from reward functions. Administrators can continuously improve their patching and configuration management practices over time using reinforcement learning algorithms such as Q-learning or deep learning with reinforcement.
- Real-time decisionmaking:this involves constantly monitoring new security vulnerabilities, assessing their potential impact, and making quick decisions based on risk mitigation. Based on the current threat landscape, business priorities, and system requirements, these systems use reinforcement learning to dynamically adjust their patching policies and configuration management practices.
- By integrating Markov game theory and reinforcement learning, organisations can manage security vulnerabilities, prioritise patch distribution, and optimise configuration settings to minimise exploitation risk while maintaining system availability and performance.

(3) Moving Target Defence (MTD)

Moving Target Defence (MTD) is a cybersecurity strategy that aims to increase the resilience of systems and networks by dynamically changing their attack surface to confuse and thwart attackers. To design and implement effective MTD strategies, Markov game theory and reinforcement learning can be applied. This text provides a detailed overview of the use of these techniques in this area:

- Modelling the interaction: Markov game theory is used to model the strategic interaction of defending players applying MTD techniques and attacking players trying to exploit vulnerabilities. In this model, the environment represents the IT infrastructure or network, while the players are the defending and attacking parties. Defenders have the ability to dynamically reconfigure the network, change system parameters, and apply deception techniques, while attackers choose which vulnerabilities to target and when to attack.
- State representation: information about current system configurations, network topology, implemented MTD techniques, attacker behaviour, and historical attack data is incorporated into the state space of the Markov game. Reflecting the dynamic nature of the cyber environment, states evolve over time based on defenders' actions, system updates, and attackers' actions.
- Action selection: reinforcement learning is used to teach defence systems to select actions that effectively disrupt and deter adversaries, while maintaining system usability and performance. These actions can include rotating IP addresses, changing firewall rules, modifying software configurations, and implementing honeypots or decoys. The goal of the defender is to dynamically adjust the attack surface to increase attacker uncertainty and reduce the likelihood of successful exploitation.
- Reward structure: rewards provide feedback on the defender's actions and their impact on the state of the IT infrastructure. Rewards can be based on factors such as successfully detecting attacker activity, preventing successful attacks, reducing attacker dwell time, and minimising false alarms. Defenders can learn to make decisions that improve overall security while maintaining system usability and performance by optimising cumulative rewards over time.
- Exploration vs. exploitation: defenders strike a balance between exploration and exploitation to effectively confuse and deter attackers. Reinforcement learning algorithms allow the defender to explore new MTD techniques and deployment strategies while leveraging existing knowledge of attacker behaviour and system vulnerabilities.
- Learning and adaptation: defenders can learn to recognise patterns of attacking behaviour, effectively prioritise MTD techniques, and proactively respond to security threats through interactions with the environment and feedback from the reward function. The use of reinforcement learning algorithms, such as Q-reinforcement learning or deep reinforcement learning, allows defenders to continuously improve their MTD strategies over time.
- Real-time decisionmaking:this involves constantly monitoring intruder activity, assessing its potential impact, and making quick decisions that disrupt and deter intruders. Based on the current threat landscape, attacker tactics, and system requirements, these systems use reinforcement learning to dynamically adapt their MTD techniques and deployment strategies.

The integration of Markov game theory and reinforcement learning enables organisations to deploy effective Moving Target Defence (MTD) strategies to make their systems and networks more resilient to sophisticated cyber threats. MTD techniques dynamically change the attack surface and confuse attackers, thereby reducing the risk of successful exploitation, while maintaining system availability, confidentiality, and integrity.

(4) Cyber threat hunting and adversarial red teaming

Markov game theory and reinforcement learning can be applied to simulate and analyse strategic interactions between threat hunters/red teams and adversaries in a cyber environment in the context of cyber threat hunting and adversarial red teaming. This section of the article provides a detailed overview of their application in this aspect.

- Modelling the interaction: Markov games can be used to model strategic interactions between threat hunters and adversaries attempting to penetrate or destroy systems. The environment represents the cyber infrastructure, while the players are threat hunters and adversaries. The actions available to the threat hunter consist of deploy-

ing defences, analysing network traffic, and conducting reconnaissance, while the adversary selects attack vectors, exploits vulnerabilities, and tries to avoid detection.

- State representation: the state space of a Markov game contains information about current system configurations, network topology, implemented defence mechanisms, attacker behaviour, and historical attack data. Reflecting the dynamic nature of the cyber environment, states evolve over time based on threat hunter actions, system updates, and adversary actions.
- Action selection: reinforcement learning algorithms are used to train threat hunters to select actions that are effective in detecting and responding to adversary actions, while minimizing false positives and maintaining system usability.These activities may include implementing intrusion detection systems, analysing logs and network traffic, conducting threat analysis, and deploying decoys or honeypots.To counter adversary tactics and avoid detection, threat hunters try to dynamically adapt their tactics and techniques.
- Reward structure: rewards provide feedback to threat hunters based on their actions and impact on the security posture of the cyber infrastructure.Rewards can be based on factors such as successfully detecting adversary activities, identifying attack vectors, preventing successful attacks, and minimizing false positives.Threat hunters learn to make decisions that improve overall safety while minimizing operational impact, optimizing cumulative benefits over time.
- Exploration vs. exploitation: threat hunters identify and neutralize adversary activities, balancing exploration and exploitation.Using reinforcement learning algorithms, a threat hunter is able toanalyse an opponent's tactics and exploit weaknesses in their strategies.
- Learning and adaptation: threat hunters learn to recognize adversary behaviour patterns, effectively prioritize hunting activities, and proactively respond to security threats through interactions with the environment and feedback from the reward function.To continually improve threat hunting capabilities over time, reinforcement learning algorithms such as Q-learning or deep reinforcement learning are used.
- Real-time decision making: this involves constantly monitoring and analysing adversary activity and its potential effects, and then quickly making decisions about detection and response. Based on the current threat landscape, adversary tactics, and system requirements, these systems use reinforcement learning to dynamically adapt their threat hunting strategies and deployment tactics.

The integration of Markov game theory and reinforcement learning enables organisations to implement effective cyber threat hunting and adversarial red teaming strategies. These strategies increase the resilience of systems and networks against sophisticated cyber threats. These techniques help identify vulnerabilities, detect adversary activity, and proactively mitigate security threats by simulating strategic interactions between threat hunters and adversaries. Combining Markov game theory with reinforcement learning holds promise for improving cybersecurity defences by enabling adaptive, proactive, and strategic decision making in the face of complex and evolving cyber threats.

Game theory and Markov Decision Processes (MDPs) form the basis of the Markov game model. Game theory deals with decision making in situations where multiple participants are interacting, while MDPs involve making choices from a set of available actions based on observed information or states. The state transition can be traced using Markov probabilities, even though the next state may be random. That is, the next state is only related to the current time. This allows for the dynamic analysis of potential hazards.

The following is a list of the basic components of the Markov game model:

- Participants can be divided into three categories: attackers, defenders, and users. Attackers perform malicious attacks to disable IoT software chains, while defenders implement security solutions to reduce vulnerability and improve IoT software chain security. Users only operate devices and are not concerned with security;
- Situational space refers to all possible IoT software chain situations;
- Behavioural space consists of all possible actions of the three participants;
- Transition probability refers to the way that the IoT system evolves based on how the participants behave; the situation is constantly changing. Participants can select appropriate behaviours from the behavioural space based on transitions and security assessments of the IoT system with a certain probability;
- Reward function refers to rewards for each participant based on their respective goals. The attacker aims to cause maximum damage to the system, while the defender seeks to enhance its security. Users, on the other hand, require sufficient network resources, and their reward is based on the degree of system service utilization.

Based on the current system state, participants choose a behaviour from the behavioural space in the game process. The participants then make decisions based on the new condition, and the system moves to a new condition. The system moves to the new state and the process is on until the state is in place. The three participants decide their behaviour and receive their reward for a given threat. To describe the rewards that each participant will receive, our method uses a reward function [7–10].

The aim of each participant is to maximise the reward function. This process is quantitatively described as follows:

$$TPN(t,k) = \left\{S_i(k), e_j(k)\right\}$$

where $S_i(k)$ is the state of the i-th propagation node at time k; $e_j(k)$ is the state of the j-th propagation path at time k. Simply, $TPN(t, k + 1)$ is the systemstate at time $k + 1$,and the state change of the system follows the Markov rule.

Attackers must analyse the nature of the threat t and determine how to exploit it for the greatest benefit. Defensive measures implemented by the administrator on node i will have two impacts:

(1) reducing the damage caused by threats that affect the IoT software chain;
(2) the security plan should aim to minimize its impact on the availability of node i.

$$V_s(S_i(k)) = \Delta\rho_{ai} \cdot value_{ai}$$

where $\Delta\rho_{ai}$ represents the change in node utilization performance, measured by the difference between the node utilization value before and after the security enhancement is implemented; and $value_{ai}$ represents the availability of node i. User reward is measured by the sum of the usage ratio of N nodes and the amount of the M utilization path.

To summarise, IoT security situation awareness involves three main procedures: data processing, constructing a threat propagation network, and evaluating IoT security using a Markov game model.

Algorithm: IoT security situation awareness based on Markov game model (Algorithm 1).
Inputs: IoT security data.
Output: IoT security situation.

- Pre-process the security data related to IoT;
- Create a threat propagation network for each threat 't' based on the security information;
- Based on the threat propagation network, construct a Markov game model for 't' and calculate the security situation of 't'. Follow a logical sequence to complete the task;
- Analyse the defender's best reinforcement plan to deal with threat t;
- Summarise the damage of all threats. Evaluate the overall security of the IoT software chain based on different requirements.

Algorithm 1. Markov threat reactionstrategy pseudocode

```
# Define the states
states = [state1, state2, state3, . . .]

# Define the transition probabilities matrix
transition_probs = [
 [prob11, prob12, prob13, . . .],
 [prob21, prob22, prob23, . . .],
 [prob31, prob32, prob33, . . .],
 . . .
]

# Define the initial state probabilities
initial_probs = [prob1, prob2, prob3, . . .]

# Simulate the Markov chain
current_state = select_initial_state(initial_probs)
for i in range(num_iterations):
 next_state = select_next_state(current_state, transition_probs)
 current_state = next_state

 # Analyze the state
 if current_state == vulnerable_state:
 alert_vulnerability()
 elifcurrent_state == attack_state:
 alert_attack()

# Function to select the initial state based on initial probabilities
def select_initial_state(initial_probs):
 # Perform a weighted random selection based on probabilities
 . . .

# Function to select the next state based on transition probabilities
def select_next_state(current_state, transition_probs):
 # Perform a weighted random selection based on probabilities
 . . .

# Function to alert vulnerability
def alert_vulnerability():
 # Take appropriate actions to handle the vulnerability
 . . .

# Function to alert attack
def alert_attack():
 # Take appropriate actions to handle the attack
 . . .
```

In this pseudocode, *states* represent the possible security states in the IoT network, *transition_probs* is the matrix representing the transition probabilities between states, and *initial_probs* represents the probabilities of starting in each state. The *select_initial_state* and *select_next_state* functions perform weighted random selections based on probabilities. The *alert_vulnerability* and *alert_attack* functions are called when the current state corresponds to a vulnerability or an ongoing attack, respectively.

2.4. Quick Response Module

To potentially isolate infected end devices at the fog or edge computing level, the neural network would require data from penetration tests of the IDS module. The objective

of this extensive autonomous attack response system should be to adjust the scope smoothly. Figure 3 provides an outline of the concept on which the process for autonomous responses to network infrastructure threats would operate.

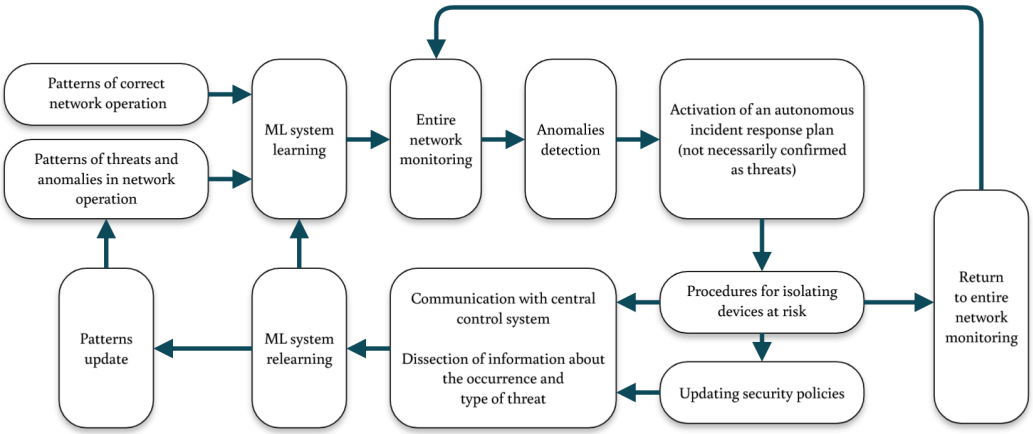

Figure 3. System overview.

The basis for the operation of the whole system is based on the correct patterns of the operating network infrastructure. We will start the description of the method presented from the upper left corner of the Figure 3:

(1) The states of the correct and incorrect (not necessarily disrupted/attacked, but also subject to warnings/intervention as a part of predictive maintenance) operation of IIoT devices are described by state vectors that are optimised and shortened for processing efficiency, classification, and downstream prediction according to edge computing principles;
(2) The analysis takes place as close to the source as possible, so if the devices are working correctly, no additional information needs to be transmitted, which relieves the burden on devices and transmission networks and allows the lowest possible level of response, e.g., within IIoT devices, software, and quality of service (QoS) mechanisms;
(3) The assumed configurations of the network model and how it operates are sent to the machine learning system as an indicator of the correct operation of the overall system;
(4) The entire network is then monitored for any deviations, anomalies, or potential threats (but the entire network is subdivided by device, i.e.,within the structure in Figure 1, so that responses can be made at the level of devices or their groups or network segments—as low as possible);
(5) Autonomous response protocols are triggered when an anomaly or threat is detected, whether it is a threat to data security integrity or simply a deviation from normal operations (e.g., a predictive maintenance response is needed)—this approach allows for better maintenancesystem integrity regardless of operating conditions;
(6) Events are identified, classified, and prioritized;
(7) When an attack is expected to be imminent, vulnerable devices are isolated from the network. Such a threat is properly identified and categorized, and the result of this process is an update of the security policy;
(8) From this point, two processes run in parallel:
    - The first is to restore the monitoring of the whole network—after isolation;
    - The second is communication with the central unit (or cloud processing) that controls the system and analyses information about the occurrence and nature of the threat.
(9) Information about the threat is sent to the machine learning system, which, after running the algorithm, updates patterns of proper system operation.These patterns are enriched with the “experience” of the existing threat/anomaly and the method of classifying the event.

(10) Such monitoring is maintained throughout the system's operation, but the training of the ML system during technical breaks is preferred [11–13].

Main mechanism of such system are analyzed in Table 1.

Table 1. Overlaying the concept onto IIoT framework.

IIoT Level	Mechanism	Related Strategy
Cloud Data Centre/Fog	General control on edge configuration and operations Data analysis and prediction	Central control on IT operations
IIoTgateways	Access control Surveillance	Enhancing physical security at the edge Additional computing layer performing cybersecurity functions (e.g., multiaccess edge computing)
IIoT devices	Monitoring of configuration and logs Application changes control Data control Auditprocedures	Edge activity of IIoT devices as public cloud operations Communication between devices/users and edge facilities constitutes high risk activity Application of the highest possible security level
Users	Guiding users on the basis of necessary authorisations Maintenance	Elimination of human errors Lowering costs and waste of time Enhanced lifetime

3. Methods

There is a significant research gap in this field since current e-commerce vendor solutions continue to rely on human intervention. Imperfect programs and systems that analyse network traffic are used, but they are improved regularly. Additionally, expensive software necessitates substantial human resources, including security engineers and analysts, to continuously monitor network traffic and its potential deviations from the set security standards. Of course, currently, it is believed that no tools can completely replace the human factor that fulfils its task. However, this work aims to find ways to augment cybersecurity teams' potential and reduce their reliance on human intervention.

It is reasonable to assume that detecting vulnerabilities early is a simpler approach to ensuring cybersecurity. This can be performed through autonomous algorithms which can search for vulnerabilities in networks, for example through penetration testing. A greater challenge is probably developing a way to counteract attacks and promptly detect the aggressor or attacked element.

The hypothesis underlying this consideration posits the possibility of creating a machine learning algorithm with suitable training data to prime the desired network for utilising an autonomous defence mechanism.

Autonomous threat detection will be introduced, along with simplified response procedures at the edge computing level. This will result in a uniform reaction time and increased effectiveness, surpassing that of the currently used centralized mechanisms.

Edge computing will process optimized threat feature vectors, rather than entire anomalies detected through monitoring.

IIoT network segmentation can facilitate the prompt elimination of infected vulnerabilities without the need to disable entire subnets or complex systems. Our goal is to isolate individual IIoT devices through gateway management.

The system will utilise adaptable machine learning. Patterns resulting from continuous monitoring and analysis will enable us to identify even the smallest deviations and anomalies, predicting events that require sudden and drastic responses. This will enhance the system's accuracy.

3.1. DataSet

We have used the Edge-IIoTset Cyber Security Dataset of IoT & IIoT [14,15]. The main parts of the dataset (.csv files) include selected datasets for ML (157,800 records):

(1) Normal traffic

- Distance: gathered by IoT ultrasonic sensor;
- Flame: gathered by IoT flame sensor;
- Heart_Rate: gathered by IoT heart rate sensor;
- IR_Receiver: gathered by IoT infrared receiver;
- Modbus: gathered by IoT modbus sensor;
- phValue: gathered by IoT ThpH sensor PH-4502C;
- Soil_Moisture: gathered by IoT Soil Moisture Sensor v1.2;
- Sound_Sensor: gathered by LM393 Sound Detection Sensor;
- Temperature_and_Humidity: gathered by IoT DHT11 Sensor;
- Water_Level: gathered by IoT water sensor.

(2) Attack traffic specific for attacks:

- Backdoor Attack;
- DDoS Http Flood Attack;
- DDoS ICMP Flood Attack;
- DDoS TCP SYN Flood Attack;
- DDoS UDP Flood Attack;
- MITM Attack;
- OS Fingerprinting Attack;
- Password Attack;
- Port Scanning Attack;
- Ransomware Attack'
- SQL Injection Attack;
- Uploading Attack;
- Vulnerability Scanner Attack;
- XSS Attack.

3.2. Computational Methods

The computational approach covers several levels of analysis shown in Figure 4 from edge computing (close to the source, within a single segment, based on AI) to central AI-based management at the level of the entire system or its segments. Machine learning, based on a distributed, local approach to data collection and preliminary analysis (edge computing), is used, combined with the Markov gaming platform, and may be a potential solution to various challenges related to processing speed, data confidentiality, and cybersecurity.

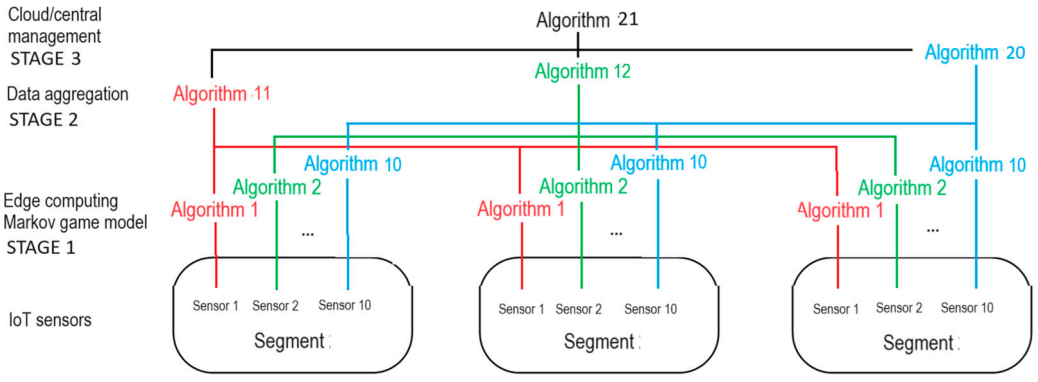

Figure 4. General architecture of AI-based part of the system. Colors show the analysis paths for sensors of the same type (for the three sensor types: 1, 2 and 10).

SVM-based safety situation classification was replaced by three-stage learning due to its simpler design and faster response. A feature vector describing the state of a sensor/IoT device allows for the classification of the device (based on the reported state of the device and its activity/load) into one of the roles in the Markov game: neutral, attacker, or defender. The desired state is neutral. The attacker and defender states are isolated from the rest of the network at the segment level—this is the primary way to respond to an attack. The advanced behaviour of the response system emerges from the rather simple interactions of a range of models.

Signals from IoT network/sensor devices in the form of optimised feature vectors representing state vectors are processed in three stages:

- In Stage 1, the vectors are recalculated according to Markov game rules, and their initial local analysis (edge processing) and modelling is performed—this stage is the most demanding both in terms of time and accuracy, as it provides fast responses to the states of feature vectors that meet the criteria for misbehaving devices (including attack).Depending on the types of devices or sensors, the data is assumed to be processed according to one of ten algorithms (the data for each of the ten algorithms are collected in a different network); moreover, at this stage, a network segment can be isolated as "attacked" or "damaged";
- In Stage 2, data from individual segments are aggregated and modelled using algorithms 11–20—this level of processing is responsible for the synchronised aggregation and evaluation of the overall network level of one type of sensor/device;
- In Stage 2, a general overview of the performance of the entire network (regardless of the type of sensors/devices in the network) takes place.

In Stage 1, the operation is performed in two stages: first, there is a recalculation of the feature vectors according to the Markov game rules presented earlier, and then, there is the learning of the models. The optimisation of feature vectors mainly concerns their form and length, i.e., features that do not affect the accuracy of the operation are cyclically evaluated and removed. Reducing the size of the feature vectors speeds up the processing and therefore the response time [16–19].

For the implementation of the first version of our solution, we chose ML.NET (OpenML) in VisualStudio 2022 because of the large number of algorithms to be tested (in our case: more than 50), the fast semi-automatic learning and testing, and the possibility to download a file or API for further use in the system under construction. The automation and flexibility of ML environments here is cloying not only for the proper implementation of the solution, but also for the large amounts of data (almost 200,000 records) and the need for cyclic learning on new data. The challenge becomes balancing the classes, which will be difficult (correct operation is the leading class). With flexibility, it will also be possible to better adapt, for example, to new generations of IoT sensors or to connect new segments and even whole clusters of them, including in stages, with gradual learning. The use of simple algorithms is expected to rule out the 'black box' phenomenon and preserve human understanding, even as systems evolve and grow in complexity.

4. Results

The algorithms considered the best for analysing individual data are shown in Figure 5 and Table 2.

Table 2 shows the results for the Stage 1 algorithm (edge computing, Markov game model)—the best of which was the LightGbmMulti algorithm. It is notable that the second, third, and fifth places here were occupied by the FastTreeOva algorithms, which learned on the same data but with different hyperparameters.

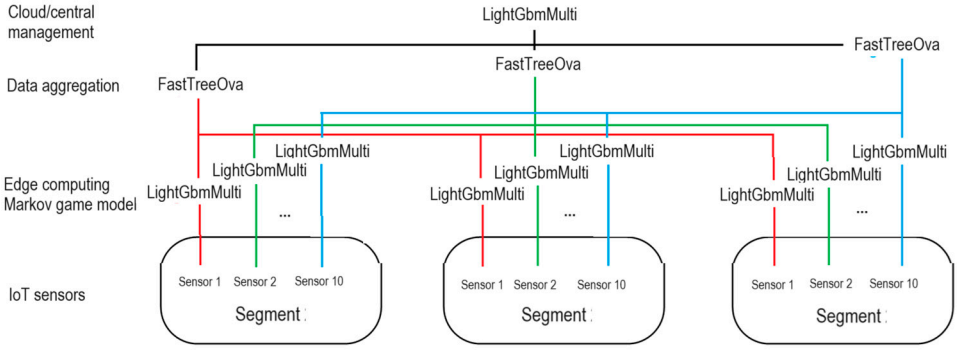

Figure 5. AI algorithms used within the system.

Table 2. Best algorithms for attack type assessment.

Agorithm	Accuracy	MRSE
LightGbmMulti	0.9347	0.001
FastTreeOva	0.8959	0.002
FastTreeOva	0.8901	0.002
FastForestOva	0.8892	0.002
FastTreeOva	0.8880	0.002

FastTreeOva (accuracy 0.9101) was considered the best data aggregation algorithm (Stage 2). Similarly, the Stage 2 algorithms are 10 FastTreeOva algorithms taught on different data, so they are different algorithms.

The algorithm considered the best for cloud management (central/overall view, Stage 3) is LightGbmMulti (accuracy 0.8991).

The challenge with detecting attacks on our infrastructure may relate to the fact that attack attempts may remain inadequately visible.

A comprehensive IoT network typically involves a similar technology, with comparable characteristics of end devices and often the same transmission protocol.

Accordingly, a potential intruder can exploit these factors to our disadvantage. The susceptibility of intrusion detection systems is frequently due to the exclusion of incidents that occur sporadically across various devices analysed at different times. To reduce the need for analysing a vast number of occurrences, experts have established benchmarks. A single deviation from the expected norm for network traffic security systems, which does not surpass the established thresholds, could potentially indicate a hostile intruder. In-depth security testing of comparable infrastructure components may lead to compelling evidence of vulnerability, thereby enabling attackers to target the entire system or significant portions of it simultaneously [20–23].

When selecting a security system, one must therefore exercise great caution. The Autonomous Reaction Mechanism of Potential Intruder Attack (ARMPIA) should be approached comprehensively, with a system consisting of three strict elements to support the reinforcement learning-based machine learning algorithm.

5. Discussion

Analysing previous studies, a review by Dino and Manivannan [24] showed that there are still many open issues in this area that need to be addressed. The use of ML techniques is one of the leading approaches used to classify reasonable and unusual behaviour in the IIoT, but so far only a few have been described in a way that allows for a comparison with our concept. ML-based IDS/IPS in combination with other security elements (e.g., firewalls) are used to identify and combat threats, abuses, intrusions, and unauthorized behaviour. Masdan and Khezri conducted astudy of fuzzy fraud detection

schemes designed using various machine learning and data mining techniques to deal with different types of intrusions, including the application of fuzzy datasets, feature extraction, and performance evaluation [25]. Some IIoT systems have to use multi-access edge computing (MEC) technology due to limited resources and the inability to deploy advanced security systems. This, in turn, places high demands on offload latency, synchronisation and execution time, and concurrent energy efficiency to ensure near real-time responses. The problem of offloading and synchronising tasks can be represented by a mathematical model based on Markov transition probabilities (i.e., estimating the clock offset using maximum likelihood) [26]. Attack methods are being developed that involve generating (using particle swarm optimisation, genetic algorithms and deep learning—generative adversarial networks) opposing examples capable of bypassing various machine learning models. This indicates new vulnerabilities in ML-based systems [27]. Datasets are becoming crucial and can give you an advantage despite the existence of at least 35 generally known cyber datasets classified into seven categories (based on network traffic, electrical network, Internet traffic, virtual private network, Android applications, IoT traffic, and devices connected toInternet).Accuracy, the false alarm rate, and the detection rate are commonly used to evaluate the performance of ML methods. This has so far identified at least seven useful ML models for binary and/or multi-class classification: recurrent or deep neural networks, constrained or deep Boltzmann machines, deep belief networks, convolutional neural networks, and deep autoencoders [28]. Detecting known attacks, blocking suspicious behaviour in near real time, and dynamically updating security rules can only partially improve the situation [29]; hence, our concept seems to go a step further into the future compared to the solutions presented so far in the literature.

A comparison and discussion of our own concept with other similar concepts from the literature is presented in Table 3 in the form of an analysis of strengths, weaknesses, opportunities, and threats (SWOT).

Table 3. SWOT analysis of ourown concept.

	Positive	**Negative**
Internal	Strengths Local, distributed, focused reaction Integrationof various levels of processing (but edge computing is preferred overcloud computing) Intrusion prevention system (IDS) is preferred over intrusion detection system (IPS) Both host-based and network-based system: single devices, segments, and lines can be cut-off and/or bypassed depending on needs Combined anomaly-based system with signature-based system, but database is built dynamically with adapted normal activity (not only from historical data) Robustness Quick reaction	Weaknesses Advanced knowledge needed Necessity of building for the near future, not quite now The best solutions are dedicated, not universal Higher initial cost Achieving assumed intrusion prevention level may require use of various comprehensive solutions (e.g.,toavoid false positive detections), such as firewalls, antivirus, SIEM, and SOAR Need for updates
External	Opportunities Scalability Adapted to assumed risk tolerance and business strategy Shorter, optimized datasets Various AI solutions may be used/combined (decision trees, fuzzy logic, neural networks, deep learning) Visualization and analytics Better understanding of network and system behaviour More holistic and proactive cybersecurity strategy Lower complexity Greater resistance to human errors (e.g., users' behaviour) Reduced downtime and damages	Threats The need to keep sensitive details of the operation secret—partial "black box" (may create lack of transparency about certain decisions) Lack of social awareness and acceptance

5.1. Limitations of Own Research

The suggested approach for identifying and addressing potential risks features specific constraints arising from the flawed signals sent over IoT and IIoT networks:

- With augmented traffic in the network, there is a high suspicion of distortions, leading to a loss of coherence in the data that could potentially cause false-positive alarms;
- Regrettably, the quantity of false-positive alarms surpasses the potential attacks by intruders—therefore, a limitation is to accurately differentiate these genuine attacks from the irrelevant ones;
- A challenge in classifying alarms arises due to the diversity of firmware versions—it is imperative to keep the devices' versions in the network as current as possible and to utilize devices from vendors with minimal diversification [30–34].

5.2. Directions for Further Studies

Autonomous threat response at the edge processing and fog level in the IIoT is a complex and evolving field. To delve deeper into this area of research, the following research directions are proposed:

- Develop and explore security standards applicable to the IIoT and fog computing and how they can evolve to support autonomous responses to new emerging threats [35–40];
- Explore different edge processing and fog computing architectures for IIoT applications and analyse how these architectures deal with security and threat responses [19,41–43];
- Explore the use of AI algorithms (specifically ML algorithms) to detect and respond to threats at the network edge and fog level, train them, and optimize them for IIoT environments [44–46];
- Utilize real-time analytics and anomaly detection techniques to identify unusual behaviour or potential security risks;
- Integrate edge and fog security measures with cloud-based security solutions, i.e., investigate how data collected at the network edge can be securely transmitted and stored in the cloud [47–51].

We obtained a balanced view and set a clear path for future research by integrating the aforementioned limitations of our current research with future research directions. This allowed priorities to emerge and yielded the following sequence of further research activities (broken down by level of task execution):

- Automation of IoT support in edge computing;
- Intelligent creation of trusted edge data for distributed IoT;
- Adaptive configuration of IoT applications in fog infrastructure;
- Utilization of cryptographic schemes for secure edge processing, including server deployment in multi-user edge processing;
- Intelligent management of resources in fog computing using RL, including multi-user placement of IoT services with QoS consideration [52–62].

6. Conclusions

An industry that uses more and more information technology is vulnerable to attacks and malicious behaviour.The use of ML in the realm of cybersecurity is becoming more prevalent with each passing day.In this research project, we have compiled the literature on reinforcement learning and categorised it into three areas: penetration testing, IDSs, and cyberattacks. Even though ML for cybersecurity is a new topic in the literature, it has been established that the techniques used give very encouraging results for the detection and prevention of attacks. It is anticipated that RL will formulate techniques for cyber defence and attack and assist in the transformation of cyber risks. The concept introducedin this study is expected to address cybersecurity issues and propose solutions, establish the groundwork for future research, and significantly influence its direction.

Author Contributions: Conceptualization, G.C., I.R. and D.M.; methodology, G.C., I.R. and D.M.; software, G.C., I.R. and D.M.; validation, G.C., I.R. and D.M.; formal analysis, I.R. and D.M.; investigation, G.C., I.R. and D.M.; resources, G.C., I.R. and D.M.; data curation, G.C., I.R. and D.M.; writing—original draft preparation, G.C., I.R. and D.M.; writing—review and editing, G.C., I.R. and D.M.; visualization, G.C., I.R. and D.M.; supervision, G.C., I.R. and D.M.; project administration, G.C., I.R. and D.M.; funding acquisition, G.C., I.R. and D.M. All authors have read and agreed to the published version of the manuscript.

Funding: The work presented in this paper has been financed under grant to maintain the research potential of Kazimierz Wielki University.

Data Availability Statement: Data not available due to cybersecurity reasons.

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

References

1. Satyanarayanan, M.; Simoens, P.; Xiao, Y.; Pillai, P.; Chen, Z.; Ha, K.; Hu, W.; Amos, B. Edge analytics in the internet of things. *IEEE Pervasive Comput.* **2015**, *14*, 24–31. [CrossRef]
2. Hassan, N.; Gillani, S.; Ahmed, E.; Yaqoob, I.; Imran, M. The Role of Edge Computing in Internet of Things. *IEEE Commun. Mag.* **2018**, *56*, 110–115. [CrossRef]
3. Markakis, E.K.; Politis, I.; Lykourgiotis, A.; Rebahi, Y.; Mastorakis, G.; Mavromoustakis, C.X.; Pallis, E. Efficient next generation emergency communications over multi-access edge computing. *IEEE Commun. Mag.* **2017**, *55*, 92–97. [CrossRef]
4. Yu, W.; Liang, F.; He, X.; Hatcher, W.G.; Lu, C.; Lin, J.; Yang, X. A survey on the Edge Computing for the Internet of Things. *IEEE Access* **2018**, *6*, 6900–6919. [CrossRef]
5. Agiwal, M.; Roy, A.; Saxena, N. Next generation 5G wireless networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1617–1655. [CrossRef]
6. Ahmed, A.; Ahmed, E. A survey on mobile edge computing. In Proceedings of the 2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA), Baku, Azerbaijan, 12–14 October 2016; pp. 1–8.
7. Porambage, P.; Okwuibe, J.; Liyanage, M.; Ylianttila, M.; Taleb, T. Survey on Multi-Access Edge Computing for Internet of Things Realization. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2961–2991. [CrossRef]
8. Kumar, A.; Abhishek, K.; Ghalib, M.R.; Shankar, A.; Cheng, X. Intrusion detection and prevention system for an IoT environment. *Digit. Commun. Netw.* **2022**, *8*, 540–551. [CrossRef]
9. Roesch, M. Snort: Lightweight intrusion detection for networks. *Lisa* **1999**, *99*, 229–238.
10. Lakhno, V.; Akhmetov, B.; Korchenko, A. Development of a decision support system based on expert evaluation for the situation center of transport cybersecurity. *J. Theor. Appl. Inf. Technol.* **2018**, *96*, 4530–4540.
11. Zhylin, A.; Hudyncev, M.; Litvinov, M. Functional model of cybersecurity situation center. *Collect. Inf. Technol. Secur.* **2018**, *6*, 51–67. [CrossRef]
12. Chen, S.; Jian, Z.; Huang, Y. Autonomous driving: Cognitive construction and situation understanding. *Sci. China Inf. Sci.* **2019**, *62*, 1–27. [CrossRef]
13. Lewis, L.; Jakobson, G.; Buford, J. Enabling cyber situation awareness, impact assessment, and situation projection. In Proceedings of the MILCOM 2008–2008 IEEE Military Communications Conference, San Diego, CA, USA, 16–19 November 2008; IEEE: Piscataway, NJ, USA, 2008.
14. Edge-IIoT Set Cyber Security Dataset of IoT&IIoT. Available online: https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot (accessed on 20 January 2024).
15. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *TechRxiv* **2022**, *10*, 40281–40306. [CrossRef]
16. Zhu, X.; Deng, H. A security situation awareness approach for IoT software chain based on Markov Game Model. *Int. J. Interact. Multimed. Artif. Intell.* **2022**, *7*, 59–65. [CrossRef]
17. Sarhan, M.; Layeghy, S.; Moustafa, N.; Gallagher, M.; Portmann, M. Feature extraction for machine learning-based intrusion detection in IoT networks. *Digit. Commun. Netw.* **2022**, *10*, 205–216. [CrossRef]
18. Howard, R.A. *Dynamic Programming and Markov Processes*; MIT Press: Cambridge, MA, USA, 1960.
19. Littman, M.L. Markov games as a framework for multi-agent reinforcement learning. In Proceedings of the Eleventh International Conference on International Conference on Machine Learning, ser. ICML'94, San Francisco, CA, USA, 21–27 July 1994; Morgan Kaufmann Publishers Inc.: Cambridge, MA, USA, 1994; pp. 157–163.
20. Phillips, C.; Swiler, L.P. A graph-based system for network-vulnerability analysis. In Proceedings of the 1998 Workshop on New Security Paradigms—NSPW '98, Charlottesville, VA, USA, 22–26 September 1998; ACM Press: New York, NY, USA, 1998; pp. 71–79.
21. Sabur, A.; Chowdhary, A.; Huang, D.; Alshamrani, A. Toward scalable graph-based security analysis for cloud networks. *Comput. Netw.* **2022**, *206*, 108795. [CrossRef]

22. Kachare, G.P.; Choudhary, G.; Shandilya, S.K.; Sihag, V. Sandbox Environment for Real Time Malware Analysis of IoT Devices. In *Communications in Computer and Information Science*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 1604.
23. Skinner, B.F. *Science and Human Behavior*; Simon and Schuster: New York, NY, USA, 1965.
24. Dino, A.S.; Manivannan, D. Intrusion detection based on Machine Learning techniques in computer networks. *Internet Things* **2021**, *16*, 100462. [CrossRef]
25. Masdari, M.; Khezri, H. A survey and taxonomy of the fuzzy signature-based Intrusion Detection Systems. *Appl. Soft Comput.* **2020**, *92*, 106301. [CrossRef]
26. Giamfi, E.; Jurcut, A. A Robust Security Task Offloading in Industrial IoT-Enabled Distributed Multi-Access Edge Computing. *Front. Signal Process* **2022**, *2*, 788943. [CrossRef]
27. Alhajjar, E.; Maxwell, P.; Bastian, N. Adversarial machine learning in Network Intrusion Detection Systems. *Expert Syst. Appl.* **2021**, *186*, 115782. [CrossRef]
28. Ferroq, M.A.; Maglaras, L.; Maschaviannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2020**, *50*, 102419. [CrossRef]
29. Mervem, A.; El Ouahidi, B. Hybrid intrusion detection system using machine learning. *Netw. Secur.* **2020**, *5*, 8–19.
30. Ma, X.; Li, Y.; Gao, Y. Decision model of intrusion response based on Markov game in fog computing environment. *Wireless Netw.* **2023**, *29*, 3383–3392. [CrossRef]
31. Bhatia, J.; Italiya, K.; Jadeja, K.; Kumhar, M.; Chauhan, U.; Tanwar, S.; Bhavsar, M.; Sharma, R.; Manea, D.L.; Verdes, M.; et al. An Overview of Fog Data Analytics for IoT Applications. *Sensors* **2023**, *23*, 199. [CrossRef]
32. Farahani, B.; Firouzi, F.; Chang, V.; Badaroglu, M.; Constant, N.; Mankodiya, K. Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Gener. Comput. Syst.* **2018**, *78*, 659–676. [CrossRef]
33. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. Internet of Things: Security and Solutions Survey. *Sensors* **2022**, *22*, 7433. [CrossRef]
34. Prasad, V.K.; Bhavsar, M.D.; Tanwar, S. Influence of Montoring: Fog and Edge Computing. *Scalable Comput. Pract. Exp.* **2019**, *20*, 365–376. [CrossRef]
35. Sobecki, A.; Szymański, J.; Gil, D.; Mora, H. Deep learning in the fog. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719867072. [CrossRef]
36. Chalapathi, G.; Chamola, V.; Vaish, A.; Buyya, R. Industrial Internet of Things (IIoT) Applications of Edge and Fog Computing: A Review and Future Directions. *arXiv* **2019**, arXiv:1912.00595.
37. Chen, B.; Wan, J.; Celesti, A.; Li, D.; Abbas, H.; Zhang, Q. Edge computing in IoT-based manufacturing. *IEEE Commun. Mag.* **2018**, *56*, 103–109. [CrossRef]
38. Industrial Internet Consortium White Paper: Introduction to Edge Computing in IIoT. Available online: https://www.iiconsortium.org/pdf/Introduction_to_Edge_Computing_in_IIoT_2018-06-18.pdf (accessed on 12 December 2023).
39. Liu, C.; Wang, P.; Xiang, F.; Sun, Z. A review of issues and challenges in fog computing environment. In Proceedings of the 2019 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka, Japan, 5–8 August 2019; IEEE: Piscataway, NJ, USA, 2019.
40. Ometov, A.; Molua, O.L.; Komarov, M.; Nurmi, J. A survey of security in cloud, edge, and fog computing. *Sensors* **2022**, *22*, 927. [CrossRef]
41. Lopes, A.; Hutchison, A. Experimenting with machine learning in automated intrusion response. In *Intelligent Distributed Computing XIII*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 505–514.
42. Aljumah, A.; Ahanger, T.A. Fog Computing and Security Issues: A Review. In Proceedings of the 7th International Conference on Computers Communications and Control (ICCCC), Oradea, Romania, 8–12 May 2018; pp. 237–239.
43. Parikh, S.; Dave, D.; Patel, R.; Doshi, N. Security and Privacy Issues in Cloud, Fog and Edge Computing. *Procedia Comput. Sci.* **2019**, *160*, 734–739. [CrossRef]
44. Xiao, Y.; Jia, Y.; Liu, C.; Cheng, X.; Yu, J.; Lv, W. Edge Computing Security: State of the Art and Challenges. *Proc. IEEE* **2019**, *107*, 1608–1631. [CrossRef]
45. Zhang, J.; Chen, B.; Zhao, Y.; Cheng, X.; Hu, F. Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues. *IEEE Access* **2018**, *6*, 18209–18237. [CrossRef]
46. Alwarafy, A.; Al-Thelaya, K.A.; Abdallah, M.; Schneider, J.; Hamdi, M. A Survey on Security and Privacy Issues in Edge-Computing-Assisted Internet of Things. *IEEE Internet Things J.* **2021**, *8*, 4004–4022. [CrossRef]
47. Almutairy, N.M.; Al-Shqeerat, K.H. A Survey on Security Challenges of Virtualization Technology in Cloud Computing. *Int. J. Comput. Sci. Inf. Technol.* **2019**, *11*, 1–19. [CrossRef]
48. Czeczot, G.; Rojek, I.; Mikołajewski, D. Analysis of Cyber Security Aspects of Data Transmission in Large-Scale Networks Based on the LoRaWAN Protocol Intended for Monitoring Critical Infrastructure Sensors. *Electronics* **2023**, *12*, 2503. [CrossRef]
49. Czeczot, G.; Rojek, I.; Mikołajewski, D.; Sangho, B. AI in IIoT Management of Cybersecurity for Industry 4.0 and Industry 5.0 Purposes. *Electronics* **2023**, *12*, 3800. [CrossRef]
50. Kazim, M.; Zhu, S.Y. Virtualization Security in Cloud Computing. In *Guide to Security Assurance for Cloud Computing*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 51–63.

51. Butun, I.; Österberg, P.; Song, H. Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 616–644. [CrossRef]
52. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
53. Mahato, G.K.; Chakraborty, S.K. Securing edge computing using cryptographic schemes: A review. *Multimed. Tools Appl.* **2023**, 1–24. [CrossRef]
54. Bahrami, B.; Khayyambashi, M.R.; Mirjalili, S. Edge server placement problem in multi-access edge computing environment: Models, techniques, and applications. *Cluster Comput.* **2023**, *26*, 3237–3262. [CrossRef]
55. Ghrab, S.; Lahyani, I.; Yangui, S.; Jmaiel, M. A core IoT ontology for automation support in edge computing. *SOCA* **2023**, *17*, 25–37. [CrossRef]
56. Zhang, J.; Ning, Z.; Cao, H. An intelligent trusted edge data production method for distributed Internet of things. *Neural Comput. Appl.* **2023**, *35*, 21333–21347. [CrossRef]
57. Anoushee, M.; Fartash, M.; Akbari Torkestani, J. An intelligent resource management method in SDN based fog computing using reinforcement learning. *Computing* **2023**, 1–30. [CrossRef]
58. Sha, Y.; Wang, H.; Wang, D.; Ghobaei-Arani, M. A multi-objective QoS-aware IoT service placement mechanism using Teaching Learning-Based Optimization in the fog computing environment. *Neural Comput. Appl.* **2023**, *36*, 3415–3432. [CrossRef]
59. Chouat, H.; Abbassi, I.; Graiet, M.; Sudholt, M. Adaptive configuration of IoT applications in the fog infrastructure. *Computing* **2023**, *105*, 2747–2772. [CrossRef]
60. Faraji, F.; Javadpour, A.; Sangaiah, A.K.; Zavieh, H. A solution for resource allocation through complex systems in fog computing for the internet of things. *Computing* **2023**, 1–25. [CrossRef]
61. Thakur, D.; Saini, J.K.; Srinivasan, S. DeepThink IoT: The Strength of Deep Learning in Internet of Things. *Artif. Intell. Rev.* **2023**, *56*, 14663–14730. [CrossRef]
62. Kaliya, N.; Pawar, D. Unboxing fog security: A review of fog security and authentication mechanisms. *Computing* **2023**, *105*, 2793–2819. [CrossRef]