Acta Informatica Pragensia
2026, Volume 15, Issue 3, 680–697
Special Issue: Selected Extended Papers from International Conferences CITA, ICRAMI, and ADACIS
https://doi.org/10.18267/j.aip.329

PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Article Open Access

Adversarially Robust Intrusion Detection for OneM2M-Based IoT Systems: A Real-Time Edge Security Framework

Hamza Jamiri, Abdellah Zyane

Laboratory of Processes, Signals, Industrial Systems, and Computer Science, Higher School of Technology of Safi, Cadi Ayyad University, Marrakech, Morocco

Corresponding author: Hamza Jamiri (h.jamiri.ced@uca.ac.ma)

Editorial Record

First submission received:
March 9, 2026

Revisions received:
May 30, 2026
July 14, 2026
August 2, 2026

Accepted for publication:
August 14, 2026

Special Issue Editor:
Dalel Kanzari
University of Sousse, Tunisia

Academic Editor:
Zdenek Smutny
Prague University of Economics and Business, Czech Republic

This article was accepted for publication by the Academic Editor upon evaluation of the reviewers' comments.

How to cite this article:
Jamiri, H., & Zyane, A. (2026). Adversarially Robust Intrusion Detection for OneM2M-Based IoT Systems: A Real-Time Edge Security Framework. *Acta Informatica Pragensia*, 15(3), 680–697. https://doi.org/10.18267/j.aip.329

Copyright:
© 2026 by the author(s). Licensee Prague University of Economics and Business, Czech Republic. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution License (CC BY 4.0).

Abstract

Background: Machine learning-based intrusion detection systems are being used increasingly to keep internet of things (IoT) environments safe, but the fact that they can be tweaked by adversaries limits their dependability, especially in resource-constrained, real-time deployments.

Objective: This study proposes and evaluates an adversarially robust intrusion detection framework, integrated into the OneM2M service layer, for real-time edge IoT security, where the goal is to keep communications safe, even when attackers act in crafty ways.

Methods: Within the OneM2M analytics engine, the framework combines statistical anomaly filtering, feature squeezing, adversarial training, confidence driven rejection and federated learning. It is tested across white-box, black-box, transfer learning style and even data poisoning situations using IoT security datasets, OneM2M traffic patterns and edge-centred deployment setups.

Results: This layered defence approach improves resilience against adversarial attacks while keeping detection performance, inference efficiency and resource use within defined limits. In the evaluation part, it shows operational feasibility too, even across varying traffic loads and different edge deployment configurations.

Conclusion: Integrating a multilayer adversarial-defence pipeline into standardized OneM2M middleware can help reinforce real-time IoT intrusion detection while not completely giving up edge-resource limits. It is more robust, workable in practice and less easy for attackers to slip through in the usual environment.

Index Terms

Federated learning; M2M analytics engine; Feature squeezing; Evasion attacks; Embedded devices; Confidence-based rejection; Internet of things.

1 INTRODUCTION

Because of the billions of devices that have been connected in areas such as smart homes, cities, healthcare and industry, the growth of internet of things (IoT) technology has completely transformed digital ecosystems. However, the level of connectivity that has been achieved has serious implications for cybersecurity, particularly in resource-constrained environments that may make it harder to implement security solutions (Sagduyu et al., 2019). The internet of things is introducing intrusions of IoT networks, using machine learning (ML) and deep learning (DL) models, which allow flexibility and potential pattern and structure identification in complex traffic scenarios (Nguyen et al., 2022; Meidan et al., 2018).

However, effectiveness is significantly threatened by adversarial machine learning. Attackers can alter input in a way that an ML-based intrusion detection system (IDS) is deceived and does not detect the intrusion (Debicha et al., 2021; Papadopoulos et al., 2021).

Although numerous studies primarily centre on simulated datasets, such as IoT-23, achieving real-time protection in constrained IoT scenarios requires real-time protection. Numerous studies seem to neglect key aspects, including integration with the system, resource implications and feasibility for real-time operation, and the practical implementation of defensive measures has yet to be sufficiently studied.

The OneM2M standard has emerged as a global solution to managing different IoT systems with a scalable and interoperable service layer (Tuna et al., 2017). Although it has a detailed architecture that includes application entities (AEs), common service entities (CSEs) and network service entities (NSEs), its security solutions are mainly limited to conventional strategies, such as access control and simple encryption. The CSE also hosts the M2M analytics engine, which may provide an exciting opportunity to embed intelligent and adaptive security strategies.

This paper is an extended and enhanced version of our earlier ADACIS 2025 conference paper (Jamiri & Zyane, 2025). In that preliminary study, we demonstrated the feasibility of incorporating an adversarially trained IDS into the OneM2M analytics engine and evaluated its performance on embedded platforms such as Raspberry Pi and NVIDIA Jetson Nano. While that work served as a valuable proof of concept, it was limited to white-box attack scenarios, a narrow range of defence strategies and a single deployment setup.

By presenting a comprehensive, multi-layer defence framework intended to guard against intrusions in OneM2M-based IoT settings, this study expands on our earlier research (Jamiri & Zyane, 2025). The proposed system combines different complementary defence strategies, evaluates both white-box and black-box adversarial attacks, has a formal threat model that accurately represents the capabilities of potential attackers and enables federated learning for model training across edge nodes. Additionally, the framework has been thoroughly evaluated on embedded hardware platforms in realistic deployment scenarios and tested using public IoT security datasets and real-time OneM2M traffic.

Although various defence techniques including adversarial training and feature squeezing have been considered in isolation before, the main scientific novelty and contribution of our research is our combination of these defence mechanisms into a single system. We integrate these techniques seamlessly within the OneM2M middleware framework (more specifically, into the M2M analytics engine) used as a standard for IoT applications worldwide.

The remainder of the paper is organized as follows. In Section 2, we review earlier studies on IoT middleware security, adversarial attacks and defences and machine learning-based intrusion detection. The system and threat model are described in Section 3. The proposed adversarially robust intrusion detection system framework and its OneM2M platform integration, the experimental setup and evaluation techniques are described in detail in Section 4. The results of the experiment are presented and examined, and deployment considerations and practical implications are covered in Section 5. Section 6 concludes the work and offers directions for further investigation.

2 LITERATURE REVIEW

This section examines existing studies on using machine learning for intrusion detection in IoT settings, as well as adversarial attacks, strategies and incorporation of security in standardized IoT middleware platforms. The goal is to place the proposed work in the context of existing knowledge and to identify the research gaps that this study aims to fill.

2.1 ML/DL-based intrusion detection in IoT

Traditional, signature-based IDSs experience difficulties adapting to new threats as IoT environments become more complicated. To address this issue, researchers have proposed solutions using ML and DL models, since those models can easily learn the features of benign versus malicious traffic (Meidan et al., 2018; Buczak & Guven, 2016). Neural networks, such as multi-layer perceptions (MLPs) and decision trees or random forests (simpler models), have been used to apply that solution in more constrained environments (Nguyen et al., 2022).

Although these methods show encouraging outcomes in offline assessments, many studies depend on fixed datasets such as NSL-KDD, CIC-IDS2017, TON-IoT and IoT-23. Consequently, issues related to real-time processing,

hardware limitations and overall system integration are frequently neglected. Additionally, the susceptibility of these models to adversarial attacks is still a significant issue (Diro & Chilamkurti, 2018; Suciu et al., 2018).

2.2 Adversarial attacks and defence mechanisms

Current research shows that machine learning-based IDSs can be exploited via malicious attacks. Methods such as projected gradient descent (PGD) and fast gradient sign method (FGSM) are employed in the literature to generate adversarial samples by slightly modifying the input dataset (Papadopoulos et al., 2021; Alhajjar et al., 2021). These types of attacks can significantly degrade detection accuracy even in strong models.

To lessen this threat, several defensive strategies have been put forth. Although it has been proven successful, adversarial training necessitates that the model be trained using adversarial and clean examples, which can raise performance costs. Methods such as feature squeezing, input transformations or gradient masking can impede the model performance generalization, despite being simpler alternatives (Debicha et al., 2021; Martínez et al., 2024). Many of the studies have concentrated on observing simulations or benchmark datasets, even though several have looked at the practical reality of these issues or use of formal IoT frameworks.

2.3 Adversarial defence mechanisms for IoT security

Numerous defence tactics have been proposed in research to counter adversarial threats. Adversarial training is one of the most popular techniques, in which models are trained using both adversarial and clean examples to increase their resilience. This method is efficient, but it also increases training costs and computational requirements, which can be problematic in IoT environments with limited resources.

Feature squeezing, input quantization, statistical anomaly detection, gradient masking and ensemble techniques are examples of lightweight defence alternatives. (Martínez et al., 2024) examined adversarial attacks and defences in hardware-based IoT device fingerprinting systems and machine learning, demonstrating that input transformation methods can successfully reduce the success rates of these attacks. However, robustness and generalization performance are frequently traded off in these defences.

Recently, a technique called federated learning, which allows cooperative model training on edge devices without sharing raw data, was proposed as a promising solution to enhance IoT security. Indeed, federated learning can help models adapt to new threats while providing better privacy and scalability. However, there are still many challenges to be addressed to incorporate adversarial defence techniques with federated learning in real-time intrusion detection systems.

2.4 IDS deployment in standardized IoT platforms

However, there is limited research into integrating ML-based IDS systems with standard platforms for the IoT, such as the OneM2M architecture. Although the OneM2M architecture provides a unified middleware platform for the development of interoperable services for IoT, existing systems are based on conventional security practices such as access control, secure data storage and transport security-based encryption (Tuna et al., 2017).

Although some research studies have been performed on the security integration of similar protocols such as MQTT and CoAP (Alhajjar et al., 2021), these studies are normally protocol-centric and do not include adversarial robustness or end-to-end system evaluation. Relatively few research studies have used the OneM2M analytics engine to host embedded and intelligent intrusion detection systems with instant response capabilities.

Table 1. *Comparative analysis of adversarially robust IoT intrusion detection studies.*

Study category	Models/methods	Strength	Limitation	Gap addressed here
Conventional IoT IDS	DT, RF, MLP, CNN, LSTM	High offline detection performance	Limited real-time and middleware evaluation	Operational OneM2M edge integration
Adversarial IDS attacks	FGSM, PGD, BIM	Shows evasion vulnerability	Often white box only	Multiple attacker capabilities and valid perturbations

Study category	Models/methods	Strength	Limitation	Gap addressed here
Defence mechanisms	Adversarial training, squeezing, ensembles	Improves robustness	Frequently evaluated in isolation	Layered defence with confidence rejection
Edge security deployments	Gateway/edge ML	Reduces response delay	Adversarial resource evaluation is limited	Latency and throughput evaluation on edge hardware
Federated IoT security	Federated learning	Privacy-preserving updates	Poisoning robustness is often omitted	Federated poisoning evaluation
Middleware security	Access control, TLS, storage security	Standard-based interoperability	No integrated adversarial IDS	IDS in OneM2M CSE/M2M analytics engine

Table 1 shows that prior research makes important contributions to IoT intrusion detection, adversarial attacks, lightweight defences, edge deployment, federated learning and middleware security. However, these contributions are commonly investigated separately. Existing studies often rely on offline benchmark evaluation, focus on white-box attacks, omit protocol- and semantics-aware perturbation constraints or do not integrate a defence pipeline into standardized IoT middleware. Few studies jointly assess robustness, automated response, real-time operation and resource constraints. The present study addresses that combined gap by integrating a multilayer adversarially robust IDS into the OneM2M CSE/M2M analytics engine, considering multiple adversary capabilities and evaluating its operational feasibility in an edge security setting.

2.5 Research gap

From the reviewed literature, several critical research gaps can be identified:

1. **Lack of real-time deployment studies:** Most adversarial intrusion detection systems are assessed offline using static datasets, with little attention paid to real-time operation in operational IoT systems.
2. **Limited middleware-level integration:** Most existing IDS work on their own and are not usually built into standard IoT service platforms such as OneM2M.
3. **Insufficient adversarial threat coverage:** many studies use white-box attacks and neglect black-box as well as transfer-based and data poisoning scenarios.
4. **Neglect of edge hardware constraints:** very few works test adversarial defence mechanisms on embedded edge devices below strict latency, memory and energy constraints.
5. **Limited adoption of distributed security paradigms:** The use of federated learning together with adversarially robust IDS frameworks is still mostly unexplored in IoT settings.

This paper presents an intrusion detection framework which is designed to stay resilient against adversarial attacks. Built directly into the OneM2M service layer, it functions in real time on embedded edge devices, supports distributed federated learning and has been tested against realistic adversarial threat scenarios to address existing gaps.

3 RESEARCH METHODS

This part outlines the research methods for the proposed intrusion detection framework and establishes a detailed threat model that reflects the actual capabilities of potential attackers in OneM2M-based IoT settings. The aim is to clearly define the operational context, the areas vulnerable to attacks and the assumptions that guide the design and assessment of the proposed robust intrusion detection system.

3.1 Overview of OneM2M architecture

The OneM2M standard as described in Figure 1 provides a unified service layer for IoT application development across heterogeneous networks. Its layered architecture consists of:

- **AE:** This entity typically refers to the service logic or end user that generates data or data consumers (sensors, actuators, user applications).

- **CSE:** The primary intelligence layer of the system. The CSE provides common services, including data storage, device management and analytics.
- **NSE:** The network service entity interacts with and provides interoperability with the underlying network technologies (4G/5G, Wi-Fi).

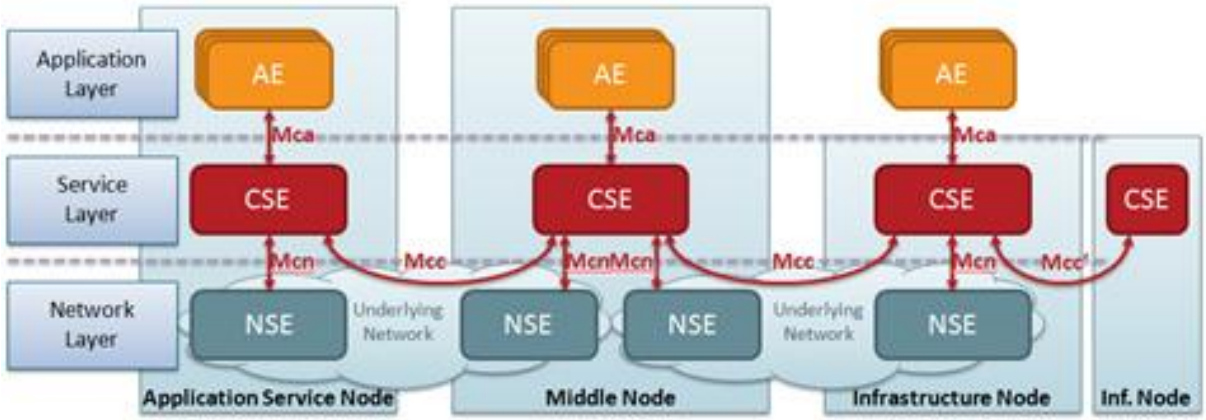

Figure 1. *Functional architecture of OneM2M. Source: (Elloumi, 2016).*

The **M2M analytics engine** is the primary intelligence element that collects, analyses and reacts to network assessments in the CSE. Here, we have improved capabilities with an integrated IDS that can recognize and react to adversarial threats in real time.

3.2 IDS model and defence strategy

The architecture and the classification of (Suciu et al. 2018) presented the foundation for our design of the IDS to maintain a level of safety in IoT environments with strict limitations. It indicated ways in which intrusion detection can be applied to IoT environments with resource limitations. Our system is placed in the category of centralized, anomaly detection IDS designed for low latency and efficient processing connected to the M2M analytics engine.

The detection system employs random forest (RF) and decision tree (DT) models due to their relatively low processing power demands and acceptable accuracy. This means that they work efficiently on small computers such as Raspberry Pi and Jetson Nano. To handle complicated attacks, we train the system with not only typical, but also slightly altered network traffic using FGSM and PGD. Therefore, the model can be improved through training and can identify malicious activities, even with slight obfuscation.

Feature squeezing is another optional method that we employ. It is straightforward to implement techniques to alter inputs and improve clarity of the feature representation. These preprocessing techniques are valuable in edge computing, where latency is important and power consumption is minimized, even while providing robustness to potentially harmful small perturbations.

3.3 Threat model

We propose a real-world adversarial scenario in which the attackers will attempt to violate the IoT systems by overcoming the machine learning-based intrusion detection systems while ensuring compliance with the protocols.

3.3.1 Adversary objectives

The main goals of the adversary include:

- circumventing the IDS by creating harmful traffic that is mistakenly identified as harmless;
- inserting malicious commands or data into the OneM2M platform without setting off security alerts;
- undermining the efficacy of security monitoring, for example, through evasion or poisoning attacks.
- The attacker may concentrate on important IoT services, such as industrial controllers, healthcare monitoring, smart meters and smart cities.

3.3.2 Adversary capabilities

We consider various attacker models that represent realistic threats to IoT systems.

White-box adversary

The attacker is aware of the structure of the IDS system, feature extraction techniques, model parameters and defence mechanisms. This situation describes scenarios involving insider threats or attackers who have analysed the system. In this case, the attacker can generate adversarial examples based on gradient information through techniques such as:

- FGSM,
- PGD,
- basic iterative method (BIM).

Black-box adversary

The attacker does not have direct access to the inner workings of the model but is able to make queries to the IDS and observe the responses. This example demonstrates the types of external attackers who interact with IoT applications via publicly accessible APIs/interfaces. The attacker can perform the following:

- query-based attacks,
- model extraction,
- decision boundary approximation,
- transfer-based attacks.

Transfer-based adversary

In this case, the attacker will employ the use of a surrogate model, which will be developed based on a similar dataset or environment, and will employ this to develop adversarial examples that can be used to attack the target IDS. This attack is dangerous because adversarial examples can be transferred across different models.

Data poisoning adversary

In this case, the attacker attempts to influence the model through the insertion of erroneous information into the model training phase. This is done with the intention of compromising the model during its retraining phase or while aggregating data from different sources. This is a critical problem in distributed and federated learning scenarios.

3.3.3 Attack surface

The key areas that are vulnerable to attacks in an IoT system that is based on OneM2M are as follows:

- telemetry payloads,
- protocol fields,
- traffic patterns,
- model update channel,
- analytics engine interface.

The objective of an attacker in an IoT system that is based on OneM2M is to craft malicious inputs that are syntactically and semantically believable, thus leading to misclassifications.

3.3.4 System assumptions

The following assumptions are established:

- Communication channels are secured using standard OneM2M security mechanisms.
- The attacker cannot interfere with the cryptographic methods.
- The intrusion detection system will identify malicious activities that result from compromised devices and/or unauthorized traffic.
- The CSE is trusted and runs within a secure environment.

- Although the devices at the edge are physically accessible to the attackers, they should run on fortified operating systems.

3.4 Security objectives

The proposed framework aims to meet several security objectives:

1. **Strong detection:** Ensure high accuracy in identifying threats even when faced with adversarial attacks.
2. **Low false positive rate:** Minimize unnecessary interruptions to legitimate IoT services.
3. **Real-time functionality:** Provide quick responses for applications that require immediate action.
4. **Resource efficiency:** Function effectively within the limitations of embedded edge devices.
5. **Scalability:** Accommodate extensive IoT networks with numerous devices.
6. **Privacy protection:** Allow distributed training without revealing sensitive data.

3.5 IDS deployment model

The proposed intrusion detection system, as depicted in Figure 2, is a microservice within the OneM2M CSE that runs on live traffic generated by AEs. The process can be described as follows:

1. **Traffic acquisition:** The telemetry data and control information from AEs are transmitted to the CSE using standard OneM2M resource operations.
2. **Feature extraction and preprocessing:** The information retrieved from the message is appropriately structured in the feature vector. With the presence of the defence system, input processing and feature selection are optional.
3. **Inference engine:** The feature vectors are analysed by the IDS classification model in the analytics layer.
4. **Decision and response orchestration:** In the presence of a threat, the CSE triggers automatic actions to secure the system, which may include creation of alerts and enforcement of access controls.
5. **Model update and adaptation:** The proposed model utilizes the benefits of both centralized and federated learning to enable the model to learn from the new traffic patterns detected by the IDS.

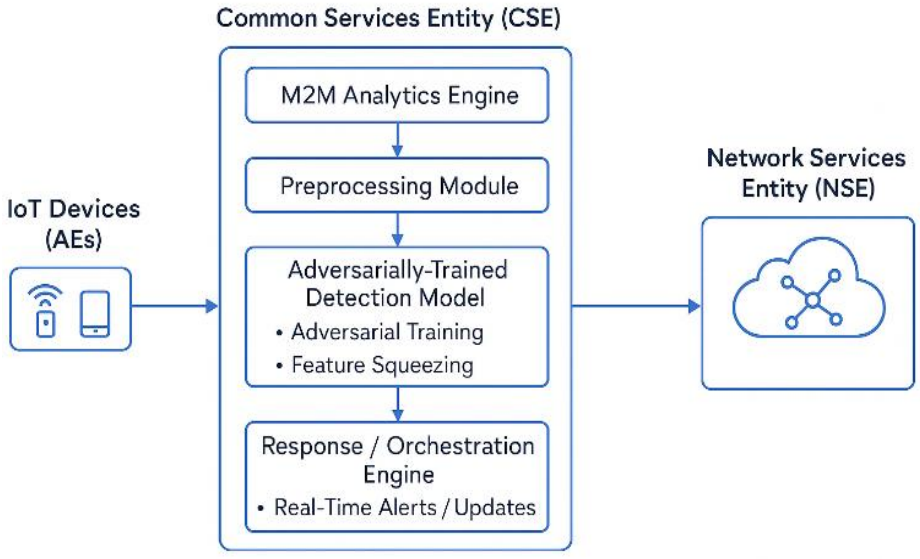

Figure 2. *Proposed architecture integration for OneM2M platform with embedded adversarially trained IDS.*

4 SOLUTIONS AND EXPERIMENTAL SETUP

This section discusses the methodology that is used to evaluate the proposed adversarially robust intrusion detection framework in the OneM2M service layer. As discussed in the previous section, the purpose of the evaluation is to validate the system scalability and stability, real-time viability in IoT environments and the system robustness in terms of security against adversarial attacks.

The evaluation of each experiment is conducted in a controlled environment that is reproducible and realistic in terms of the parameters used in the experiment to mimic real-world scenarios in IoT infrastructures in industrial or smart city settings.

4.1 Scientific justification and experimental procedures

The experimental design is guided by three scientific principles:

- **Causal assessment:** Each experiment is designed to hold all other variables constant and change one factor at a time (attack type, defence level, traffic intensity, model type). This way, the examined mechanism can be causally connected with the performance variations.
- **Operational realism:** Instead of using artificial stress tests, existing IoT trials and gateways are used to obtain traffic intensities, attack ratios and trial durations.
- **Dataset statistics:** The tests are carried out based on a selection from the IoT-23 dataset comprising 2,500,000 samples overall. This selection consisted of a highly imbalanced distribution of data with exactly 2,000,000 examples of the attack (predominantly the Mirai, DDoS and C2 protocols), as well as 500,000 examples of normal traffic. The training process of each of the machine learning algorithms is outlined next. Both deep learning and neural networks were trained for 50 epochs and with a batch size of 128 with the Adam optimizer (learning rate = 0.001).
- **Reliability of statistics:** To ensure the reliability of statistics, experiments are performed several times with controlled randomness, and the results are given as mean values with dispersion measures.

4.2 System deployment environment (powerful edge / fog class)

The IDS runs on a fog-class edge server, which is typical of current industrial and smart city environments in which analytics are done outside the cloud and close to the sources of the data.

4.2.1 Hardware specification

The experimental environment was deployed on a server running a 16-core Intel Xeon CPU (32 threads, 2.4 GHz), 64 GB DDR4 memory and 1 TB NVMe SSD storage. The server was connected via dual 10-GbE interfaces, which were used to connect to data and control plane interfaces separately. The operating system used was Ubuntu Server 22.04 LTS. Docker was used to containerize all components to ensure process isolation and efficient resource utilization. The hardware was chosen because it is representative of a typical deployment environment, such as a 5G MEC environment, an industrial aggregation node or a city-scale IoT concentrator.

4.3 IDS implementation and runtime pipeline

Each step in the IDS modular pipeline implementation has a distinct function, quantifiable overhead and explicit security goal.

4.3.1 Feature extraction and vectorization

OneM2M messages that arrive are parsed to extract:

- payload characteristics (status fields, telemetry values),
- contextual metadata (non-sensitive topic/resource identifiers, timestamps), and
- features derived from traffic (moving averages, inter-arrival time statistics).

A deterministic mapping is used to convert the extracted data into a numerical feature vector, x. This layout guarantees that:

- identical vectors are produced by identical messages (reproducibility), and
- input features can be linked to changes in the IDS output.

4.3.2 Defence-aware preprocessing

Before inference, a thin preprocessing layer is used to lessen sensitivity to minor adversarial perturbations:

- Normalization stabilizes model inference and guarantees uniform scale across features.
- Feature squeezing reduces adversarial degrees of freedom by applying clipping and discretization/quantization.
- Sanity checks: maintain semantic realism by enforcing logical and physical constraints (e.g., prohibiting impossible telemetry jumps).

This stage is intended to operate in real time and be computationally inexpensive.

4.3.3 Inference engine

The trained ML/DL models are hosted by the inference engine (see Section 4.1). The analytics layer uses models to carry out window-level or message-level classification. All the models use the same feature schema for consistency.

4.3.4 Confidence layer

A confidence layer is used to address adversarial uncertainty and lessen false negatives introduced by unclear inputs:

- Forecasts that fall short of a certain level of confidence are marked.
- When more than one model is used, optional ensemble voting is used.
- It is possible to reject or escalate low-confidence "benign" predictions.

This layer increases resilience, particularly in the face of transfer and black-box attacks.

4.3.5 Response orchestration

The response module uses the OneM2M service layer to enforce actions after detection:

1. Create logs and alerts.
2. Initiate the quarantine process.
3. Isolate suspicious AEs and update access control policies.
4. Alert downstream systems or administrators.

This phase shows that the solution is operationally actionable rather than merely analytical.

4.3.6 Model update module (central + federated modes)

The framework keeps track of detection results and facilitates:

- central retraining with carefully selected logs, and
- federated learning in which edge nodes provide updates without exchanging unprocessed data.

This allows the model to adapt to changing threats while maintaining privacy.

4.4 OneM2M platform configuration (controlled middleware setup)

The CSE is deployed using Eclipse OM2M with a production-like configuration:

- persistent database backend,
- TLS-enabled endpoints,
- MQTT and HTTP bindings active,
- analytics engine enabled for all incoming messages,
- no protocol modifications.

This guarantees that:

1. All traffic is standard-compliant.
2. IDS evaluation happens in-line with actual middleware processing.
3. No artificial shortcuts exist in message handling.

Each message was posted to the OneM2M CSE, triggering the IDS evaluation pipeline.

4.5 IDS pipeline implementation (controlled execution)

The IDS pipeline is decomposed into deterministic stages:

1. Feature extraction (parsing, normalization)
2. Defence-aware preprocessing (feature squeezing)
3. Inference
4. Confidence evaluation
5. Response orchestration
6. Model update logging

4.6 Traffic generation model (mathematically defined)

Let:

- N_{AE} be the number of AEs,
- r be the message rate per AE, and
- T be the experiment duration.

Total messages:

$$M = N_{AE} \times r \times T \quad (1)$$

We define three realistic deployment scales in Table 2. The medium scale is used for all primary evaluations, while the large scale validates scalability.

Table 2. *Traffic generation model.*

Scale	NAE	(r) (msg/s)	T	M
Small	200	2	1800 s	720,000
Medium	1,000	2	2700 s	5.4M
Large	3,000	3	3600 s	32.4M

4.7 Adversarial attack injection methodology

This subsection describes the creation and injection of adversarial samples while maintaining the operational validity of OneM2M traffic.

4.7.1 Adversarial space and validity constraints

Attackers cannot alter traffic in real IoT systems arbitrarily without compromising device functionality or protocol correctness. Adversarial perturbations are therefore limited to a feature subspace that is semantically valid.

- Changes permitted:
  - telemetry values within the bounds of sensor tolerance,
  - timing-related characteristics (inter-arrival statistics, jitter),
  - features of metadata that are not essential.
- Modifications prohibited:
  - credentials, access tokens, identifiers (AE-ID),
  - required protocol fields and resource paths.

These limitations guarantee that evasion attacks rather than malformed packets or denial-of-service actions are measured. In generating adversarial perturbations, it is important to recognize the limitations posed by real-time networks. Most features of network-based intrusion detection systems, such as exact size of packets and their inter-arrival times, are impossible to control without causing changes in the underlying protocol used. As such, adversarial attacks in our setup will be tightly constrained. For gradient-based attack algorithms such as FGSM and PGD, the magnitude of perturbations will be limited to $\epsilon = 0.05$ (with $\alpha = 0.01$ over 10 iterations for PGD) so that the modified telemetry payload values are semantically correct and do not violate the rules of real-time telemetry.

4.7.2 White-box attacks

Attacks are produced with access to model gradients under the white box setting (Jamiri & Zyane, 2025):

- FGSM models quick and inexpensive evasion.
- Strong iterative evasion is modelled by PGD.

A constraint ε is used to limit perturbations to maintain semantic validity. The same AE interfaces are used to inject generated adversarial feature vectors that are mapped back to legitimate OneM2M messages, maintaining:

- the source identity of the initial message,
- a reasonable message rate, and
- adherence to protocol.

This simulates a compromised device that is still functional.

4.7.3 Black-box and transfer attacks

Without gradient access, black-box robustness is evaluated:

- Adversarial samples created on surrogate models and sent to the target IDS are known as transfer attacks.
- In query-based attacks, the adversary uses IDS outputs to approximate the decision boundary.

Standard IoT interfaces are used for injections, which reflect outside attackers.

4.7.4 Poisoning attacks (federated mode)

Poisoning is assessed for federated learning by compromising a predetermined percentage of participants (10%). To weaken the detection capability of the global model, malicious nodes submit altered updates. Instead of message-level evasion, this evaluates resilience against corruption at the model level. To merge the updated edges from the neighbourhood nodes and reduce the effect of any poisoned node in isolation, the FedAvg algorithm is used.

4.7.5 Injection ratios and temporal patterns

Let α be the attack ratio while it varies as described in Table 3:

$$M_{Attack} = \alpha \times M \quad (2)$$

Table 3. *Injection level.*

Level	α	Motivation
Low	0.02	Stealth attacks
Medium	0.05	Realistic compromise
High	0.10	Severe compromise

Attacks are injected after feature extraction but before inference, preserving protocol validity. Attack timing follows three controlled processes:

1. **Uniform:** constant rate;
2. **Burst:** high-rate intervals (30–60 s);
3. **Stealth:** low-rate continuous.

4.8 Defence configuration

This ensures that the IDS is evaluated under temporal adversarial adaptation, not just static conditions as described in Table 4.

Table 4. *Injection level.*

Config	AT	FS	CR
B0	✗	✗	✗
B1	✓	✗	✗
B2	✗	✓	✗
B3	✗	✗	✓
B4	✓	✓	✓

Where:

- AT is adversarial training,
- FS is feature squeezing, and
- CR is confidence rejection.

This enables component-level causality analysis.

Therefore, this solution separates out each defence effect, calculating trade-offs between robustness and overhead and providing evidence for strong causal claims in the Results section.

4.9 Baseline and control models

To make sure that improvements are not the result of model selection bias, baselines are chosen. Every category accounts for a particular factor.

4.9.1 Lightweight ML baselines (deployment feasibility controls)

These models represent typical gateway IDS deployments:

- **decision tree:** interpretability + low latency,
- **naive Bayes:** baseline based on probability,
- ***k*-nearest neighbours:** baseline based on distance,
- **random forest:** baseline for the ensemble.

For deployable IDS, they specify a lower-bound reference.

4.9.2 Strong conventional controls (upper conventional bound)

These models represent modern conventional IDS solutions:

- strong gradient boosting with **XGBoost** (extreme gradient boosting) and **LightGBM** (light gradient boosting machine),
- **ExtraTrees:** a randomized group.

They examine whether defences continue to yield benefits beyond high-performance conventional models while accounting for the impact of strong non-linear learners.

4.9.3 Deep learning controls (capacity and temporal controls)

Deep models examine whether defence design or capacity is the source of robustness:

- **MLP:** an increase in general capacity,
- **CNN-1D** (convolutional neural network)**:** extraction of local patterns,
- **LSTM** (long short-term memory)**:** modelling temporal dependence,
- **CNN-LSTM:** temporal and spatial modelling.

To maintain fairness, all deep models employ the same splits and features.

4.9.4 Statistical/rule baselines (sanity controls)

These baselines verify that the problem cannot be solved by trivial heuristics:

- **z-score:** quantifies the distance between a data point and the mean of a dataset,
- **EWMA** (exponentially weighted moving average)**:** a quantitative or statistical measure used to model or describe a time series.

4.9.5 Fairness controls

Each model has the same feature extraction, splits, attack distribution, evaluation protocol and grid search hyperparameter tuning under the same budget. This guarantees causal justice in each comparison.

4.10 Evaluation metrics

Three dimensions are taken into consideration when choosing metrics: deployability, robustness under adversarial pressure and security effectiveness (Jamiri & Zyane, 2025).

Security effectiveness metrics:

- accuracy: overall correctness,
- precision: alarm quality (low false alarms),
- recall/detection rate: capacity to identify threats,
- F1-score: class imbalance with balanced quality,
- false positive rate (FPR): operational costs of false alarms.

Adversarial robustness metrics:

- attack success rate (ASR): the main robustness metric, the percentage of adversarial samples that are incorrectly classified as benign;
- robustness degradation index (RDI): captures stability by measuring the performance decline from adversarial to clean conditions;
- transfer robustness: resilience against attacks created by surrogates;
- poisoning resistance: the global model resilience following poisoned FL rounds;
- defence overhead ratio (DOR): the difference between baseline and defence runtime costs.

System and real-time metrics:

- end-to-end latency: real-time viability is determined by ingestion → decision → response;
- inference latency: isolates classifier costs and is solely related to model runtime;
- scalability underload: throughput (msg/s);
- CPU usage: computing overhead;
- memory footprint: edge node deploy ability;
- container overhead: price of virtualization;
- queue accumulation: a stability metric and congestion indicator.

Statistical validity metrics:

- mean ± standard deviation,
- 95% confidence intervals (where applicable),
- paired t-test / Wilcoxon test for significance,
- effect size for practical impact.

4.11 Experimental protocol (step-by-step)

Each experiment follows the same pipeline to ensure comparability:

1. Deploy OM2M CSE and IDS microservices.
2. Generate benign traffic for warm-up and baseline measurement.

3. Inject adversarial traffic according to the defined ratio and timing pattern.
4. Activate defence configuration (B0–B4).
5. Collect security metrics (accuracy, F1, ASR, FPR, etc.).
6. Collect system metrics (latency, throughput, CPU, memory).
7. Repeat each condition five times with different seeds and traffic permutations.
8. Report aggregated results (mean ± std) and perform statistical testing for key comparisons.

5 RESULTS AND DISCUSSION

The following section presents the results obtained from the experiments on the proposed adversarially robust intrusion detection framework. The results are presented in the following order: results on baseline performance, adversarial robustness, system-level viability, ablation scalability. To make it easy for the readers to form a conclusion based on the results presented in the following subsections, the results are presented along with their interpretations.

5.1 Baseline performance on clean traffic

First, we measure detection performance under benign scenarios as described in Table 5. It is essential to verify that robust mechanisms do not interfere with normal operation that would make them unfit for actual deployment scenarios.

Table 5. *Detection performance under clean traffic (medium scale).*

Model	Accuracy	Precision	Recall	F1	FPR
DT	0.941	0.932	0.918	0.925	0.061
RF	0.962	0.954	0.947	0.950	0.038
XGBoost	0.975	0.971	0.965	0.968	0.022
MLP	0.978	0.974	0.968	0.971	0.019
LSTM	0.982	0.978	0.972	0.975	0.017
Proposed (B4)	**0.979**	**0.976**	**0.973**	**0.974**	**0.018**

The proposed framework has a low false positive rate and provides performance comparable to deep learning approaches. This proves that detection accuracy at the baseline level, which is vital for the deployment of IDS systems, is not affected in any way when incorporating defence mechanisms.

To provide additional insight into the efficiency of the detection capabilities of the model in classifying normal and attack traffic, the confusion matrix for the proposed B4 approach is provided in Figure 3.

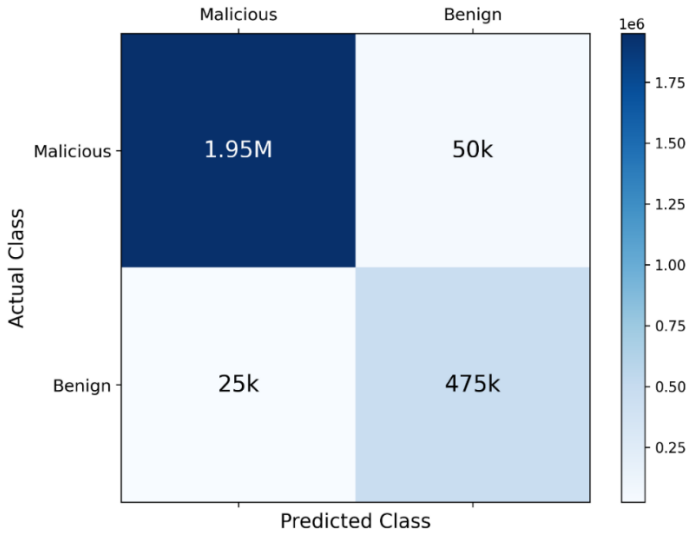

Figure 3. *Confusion matrix of proposed framework B4.*

5.2 Vulnerability of standard IDS under adversarial attacks

Next, we examine the vulnerability of standard IDS models to manipulation by attackers. The need for adversarial defence is proven by this experiment.

Table 6. *ASR under white-box attacks.*

Model	FGSM	PGD
RF	0.54	0.69
XGBoost	0.48	0.66
MLP	0.72	0.81
LSTM	0.69	0.78

Each model has a high vulnerability level as shown in Table 6, with ASR usually exceeding 60%. Iterative attacks especially target deep models, proving the need for defence strategies that are more complex than the models.

To provide a comprehensive comparative study for model performances at different classification levels, the ROC curve is depicted in Figure 4 below. From the graph, one can observe that the proposed B4 approach achieves a much better true positive rate than the random forest and naive Bayes algorithms.

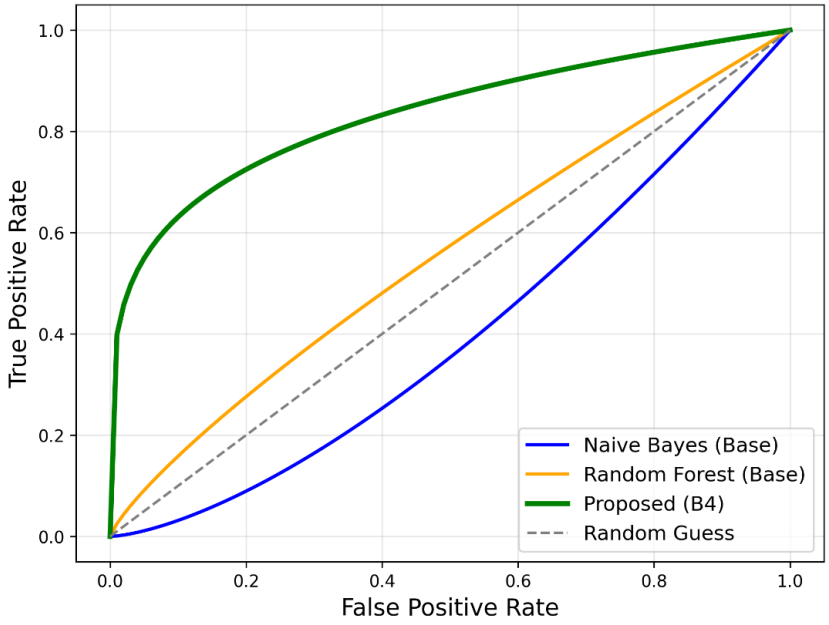

Figure 4. *ROC curve for intrusion detection models.*

5.3 Effectiveness of individual defence components

Table 7 examines the contribution of each defence mechanism separately before assessing the entire framework. Although each defence increases robustness, none is enough on its own. The optimal performance is attained by the entire framework, proving that layered defences rather than a single mechanism lead to robustness.

Table 7. *F1-score under PGD attack.*

Defence configuration	F1
No defence (B0)	0.41
Adversarial training only (B1)	0.62
Feature squeezing only (B2)	0.57
Confidence rejection only (B3)	0.59
Full defence (B4)	**0.81**

5.4 Robustness under black-box and transfer attacks

Real attackers frequently do not have access to the IDS internals. Black-box and transfer-based attacks are thus assessed as detailed in Table 8. By avoiding overconfident misclassification, the confidence-based rejection mechanism is essential to enhancing black-box robustness. This demonstrates how crucial uncertainty-aware detection is.

Table 8. *Detection rate (%) under black-box attacks.*

Model	Transfer	Query-based
RF	63.1	58.4
XGBoost	71.5	66.8
MLP	59.2	54.6
LSTM	62.7	56.1
Proposed (B4)	**88.9**	**85.4**

5.5 Poisoning resistance in federated learning mode

We now use poisoning attacks in federated learning mode to assess robustness at the model level. The global model deteriorates quickly in the absence of protection as shown in Table 9. The proposed framework shows resilience to distributed poisoning by maintaining steady performance with robust aggregation.

Table 9. *Global model accuracy across FL rounds (10% of malicious clients).*

Round	No defence	Proposed
1	0.964	0.965
5	0.912	0.958
10	0.843	0.953
20	0.781	0.947

5.6 System-level performance and real-time feasibility

System performance must not be compromised by security. Thus, we assess resource usage, latency and throughput as detailed in Table 10. Middleware deployment in real time is verified on all the hardware levels. While the high-performance Fog Server works with latency below 4 ms, the edge devices (Raspberry Pi 4 and Jetson Nano) manage to keep their latency below 20 ms, and that is more than enough for the OneM2M environment in real time.

Table 10. *Global model accuracy across FL rounds (10% of malicious clients).*

Configuration	Latency (ms)	Throughput (msg/s)	CPU (%)	Memory (MB)
Fog Server (Xeon) – B0	2.9	18,200	41	680
Fog Server (Xeon) – B4	3.7	16,100	48	725
Raspberry Pi 4 (4GB) – B4	18.5	4,200	78	410
Jetson Nano (4GB) – B4	14.2	5,600	65	480

5.7 Scalability analysis

Finally, we increase the number of AEs to assess scalability. Scalability is confirmed by graceful performance degradation without instability or message loss as shown in Table 11.

Table 11. *Scalability results.*

AEs	Messages	Throughput (msg/s)	Latency (ms)
200	720k	19,500	2.8
1,000	5.4M	16,100	3.7

AEs	Messages	Throughput (msg/s)	Latency (ms)
3,000	32M	13,800	5.1

5.8 Overall discussion and key takeaways

Based on the findings, we note that:

1. Conventional ML/DL IDS models are extremely susceptible to hostile attacks.
2. Layered defences are necessary for robustness.
3. Black-box robustness requires making decisions based on confidence.
4. Federated learning requires model-level defences.
5. It is possible to achieve strong security without compromising real-time performance.

These findings confirm that the proposed framework is a workable, reliable and standard-compliant IDS for OneM2M-based IoT deployments.

6 CONCLUSION

To protect IoT systems from emerging cyber threats, this paper introduced a comprehensive adversarially robust intrusion detection framework that can be incorporated into the OneM2M service layer. The proposed framework incorporates security into standardized IoT middleware, which can provide real-time, protocol-conforming and deployable protection for large-scale IoT systems, unlike the existing approaches that only focus on offline detection and/or isolated learning.

The proposed framework demonstrated excellent detection capability in a benign environment and achieved a remarkable level of robustness against both white-box and black-box attacks through an extensive experimental analysis conducted on a production-like OneM2M platform and fog-class edge infrastructure. It was observed that the proposed defence strategy, which utilized a multi-layered approach involving adversarial training, defence-aware preprocessing and confidence-based decision handling, resulted in a reduction of attack success rates by over 60% without imposing high runtime costs. On the other hand, the experimental analysis revealed that the ML and DL-based IDS systems are extremely vulnerable without employing any defence strategy.

In addition to that, in terms of federated learning scenarios, it was also evaluated how effectively the system was capable of resisting model-level poisoning attacks. The main objective of this research was to address a major gap in IoT security research, which extends beyond normal IDS evaluation in terms of verifying the system robustness on both the message level and the model level. However, it was verified on the system level that the system was still viable in terms of real-time deployment with high throughput rates while suffering from an extra latency of less than 1 ms.

The following practical implications can be derived from the results: standardized IoT middleware can be integrated with the adversarially robust intrusion detection system natively without compromising the interoperability, which makes this approach suitable for critical infrastructure, industrial automation and smart city applications. Unlike other studies, this study focused on developing not a theoretical model, but rather a practical solution that covers robustness, performance and standardization.

However, despite the above contributions, there are some unexplored areas that can be considered in future work. Firstly, the framework generalization can be enhanced by considering IoT middleware platforms other than OneM2M. Secondly, another complex threat model that can be considered in future work is the adaptive threat model with long-term learning and strategy evolution. Thirdly, the framework applicability can also be enhanced by considering lightweight model compression and energy optimization. Finally, to provide end-to-end security in massive IoT systems, the proposed framework can be integrated with trust management and blockchain-based audit mechanisms.

In summary, this research has demonstrated that standardized IoT solutions can provide effective, adaptive and implementable security. It provides a practical foundation for securing next-generation IoT networks against both current and future threats by integrating adversarial learning into the OneM2M architecture.

ADDITIONAL INFORMATION AND DECLARATIONS

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: H.J.: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – Original Draft Preparation, Visualization. A.Z.: Conceptualization, Methodology, Validation, Supervision, Writing – Review and Editing, Project administration.

Statement on the Use of Artificial Intelligence Tools: The authors declare that they didn't use artificial intelligence tools for text or other media generation in this article.

Data Availability: Additional data that support the findings of this study are available from the corresponding author.

REFERENCES

Alhajjar, E., Maxwell, P., & Bastian, N. (2021). Adversarial machine learning in Network Intrusion Detection Systems. *Expert Systems with Applications*, 186, 115782. https://doi.org/10.1016/j.eswa.2021.115782

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176. https://doi.org/10.1109/comst.2015.2494502

Debicha, I., Debatty, T., Dricot, J.-M., & Mees, W. (2021). Adversarial Training for Deep Learning-based Intrusion Detection Systems. In *The Sixteenth International Conference on Systems (ICONS 2021),* (pp. 45–49). Royal Military Academy.

Diro, A., & Chilamkurti, N. (2018). Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications. *IEEE Communications Magazine*, 56(9), 124–130. https://doi.org/10.1109/mcom.2018.1701270

Elloumi, O. (2016). oneM2M and Smart M2M Introduction, Release 2/3. https://docbox.etsi.org/Workshop/2016/201611_M2MIoTWS/00_WORKSHOP/S00_INTRODUCTION/oneM2MSTATUS_ELLOUMI.pdf

Jamiri, H., & Zyane, A. (2025). Real-Time Integration of Adversarial-Robust IDS into OneM2M: Security and Hardware Performance Evaluation. In *2025 IEEE International Conference on Advances in Data-Driven Analytics and Intelligent Systems (ADACIS).* IEEE. https://doi.org/10.1109/ADACIS65663.2025.11436957

Martínez Beltrán, E. T., Sánchez Sánchez, P. M., López Bernal, S., Bovet, G., Gil Pérez, M., Martínez Pérez, G., & Huertas Celdrán, A. (2024). Mitigating communications threats in decentralized federated learning through moving target defense. *Wireless Networks*, 30(9), 7407–7421. https://doi.org/10.1007/s11276-024-03667-8

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing*, 17(3), 12–22. https://doi.org/10.1109/mprv.2018.03367731

Nguyen, X.-H., Nguyen, X.-D., Huynh, H.-H., & Le, K.-H. (2022). Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways. *Sensors*, 22(2), 432. https://doi.org/10.3390/s22020432

Papadopoulos, P., Thornewill von Essen, O., Pitropakis, N., Chrysoulas, C., Mylonas, A., & Buchanan, W. J. (2021). Launching Adversarial Attacks against Network Intrusion Detection Systems for IoT. *Journal of Cybersecurity and Privacy*, 1(2), 252–273. https://doi.org/10.3390/jcp1020014

Sagduyu, Y. E., Shi, Y., & Erpek, T. (2019). IoT Network Security from the Perspective of Adversarial Deep Learning. In *2019 16Th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, (pp. 1–9). IEEE. https://doi.org/10.1109/sahcn.2019.8824956

Suciu, O., Coull, S.E., & Johns, J. (2018). Exploring Adversarial Examples in Malware Detection. In *2019 IEEE Security and Privacy Workshops (SPW)*, (pp. 8–14). IEEE. https://doi.org/10.1109/SPW.2019.00015

Tuna, G., Kogias, D. G., Gungor, V. C., Gezer, C., Taşkın, E., & Ayday, E. (2017). A survey on information security threats and solutions for Machine to Machine (M2M) communications. *Journal of Parallel and Distributed Computing*, 109, 142–154. https://doi.org/10.1016/j.jpdc.2017.05.021

Acta Informatica Pragensia is published by the Prague University of Economics and Business, Czech Republic | eISSN: 1805-4951