Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/xxxx

Trustworthy Analytics in ETSI ZSM: A 5G Security Case Study

PANAGIOTIS RADOGLOU-GRAMMATIKIS[1,5], EFKLIDIS KATSAROS[1], EVANGELOS SYRMOS[1], GEORGIOS P. KATSIKAS[2], DIMITRIOS KLONIDIS[2], VASILEIOS ARGYRIOU[3], THOMAS LAGKAS[4] and PANAGIOTIS SARIGIANNIDIS[5]

[1]K3Y LTD, Studentski district, Vitosha quarter, bl. 9, 1700, Sofia, Bulgaria (e-mails: pradoglou@k3y.bg, ekatsaros@k3y.bg, esyrmos@k3y.bg)
[2]UBITECH LIMITED, 95B Archiepiskopou Makariou C' 3020 Limassol Cyprus (e-mails: gkatsikas@ubitech.eu, dklonidis@ubitech.eu)
[3]Department of Networks and Digital Media, Kingston University London, Penrhyn Road, Kingston upon Thames, Surrey KT1 2EE, UK (e-mail: vasileios.argyriou@kingston.ac.uk)
[4]Department of Computer Science, Democritus University of Thrace, 65404 Kavala, Greece (e-mail: tlagkas@cs.duth.gr)
[5]Department of Electrical and Computer Engineering, University of Western Macedonia, 50100 Kozani, Greece (e-mail:pradoglou@uowm.gr, psarigiannidis@uowm.gr).

CORRESPONDING AUTHOR: Panagiotis Radoglou-Grammatikis (e-mail: pradoglou@k3y.bg)

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101097122 (ACROSS). Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

ABSTRACT Towards the advent of the sixth generation (6G) wireless networks, smart technologies play a key role in the end-to-end automation of services across multiple domains. In particular, they create a new reality with multiple benefits, including intent-driven management, ultra-speed communication services and holistic integration within the Internet of Things (IoT). In this context, the Zero-touch Network and Service Management (ZSM) group within the European Telecommunications Standards Institute (ETSI) aims to provide an architectural framework which will allow the zero-touch orchestration of network services in a multi-domain fashion. Therefore, in this paper, we investigate the role of cross-domain storage, communication and analytics services within the architectural framework of ETSI ZSM. For this purpose, we take into account a particular case study which focuses on the orchestration of security services within the 5G core. More specifically, the deployment of a new User Plane Function (UPF) equipped with intrusion detection services that leverage Artificial Intelligence (AI) is investigated. For this purpose, the aforementioned cross-domain services are used to assess the security of the respective AI models before the onboarding of the new UPF within the 5G core. Based on this case study, a new security game is investigated, exploring and modelling the strategies of potential attackers and defenders. Furthermore, the architectural design and implementation of the cross-domain services are provided. Finally, the evaluation results show that the cross-domain analytics services are able to assess the security and resilience of the AI models and guide the orchestration functions.

INDEX TERMS 6G, Artificial Intelligence, Adversarial Attacks, Security, ZSM

I. Introduction

The integration of smart technologies within the sixth generation (6G) era of wireless communication technology provides several benefits. In particular, the advent of 6G [1] creates a new reality with real-time and ultra-speed communication services across multiple stakeholders. Moreover, 6G will allow the smooth integration of the Internet of Things (IoT) [2] at all levels, from Home Area Networks (HANs) to Wide Area Networks (WANs). Furthermore, 6G will combine and integrate Artificial Intelligence (AI) and automation services, thus enabling predictive maintenance and energy efficiency. Both academia and industrial players (such as operators, equipment vendors and technology providers) are working towards the 6G era, driving and integrating innovations in network management. A characteristic example is the Zero-touch Network and Service Management (ZSM) group

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

within the European Telecommunications Standards Institute (ETSI), focusing on creating a framework and standards for intelligent, automated and effective management of network services and operations.

In particular, towards the full automation across multiple network domains, the goal of ETZI ZSM is to deliver a framework with four functional objectives: (a) zero-touch automation, (b) end-to-end management, (c) service-oriented approach and (d) intent-driven operations. First, zero-touch automation focuses on minimising the role of human, thus allowing faster deployment and adaptation to network and service requirements. Next, end-to-end management is responsible for handling the network resources in a holistic manner across multiple domains and technologies. This kind of management includes the entire lifecycle of network resources and services, from their creation and deployment to real-time operation phases. Third, it is worth mentioning that ETSI ZSM manages services as independent entities, abstracting the complexities of network configuration with the goal of allowing the efficient creation, modification and scaling of these services based on the user needs. Finally, ETSI ZSM allows the users to define and rely on high-level objectives or intents, such as Service Level Agreements (SLAs), and the framework converts these intents into particular actions across the network.

Towards these objectives, the role of data storage and communication services is significant. In particular, these services are responsible for acquiring and storing huge amounts of data, respectively, including multiple types, such as network traffic data, system logs and telemetry data. This data is critical for monitoring the network operations in a multi-domain fashion and allowing intelligent decision-making and automated services. To acquire this data is necessary to establish the appropriate communication channels. Therefore, the availability of synchronous and asynchronous communication services is necessary. On the other hand, analytics and intelligence also play a crucial role in accomplishing the objectives of ETSI ZSM. More specifically, these services monitor the network health, performance and quality of services by processing various kinds of data, as mentioned before. Analytics services are responsible for interpreting and transforming the network data into appropriate formats, integrating also some predictive models. Then, building on the outcomes of analytics, the intelligence services provide powerful models that can guide automated actions.

In light of the aforementioned remarks, in this paper, we focus our attention on the storage, communication and analytics services within ETSI ZSM. Through a game theoretic case study, in this paper, we design, instantiate and validate these services. First, we define a security game which focuses on the security of 5G core environments. In this game, the attacker aims to mislead the security services of the 5G core, while the defender leverages the storage, communication and analytics services in order to protect the security services of the 5G core. Consequently, the contributions of this paper are summarised as follows:

- **New Security Game**: A new security game is introduced, investigating the attack and defence actions regarding malicious activities that target the security services of the 5G core.
- **Design and Implementation of ZSM Storage Communication and Analytics Services**: Following the principles of ETSI ZSM, the storage, communication and analytic services are designed and implemented in order to solve the previous security game.

Therefore, the rest of this paper is organised as follows. Section II provides a background on the ZSM architecture and summarises similar works in this research area. Next, section III presents a security game that investigates malicious activities against the security services of 5G core and how they can impact energy facilities. Next, section IV focuses on the design and implementation of the storage, communication and analytics services. Finally, section V focuses on the evaluation analysis, paying special attention to the role of analytics, while section VI concludes this paper.

II. Background & Related Work

In the context of telecommunication settings, the networks can be divided into various domains, such as core networks, access networks and edge networks. Each domain serves a dedicated role and includes different technologies and services. However, it is evident that managing each domain in an isolated manner appears to be a complex, demanding and inefficient task, considering that in the era of 6G and smart networks, the computing and communication services span multiple domains. Therefore, by introducing a multi-domain context, the ETSI ZSM aims to provide, instantiate and integrate an entire architectural framework that allows the orchestration and automation of network services across multiple domains in a coordinated and synchronised manner. This framework will bring multiple benefits, such as end-to-end automation, cross-domain orchestration, improved resource optimisation, faster fault detection and resolution, consistent policy enforcement and agility in service deployment. For instance, considering a 6G-related case study, an operator may need to provide a low-latency video streaming service. To this end, synchronisation and collaboration between the access, core and edge computing resources is required. Thus, through the ETSI ZSM framework, the orchestration of these computing resources is automated with the goal of ensuring that the video stream is delivered with low latency, the data is routed in an optimal way, and only the appropriate resources are used, taking into consideration the objectives of each domain as a whole. However, despite the beneficial services provided by the ETSI ZSM architecture, it also outlines the need for coordinated and synchronised security mechanisms in a multi-domain fashion. In particular, without the presence of efficient security mechanisms, security issues

can span other domains, resulting in cascading effects. Therefore, cross-domain analytics services should not only be limited to optimising service efficiency but also handling security breaches across interconnected systems. Finally, it is evident that consistent and smart policy enforcement across domains becomes even more critical in terms of preventing cyberthreats like unauthorised access, data manipulation and service disruption. In this paper, we focus our attention on mitigating data manipulation actions that may mislead the decision-making process of AI-powered security services used by 5G core environments.

Based on the aforementioned remarks, ETSI ZSM follows a modular and scalable architectural design composed of several layers with a specific scope (Fig. 1). As highlighted in Fig. 1, in this paper, we focus our attention on Cross-domain Data Services in terms of providing cross-domain data storage and analytics services. First, a key building block of ETZI ZSM is the Management Domains, which can refer to different network segments and services, such as IoT, Radio Access Network (RAN), and cloud environments. In particular, each domain is responsible for handling computing resources through (a) Data Services (DS) and (b) Management Functions (MF). On the one hand, DS offer the necessary storage infrastructure and mechanisms for data collection, data processing, and data access. On the other hand, MFs are modular services with a specific task, such as for example performance management, configuration management and security management. To enhance scalability, each MF consists of several microservices with close-loop automation mechanisms that include data monitoring, analysis, decision-making and orchestration. The communication across the DS and MFs within a single domain is accomplished through a Domain Integration Fabric, which provides the necessary interfaces and event messaging infrastructure.

Another key element of the ZSM architecture is the E2E service Management Domain. Similarly, this domain is composed of DS and MFs that are responsible for controlling and managing the overall network dynamically in a multi-domain fashion. The same close-loop automation mechanisms are also utilised in the operational context of this domain. The communication between the E2E service Management Domain and the Management Domains is carried out through the Cross-Domain Integration Fabric, which includes the necessary communication channels and interfaces regarding the entire network management and synchronisation. More specifically, on the one hand, the Domain Integration Fabrics receives and sends through the Cross-Domain Integration Fabric various information like domain data and domain orchestration settings. On the other hand, the E2E service Management Domain receives and processes this data and, through the Cross-Domain Integration Fabric, guides the end-to-end orchestration through E2E analytics and intelligence. Finally, it is worth mentioning that for this network coordination and synchronisation between the E2E Service

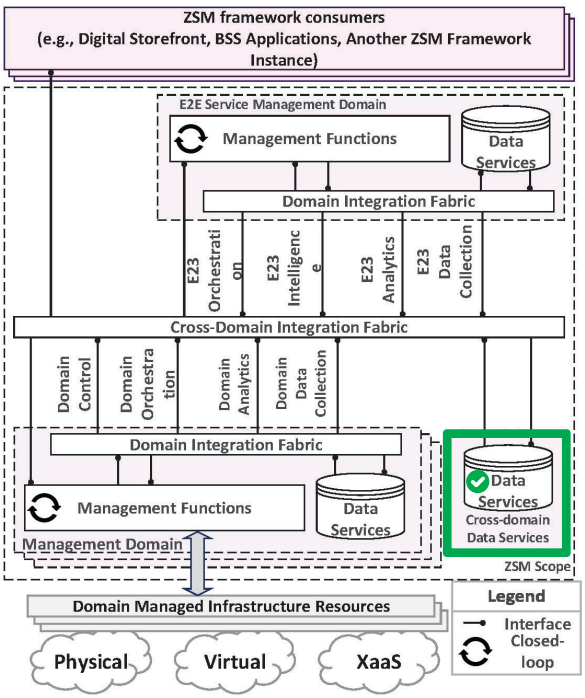

FIGURE 1. ETSI ZSM Architecture Framework

Management Domain and the Management Domain, the presence of cross-domain DS and MFs is important.

Several studies investigate the ZSM security issues. For instance, in [3], M. Linyanage et al. provide a survey about ZSM for 5G and beyond networks, investigating and assessing the severity of the ZSM threats in a qualitative manner. In [4], M. Xevgenis et al. introduce a blockchain approach in order to enhance the resilience of the ZSM framework. Similarly, in [5], the authors provide a federated intrusion detection mechanism applicable to the ZSM paradigm. Finally, in [6], the authors describe the advancements of the INSPIRE-5GPlus project, which combines a set of technologies like AI, network softwarization, distributed ledger technologies and Trusted Execution Environments (TEE), taking into account the ZSM and SD-SEC models. However, despite the fact that the previous works provide valuable solutions, there are no papers elaborating on the role of cross-domain storage, communication and analytics services within ETSI ZSM. Furthermore, although many works emphasise the significance of intelligence services within the ZSM framework, security issues against these services have not been thoroughly examined. Finally, although there are several studies that combine game theory and adversarial attacks, such as [7], [8], there are no security games focusing on adversarial attacks against AI-powered security services against 5GC.

Therefore, to the best of our knowledge, this is the first work elaborating on the cross-domain storage, communication and analytics services within the ETSI ZSM, while the security issues of the intelligence services are also

investigated and demonstrated practically in the context of a security game.

TABLE 1. Symbols and Notation

Symbol	Definition
A	Attacker
D	Defender
θ_A	Type of the Attacker
θ_D	Type of the Defender
Θ_A	Set of possible types of the Attacker
Θ_D	Set of possible types of the Defender
s_A	Strategy of the Attacker
s_D	Strategy of the Defender
$p_A(\theta_D)$	Attacker's belief about the Defender's type
$p_D(\theta_A)$	Defender's belief about the Attacker's type
$P(s_A, s_D)$	Probability of success of the attack
C_A	Cost function of the Attacker
C_D	Cost function of the Defender
α	Coefficient for the Attacker's success
β	Coefficient for the Attacker's cost
γ	Coefficient for the Defender's success
δ	Coefficient for the Defender's cost
k_1	Impact of s_A on success probability
k_2	Impact of s_D on success probability
c_1	Coefficient in the Attacker's cost function
c_2	Coefficient in the Defender's cost function
U_A	Payoff function of the Attacker
U_D	Payoff function of the Defender
$E[U_A]$	Expected payoff of the Attacker
$E[U_D]$	Expected payoff of the Defender
s_A^*	Optimal strategy of the Attacker
s_D^*	Optimal strategy of the Defender

III. A Security Game of Attacking and Defending Security Services of 5G Core

In this paper, we investigate a security game against the security services of 5G core environments. Based on the ZSM framework, this 5G core environment is considered a single network domain. Our attention focuses on the N4 interface and the security services used by the User Plane Function (UPF). More precisely, based on our previous works [9]–[12], a malicious Session Management Function (SMF) can target UPF by leveraging the security weaknesses of the Packet Forwarding Control Protocol (PFCP). Characteristic attacks that target the PFCP communications between SMF and UPF are PFCP Session Establishment DoS Attack, PFCP Session Deletion DoS Attack, PFCP Session Modification DoS Attack (DROP) and PFCP Session Modification DoS Attack (DUPL). All these attacks are explained in [11]. For this purpose, as illustrated in Fig. 2, in this scenario, a Domain Security Orchestrator (DSO) aims to deploy a UPF equipped with an AI-powered Intrusion Detection System (IDS) (as described in our previous work [10]) that can successfully recognise these attacks. However, before the deployment process, the DSO leverages the cross-domain services in order to verify that the AI models within UPF are resilient against adversarial attacks. In particular, the Cross-Domain Analytics Services are responsible for evaluating if the AI models are resilient against adversarial attacks. On the other hand, cross-domain storage and communication services are used to store the AI models and facilitate the communication between the DSO and the cross-domain services, respectively.

Therefore, we can consider a security game with two players: (a) Attacker (A) and (b) Defender (D). A intends to deceive the AI-powered IDS by generating adversarial samples, while the goal of D is to protect the AI models behind the operation of the AI-powered IDS. The strategies of A refer to particular black-box evasion adversarial attacks. In particular, for the sake of simplicity, in this paper, we consider only three adversarial attacks, namely: (a) Zeroth Order Optimization (ZOO), (b) HopSkipJump, and (c) Boundary attacks. while the strategies of the D are limited to adversarial training and the early detection of the above adversarial attacks. Therefore, the strategies of A and D are summarised below.

Attacker's - A' strategies: $S_A = \{A1, A2, A3\}$

$A1$ **- Zeroth Order Optimization (ZOO)**: The ZOO attack [13] works by iteratively querying the target model and using the responses to craft adversarial examples. It does not require access to gradients or internal model information. Instead, it relies solely on the model output to estimate the gradient and iteratively modify the input samples to generate adversarial perturbations. The gradient is estimated with zeroth-order optimization, more specifically, with the symmetric difference quotient method. The ZOO attack starts with an initial guess and refines this perturbation over multiple iterations to maximize the model prediction error. By iteratively querying the model and adjusting the perturbation based on the model responses, the attacker can craft adversarial examples that the target model misclassifies. The ZOO attack is widely applicable to a wide range of models and scenarios, due to its black-box nature.

$A3$ **- HopSkipJump**: The HopSkipJump attack [14] is another type of black-box attack commonly used in adversarial machine learning. It is known for its efficiency and effectiveness in generating adversarial examples, particularly against deep neural networks [15]–[17]. The attack starts with an initial guess for the adversarial perturbation and then iteratively refines it to maximize the model's prediction error while minimizing the perturbation size. The HopSkipJump attack can often find adversarial examples with minimal computational cost and query complexity, rendering it significantly more efficient than both the ZOO and Boundary attacks.

$A2$ **- Boundary**: Similar to ZOO, the Boundary attack [18] does not require access to either the model gradients or the

This article has been accepted for publication in IEEE Open Journal of the Communications Society. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJCOMS.2024.3505555

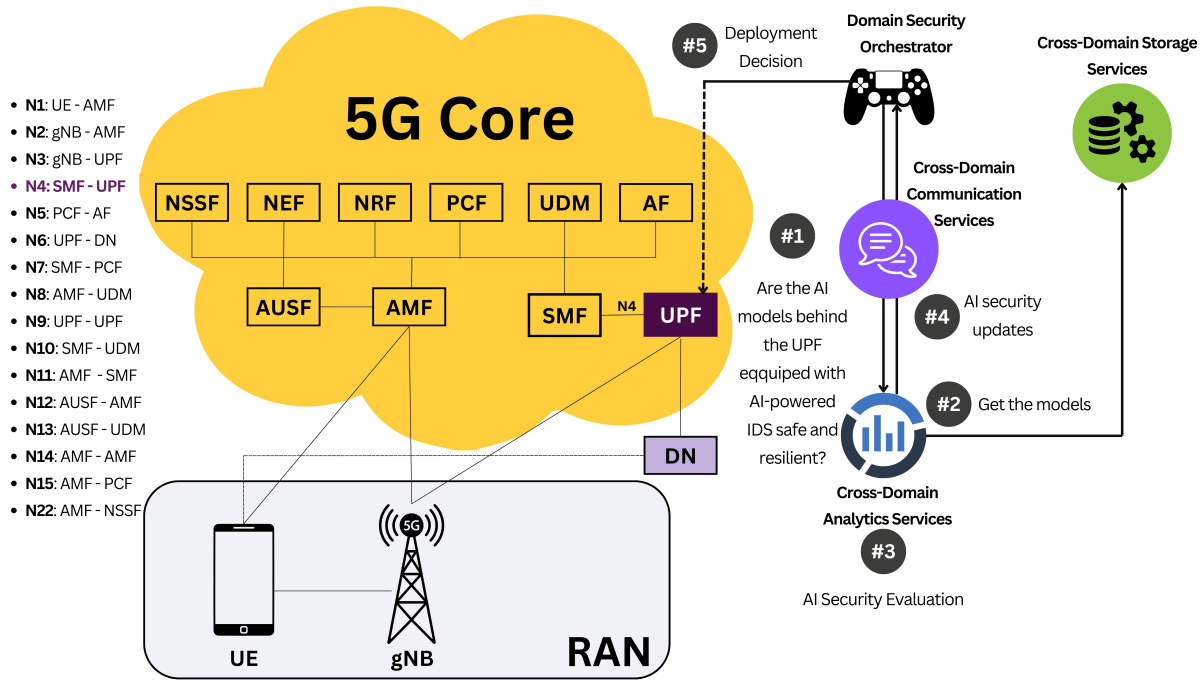

FIGURE 2. 5G Security Case Study

training data, and is thus model agnostic. In contrast to the ZOO attack, Boundary does not access the probability vectors, as the authors argue that it is uncommon for a publicly accessible model to reveal such information. Instead, this work relies on the raw model decision. The Boundary attack starts from a random adversarial sample and attempts to minimize its distance from the input sample by traversing the decision boundary and performing rejection sampling. The attack aims to find the smallest possible perturbation that causes the model to misclassify the input while staying within a specified distance threshold from the original data point.

It is worth noting that the aforementioned adversarial attacks can also raise severe privacy issues. For instance, both ZOO and HopSkipJump attacks can lead to successful membership inference or model invasion attempts. In particular, by querying the model repeatedly, the attacker may have the ability to understand whether the data samples were part of the training set and extract significant information regarding the trained model. On the other hand, by identifying the decision boundary of the model, the attacker can understand and expose how the model handles specific data samples. Consequently, the boundary attacks can result in successful model inversion or attribute inference attempts. Within a multi-domain context, potential adversaries can target the interconnection settings across the domains to perform repetitive queries against a specific domain, such as an edge environment, to understand and violate private information from another domain, like a core network. In the context of 5GC, such adversarial attacks can expose sensitive information, such as usage patterns, especially in the case of network slices relying on shared data.

Defender's - D' strategies: $S_D = \{D1, D2\}$

$D1$ **- Adversarial Training**: A natural strategy for defense against adversarial attacks is to utilize them as additional training data to make the models more robust. This approach, known as adversarial training [19], [20], involves generating adversarial examples using known attack methods and incorporating these examples into the training dataset. By exposing the model to adversarial samples during training, the model learns to recognize and correctly classify these perturbed inputs, thereby improving its overall robustness and resilience to future attacks. Adversarial training can be viewed as a form of data augmentation where the training set is enriched with examples that are specifically designed to exploit the model's vulnerabilities. This strategy helps the model to generalize better and reduces the risk of overfitting to the clean, unperturbed data. As a result, the model becomes better equipped to handle both seen and unseen adversarial attacks, improving its security in real-world applications.

$D2$ **- Detection of Adversarial Attacks**: Another strategy for mitigating adversarial attacks is their detection [21], [22], i.e. with a classification model capable of differentiating between original and adversarial input samples. This approach involves developing a separate "detector" model that can analyze input data and determine whether it has been manipulated to deceive the primary model. The detection of adversarial examples typically relies on identifying anomalies or unusual patterns in the input data that are characteristic of adversarial manipulations. These detectors

can be trained using supervised learning, where the model is trained on a labeled dataset containing both clean and adversarial examples. By learning the distinguishing features of adversarial inputs, the detector model can flag potentially harmful samples for further inspection or rejection. This strategy aims to prevent the harmful effects of such attacks, by identifying adversarial inputs before they are processed by the target model.

Beliefs of D: The beliefs of D regarding the attacker's type are given by $p(\theta_A)$, where $p(A1) = p1$, $p(A2) = p2$, $p(A3) = p3$, and $p1 + p2 + p3 = 1$.

Payoff Functions: Let $P_A(\theta_A, S_D)$ denote the payoff for the attacker A and $P_D(\theta_A, S_D)$ denote the payoff for the defender D, where θ_A is the attacker's A type and S_D is the defender's D strategy. The payoff functions are defined as follows:

$$P_A(\theta_A, S_D) = \begin{cases} P_A(A1, D1), & \text{if } \theta_A = A1 \text{ and } S_D = D1 \\ P_A(A1, D2), & \text{if } \theta_A = A1 \text{ and } S_D = D2 \\ P_A(A2, D1), & \text{if } \theta_A = A2 \text{ and } S_D = D1 \\ P_A(A2, D2), & \text{if } \theta_A = A2 \text{ and } S_D = D2 \\ P_A(A3, D1), & \text{if } \theta_A = A3 \text{ and } S_D = D1 \\ P_A(A3, D2), & \text{if } \theta_A = A3 \text{ and } S_D = D2 \end{cases}$$

$$P_D(\theta_A, S_D) = \begin{cases} P_D(A1, D1), & \text{if } \theta_A = A1 \text{ and } S_D = D1 \\ P_D(A1, D2), & \text{if } \theta_A = A1 \text{ and } S_D = D2 \\ P_D(A2, D1), & \text{if } \theta_A = A2 \text{ and } S_D = D1 \\ P_D(A2, D2), & \text{if } \theta_A = A2 \text{ and } S_D = D2 \\ P_D(A3, D1), & \text{if } \theta_A = A3 \text{ and } S_D = D1 \\ P_D(A3, D2), & \text{if } \theta_A = A3 \text{ and } S_D = D2 \end{cases}$$

Defender's - D**' Expected Payoff**: Defender's expected payoff for each strategy is calculated as follows:

For $D1$ (Adversarial Training):

$$\begin{aligned} E[P_D|D1] =& p_1 \cdot P_D(A1, D1) + \\ & p_2 \cdot P_D(A2, D1) + p_3 \cdot P_D(A3, D1) \end{aligned} \quad (1)$$

For $D2$ (Detection):

$$\begin{aligned} E[P_D|D2] =& p_1 \cdot P_D(A1, D2) + \\ & p_2 \cdot P_D(A2, D2) + p_3 \cdot P_D(A3, D2) \end{aligned} \quad (2)$$

Optimal Defense Strategy: The defender will choose $D1$ if $E[P_D|D1] > E[P_D|D2]$, otherwise they will choose $D2$.

Attacker's Best Response: Given the defender's strategy S_D, the attacker will choose the strategy that maximizes their payoff:

$$P_A(\theta_A, S_D)$$

The attacker will choose θ_A such that $P_A(\theta_A, S_D)$ is maximized.

Bayesian Nash Equilibrium: The Bayesian Nash equilibrium is a strategy profile where each player's strategy is the best response given their beliefs about the other player's type.

Attacker's A strategy: θ_A that maximizes $P_A(\theta_A, S_D)$.

Defender's D strategy: S_D that maximizes the expected payoff $E[P_D|S_D]$.

This combination forms the Bayesian Nash equilibrium because, given the defender's D belief, the defender's optimal strategy is S_D. Given the defender's strategy S_D, the attacker's A optimal strategy is θ_A. If we suppose that the payoffs of the attacker and the defender follow a non-linear function, the belief of A regarding the strategy of D can be expressed as: $\theta_A \in \Theta_A$. On one hand, the belief of D regarding the strategy of A can be expressed as $\theta_D \in \Theta_D$. On the other hand, the payoffs of A and D are summarised below.

$$U_A(s_A, s_D, \theta_A, \theta_D) = \alpha \cdot P(s_A, s_D) - \beta \cdot C_A^2 \quad (3)$$

$$U_D(s_A, s_D, \theta_A, \theta_D) = \gamma \cdot (1 - P(s_A, s_D)) - \delta \cdot C_D^2 \quad (4)$$

Similarly, the expected payoffs of Attacker's A and Defender's D are summarised below.

$$\begin{aligned} &E[U_A(s_A, s_D)] = \\ &\sum_{\theta_D \in \Theta_D} p_A(\theta_D) \left(\alpha \cdot P(s_A, s_D) - \beta \cdot C_A^2\right) \end{aligned} \quad (5)$$

$$\begin{aligned} &E[U_D(s_A, s_D)] = \\ &\sum_{\theta_A \in \Theta_A} p_D(\theta_A) \left(\gamma \cdot (1 - P(s_A, s_D)) - \delta \cdot C_D^2\right) \end{aligned} \quad (6)$$

If we consider the assumptions: $P(s_A, s_D) = k_1 \cdot s_A - k_2 \cdot s_D$, $C_A = c_1 \cdot s_A^2$ and $C_D = c_2 \cdot s_D^2$, to find the equilibrium, we can derive the first-order conditions by taking the derivatives of the expected payoffs with respect to each player's strategy and setting them to 0 as denoted below.

$$\begin{aligned} &\frac{\partial E[U_A(s_A, s_D)]}{\partial s_A} = \\ &\sum_{\theta_D \in \Theta_D} p_A(\theta_D) \left(\alpha \cdot \frac{\partial P(s_A, s_D)}{\partial s_A} - 2\beta \cdot C_A \cdot \frac{\partial C_A}{\partial s_A}\right) = 0 \end{aligned} \quad (7)$$

$$\begin{aligned} &\frac{\partial E[U_D(s_A, s_D)]}{\partial s_D} = \\ &\sum_{\theta_A \in \Theta_A} p_D(\theta_A) \left(\gamma \cdot \frac{\partial (1 - P(s_A, s_D))}{\partial s_D} - 2\delta \cdot C_D \cdot \frac{\partial C_D}{\partial s_D}\right) = \\ &0 \end{aligned} \quad (8)$$

Given the above assumptions, the payoff function of the Attacker A can be simplified as follows.

This article has been accepted for publication in IEEE Open Journal of the Communications Society. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJCOMS.2024.3505555

$$E[U_A(s_A, s_D)] = \alpha \cdot (k_1 \cdot s_A - k_2 \cdot s_D) - \beta \cdot (c_1 \cdot s_A^2) \quad (9)$$

Therefore, the first-order condition for the attacker is given below.

$$\alpha \cdot k_1 - 2\beta \cdot c_1 \cdot s_A = 0 \quad (10)$$

Solving for s_A:

$$s_A^* = \frac{\alpha \cdot k_1}{2\beta \cdot c_1} \quad (11)$$

Similarly, given the above assumptions, the payoff function of the Defender D can be simplified as follows.

$$E[U_D(s_A, s_D)] = \gamma \cdot (1 - (k_1 \cdot s_A - k_2 \cdot s_D)) - \delta \cdot (c_2 \cdot s_D^2) \quad (12)$$

Thus, the first-order condition for the Defender D is given below.

$$-\gamma \cdot (-k_2) - 2\delta \cdot c_2 \cdot s_D = 0 \quad (13)$$

Solving for s_D:

$$s_D^* = \frac{\gamma \cdot k_2}{2\delta \cdot c_2} \quad (14)$$

The strategies s_A^* and s_D^*, as provided below, constitute the Nash Equilibrium of the game. To prove that s_A^* and s_D^* are indeed the Nash Equilibrium, we need to show that given the strategies s_A^* and s_D^*, neither player can change their strategy to improve their expected payoff. Therefore, $s_D = s_D^*$. The attacker's A payoff function is provided as:

$$E[U_A(s_A, s_D^*)] = \alpha \cdot (k_1 \cdot s_A - k_2 \cdot s_D^*) - \beta \cdot (c_1 \cdot s_A^2) \quad (15)$$

Substitute s_D^* into the attacker's A payoff function:

$$E[U_A(s_A, s_D^*)] = \alpha \cdot (k_1 \cdot s_A - k_2 \cdot \frac{\gamma \cdot k_2}{2\delta \cdot c_2}) - \beta \cdot (c_1 \cdot s_A^2) \quad (16)$$

Taking the derivative with respect to s_A and setting it to zero:

$$\alpha \cdot k_1 - 2\beta \cdot c_1 \cdot s_A = 0 \quad (17)$$

$$s_A^* = \frac{\alpha \cdot k_1}{2\beta \cdot c_1} \quad (18)$$

Consequently, this s_A^* maximises the attacker's A expected payoff given s_D^*.

On the other hand, given $s_A = s_A^*$, the defender's D expected payoff is written by:

$$E[U_D(s_A^*, s_D)] = \gamma \cdot (1 - (k_1 \cdot s_A^* - k_2 \cdot s_D)) - \delta \cdot (c_2 \cdot s_D^2) \quad (19)$$

Similarly, by substituting s_A^* into the defender's D payoff function:)

$$E[U_D(s_A^*, s_D)] = \gamma \cdot (1 - (k_1 \cdot \frac{\alpha \cdot k_1}{2\beta \cdot c_1} - k_2 \cdot s_D)) - \delta \cdot (c_2 \cdot s_D^2) \quad (20)$$

Taking the derivative with respect to s_D and setting it to zero:

$$-\gamma \cdot (-k_2) - 2\delta \cdot c_2 \cdot s_D = 0 \quad (21)$$

$$s_D^* = \frac{\gamma \cdot k_2}{2\delta \cdot c_2} \quad (22)$$

Thus, s_D^* maximizes the defender's expected payoff given s_A^*.

IV. Cross-Domain Storage, Communication & Analytics Services

On the basis of the previous security game, the cross-domain storage, communication and analytics services are designed and implemented following the principles of the ZSM framework. More specifically, these services refer to the cross-domain DS and MFs; however, they also could serve (if needed) the MFs and DS of the E2E Service Management Domain and the single Management Domains. The goal is to allow the analytics services to support all the strategies of the previous security game, including the attack strategies as a proactive measure for enhanced security. Fig. 3 depicts the architectural design of these services and how they are interconnected with each other. Before going ahead with the analytics services, we first describe the cross-domain storage and communication services.

A. Cross-Domain Storage Services

Two cross-domain storage services are provided, namely (a) Block, Object and File Storage Service and (b) Data Ingestion, Storage and Search Service. The first service is responsible for storing data blocks, objects and files. For this purpose, MinIO is utilised as a software container. The data interactions are carried out through (a) Representational State Transfer (REST) Application Programming Interface (API) and (b) web User Interface (UI). More specifically, the architectural design of this service includes three functional layers. The lowest layer comprises the available buckets managed by the host environment. Next, the intermediate layer features the operating Docker instance of MinIO. Finally, the top layer represents a lightweight REST API developed with the FastAPI framework, which abstracts the MinIO functions. To ensure future cloud migration, the API adheres to the S3 API interface and uses the Python MinIO Client SDK for a persistent connection with MinIO's

This article has been accepted for publication in IEEE Open Journal of the Communications Society. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJCOMS.2024.3505555

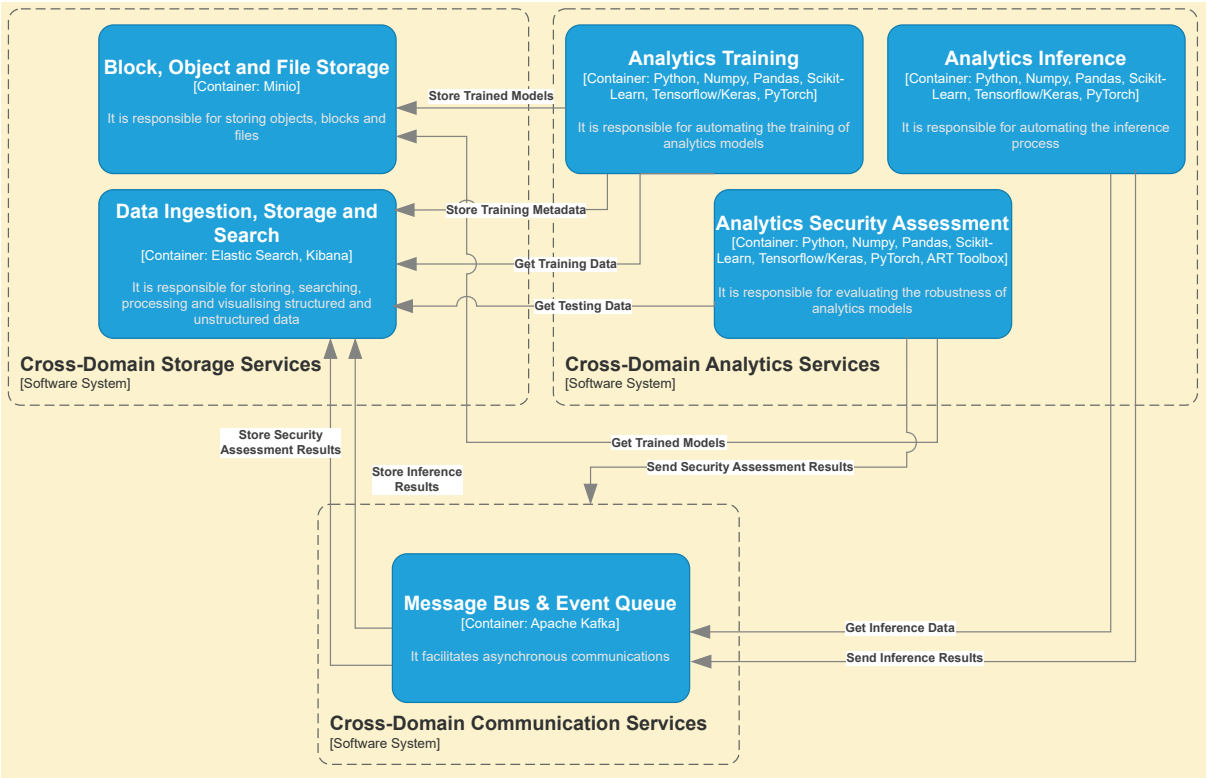

FIGURE 3. **Architecture of Cross-Domain Storage, Communication and Analytics Services**

ACCESS_KEY. The API extends the HTTP response timeout to handle large files, supporting HTTP POST, PUT DELETE and GET methods for file management. On the other hand, the Data Ingestion, Storage and Search Service is a powerful engine allowing distributed data search. For this purpose, Elasticsearch is utilised as a software container. Table A and Table 3 summarise the interfaces of the Block, Object and File Storage and the Data Ingestion, Storage and Search services, respectively.

B. Cross-Domain Communication Services

The cross-domain communication services are implemented through a Message Queue and Event Bus component. The role of this component is to facilitate data transfer across different components and services. Apache Kafka and Zookeeper are used for this service. Apache Kafka is an open-source distributed event streaming platform capable of handling real-time data feeds and serves as the core message bus for transferring data from data aggregators to other ACROSS services. Zookeeper manages and coordinates Kafka's distributed processes. To secure data transfer and prevent man-in-the-middle attacks, Kafka is configured with Transport Layer Security (TLS), requiring each client to use the provided certificates and credentials for a persistent connection. Kafka handles the creation, validation, and rollout of these certificates.

C. Cross-Domain Analytics Services

The cross-domain analytics services in this paper include the following tree functions: (a) training, (b) inference and (c) security assessment.

Analytics Training: The Analytics Training service constitutes a pipeline for training and evaluating machine learning models using k-fold cross-validation, selecting the best-performing model configuration, and sending the trained models and metadata to the Storage Services. A "main" python function orchestrates the pipeline by parsing command-line arguments, executing data retrieval, model training, validation and output handling. The Analytics Training Service requests and receives the indexed and labelled training data from the Data Ingestion, Storage and Search Service, with python requests and json. The same libraries are utilized to send the trained models back to the Storage Services, namely, back to the Block, Object and File Storage Service. The pandas and scikit-learn libraries are used to process the data and train the AI models. The output models are saved with the python joblib library.

The pipeline starts with obtaining data from a specified API endpoint using the provided URL and query parameters, with the Data Receptor, and dropping the irrelevant columns with the Data Processor, which subsequently returns a pandas DataFrame. Thereafter, the pipeline trains and evaluates different types of machine learning models with the K-fold Cross-Validator: Linear Discriminant Analysis (LDA), Logistic Regression (LR), Decision Tree (TR), Random

TABLE 2. Interfaces of the Block, Object and File Storage

Operation	HTTP Method	Endpoint	Description
Status Check	GET	`/api/healthcheck/`	Check if the API is operational.
Create	POST	`/api/buckets/{\ bucket_name}`	Create a new bucket.
Read	GET	`/api/buckets/{\ bucket_name}`	Retrieve details of a specific bucket.
Update	PUT	`/api/buckets/{\ bucket_name}`	Update settings or properties of a bucket.
Delete	DELETE	`/api/buckets/{\ bucket_name}`	Delete a specific bucket.

TABLE 3. Interfaces of Data Ingestion, Storage and Search Services

Operation	HTTP Method	Endpoint	Description
Create	POST	`/api/indexes/{\ index_name}`	Create a new index.
Read	GET	`/api/indexes/{\ index_name}`	Retrieve configuration and status of a specific index.
Update	PUT	`/api/indexes/{\ index_name}/settings`	Update settings of a specific index.
Delete	DELETE	`/api/indexes/{\ index_name}`	Delete a specific index.

TABLE 4. Interfaces of the Message Bus and Event Queue

Operation	HTTP Method	Endpoint	Description
Create	POST	/topics/{topic}	Produces a message to the specified Kafka topic.
Read	GET	/topics/{topic}	Consumes messages from the specified Kafka topic.
Update	PUT	/topics/{topic}	Updates the configuration of the specified Kafka topic.
Delete	DELETE	/topics/{topic}	Deletes the specified Kafka topic.

TABLE 5. Interfaces of the Analytics Services

HTTP Method	Endpoint	Description
POST	/analytics/training	Trains, cross-validates and saves multiple AI models.
POST	/analytics/inference	Evaluates a specific AI model over a given data sample or batch of samples.
POST	/analytics/data	Retrieves and processes data.
POST	/analytics/generator	Generates adversarial data for a specific AI model with a specific adversarial attack.
POST	/analytics/verificator	Assesses performance on adversarial data for a specific AI model.
POST	/analytics/advtrainer	Retrains an AI model with both real and adversarial data for improved robustness.
POST	/analytics/discriminator	Trains an XGBoost AI Model for adversarial data classification.

Forest (RF), Extreme Gradient Boosting (XGB) and the Multi-Layer Perceptron (MLP). This process involves cross-validation and model selection for each of the models. Firstly, a wide range of models for each model type are initialized with diverse sets of hyperparameters. Thereafter, the dataset is split into k-folds. The models are trained and evaluated iteratively on different folds to ensure robust performance estimation. All possible combinations of these hyperparameters are explored to find the best model configuration. For each fold and parameter combination, the model is evaluated using various metrics including accuracy, F1-score, ROC AUC, True Positive Rate (TPR), and False Positive Rate (FPR). After cross-validation, the best-performing model configuration for each model type is determined based on the optimal performance of a specified metric (e.g., F1-score). Consequently, the output of this function is six different models (one for each class) and the respective metadata, including performance across all five metrics and a dictionary with the explicit hyperparameters. The best-performing models and their metadata are both saved locally and sent back to the Block, Object and File Storage Service via the Model Dispatcher.

The Analytics Training Service interfaces with the Data Ingestion, and Search Service to retrieve the data required for model training. The trained models are subsequently sent to the Block, Object and File Storage Service.

Analytics Inference: The Analytics Inference Service implements a framework to perform real-time, online predictions on streaming data sourced from Kafka via a designated topic. The system continuously ingests data via the Data Receptor and Processor modules. Thereafter, a selected AI model is loaded via the Model Receptor. The incoming data

streams, along with the model predictions are transferred to the Communication Services via the Data Dispatcher. Here, Confluent Kafka is employed to i) ingest the incoming data streams from and ii) send them back to the Communication Services after the prediction, through a designated consumer and producer, respectively. Moreover, pandas is utilized to transform the data instances into a structured DataFrame and concatenate them with the model predictions.

Initially, within the Data Receptor, a Kafka consumer is initialized with the provided configurations, including broker IP address, port, consumer group ID, and auto offset reset. The consumer subscribes to the aggregator topic to start receiving messages. Simultaneously, the model specified from the configurations is downloaded from the Block, Object and File Storage Service and is loaded onto the memory with the Model Receptor and Initializer respectively. In the second phase, the Kafka consumer starts receiving payloads. Each received payload is decoded into a DataFrame representing the data instance with the Data Processor. The resulting data instance is passed through the loaded machine learning model which makes predictions with the Model Evaluator. Consequently, the prediction result is appended to the instance DataFrame and serialized into JSON format. The serialized JSON payload is published to the respective topic, with the Data Dispatcher.

The communication of the Analytics Inference Service with the Communication Services is effectuated via subscribing to the Kafka Topic. The Analytics Inference Service further interfaces with the Storage Services, specifically with the Block, Object and File Storage Service, to download the model requested to be utilized for performing predictions.

Analytics Security Assessment: The Analytics Security Assessment Service consists of various functions. First, the Data Receptor receives the data from the Storage Services while the Data Processor preprocesses them. The Model Receptor loads the trained models. Thereafter, the Adversarial Generator is utilized to generate adversarial samples capable of deceiving the machine learning models. The generated datasets are transferred to the Storage Service via the Data Dispatcher. The Resilience Verificator function measures the performance of a given machine learning classifier on both real and adversarial data, assesses whether accuracy degradation is critical for that model, and notifies the user accordingly via the User Notificator. Another, defense-related functionality, namely the Adversarial Discriminator, is trained to detect and filter out potential evasion adversarial samples. Last, the Adversarial Trainer (AT) uses the adversarially generated data to augment the training dataset and retrain the AI models to make them more robust. The popular Adversarial Robustness Toolbox (ART) is used with python for the adversarial attacks, whereas the machine learning models and metrics rely again on the scikit-learn library.

The pipeline starts with the Data Receptor and the Data Processor, which load and process JSON data, drop specific columns based on the chosen aggregator, and save the results to a CSV file. The Generator performs adversarial data generation using a pre-trained machine learning model and an adversarial attack, configured via command-line arguments. It starts by fetching and processing data using the Data Receptor and Processor functions, then loads the model from the Block, Object and File Storage Service with the Model Receptor. Multiple instances of the attack are subsequently configured using a parameter grid, and each attack generates adversarial datasets from the original data. Different metrics are utilized to measure affinity to the original data, including correlation, L1, and Wasserstein distance. The adversarial datasets are sent back to the Storage Services.

Subsequently, the Resilience Verificator evaluates the performance degradation of a trained AI model under some of the generated adversarial attacks. Specifically, AI model performance is evaluated on both the actual and adversarial data, and the results are compared. Metrics are computed to determine the accuracy degradation caused by the adversarial attacks, with the results saved for further analysis with the User Notificator module.

The Analytics Security Assessment Service offers two additional defense mechanisms. The Adversarial Trainer (AT) utilizes the generated data to augment the training dataset with informative, adversarial samples and retrains the AI models to increase their performance, aiming for robustness. The User Notificator is employed to communicate the improvements with the user. The Discriminator, on the contrary, proactively processes and decides whether a data sample is real or adversarially generated. Specifically, it loads real and adversarial data, combines them into a single dataset, and constructs their labels as to whether they are adversarial or not. Then, it utilizes cross-validation to train and validate an XGBoost ML model to classify whether an instance is real or adversarially generated data, and sends the results to the user, again, via the User Notificator.

The Analytics Security Assessment Service communicates with the Storage Services for downloading the actual and uploading the adversarially generated data, and with the Block, Object and File Storage Service for downloading and uploading models. A general description of all Analytics Services APIs is provided in Table 5.

V. Evaluation Analysis

Initially, we validate the predictive capacity of the training component. Specifically, we train the aforementioned machine learning classifiers, including Linear Discriminant Analysis (LDA), Logistic Regression (LR), Decision Tree (TR), Random Forest (RF), Extreme Gradient Boosting (XGB) and Multi-Layer Perceptron (MLP). We resort to 5-fold cross-validation to tune each model class hyperparameters and estimate performance. Regarding MLP, we further optimize the neural network architecture. To ensure impartiality, we maintain identical splits for cross-validation and consider a diverse range of candidate models, spanning a total of 264 trained models to mimic production environment

processes. We choose the best model based on the accuracy metric since the dataset is balanced. However, the training component further supports F1, TPR, FPR and AUC.

Next, adversarial attacks are implemented to assess the robustness of the trained models. The proposed methodology subjects the test set to three different, widely recognized black box attacks, namely the ZOO, HopSkipJump, and Boundary attacks. Rather than blindly employing adversarial attacks, such as in computer vision tasks where subtle pixel alterations do not affect image semantics, we adopt a more sophisticated approach, where we prioritize maintaining a strong similarity between the adversarial and actual test dataset, aiming for high affinity. To quantify this affinity, we utilize metrics including the columnwise correlation matrix, L1 distance, and Wasserstein distance. Therefore, for each type of attack and model, we generate multiple adversarial datasets and select the one that maximally degrades classification performance while maintaining high fidelity with the original data. We evaluate the ML models on the adversarial data and report performance for all combinations of models and attacks. The exact same procedure is further followed for three popular white-box attacks, namely FGSM (Fast Gradient Sign Method) [23] and PGD (Projected Gradient Descent) [19] for gradient-learnt methods and DTA (Decision Tree Attack) [18] for decision trees, that circumvents gradient computation by traversing the learnt tree structure.

After generating the adversarial datasets and computing performance degradation, we retrain all ML models to assess the impact of the Adversarial Trainer on the Analytics Service. Specifically, we concatenate the original and adversarial datasets, retrain each model, and report performance. We compare the performance of models trained on the adversarially-augmented datasets with those trained on the plain dataset. To account for the increased size of the adversarially-augmented datasets, we also compare performance with a dataset augmented via SMOTE [24], a very popular oversampling method, to confirm that any performance improvement is due to the generated adversarial data quality rather than size.

The models are implemented in the scikit-learn v1.1.3 [25], pytorch v1.12 [26] and xgboost v1.6.2 libraries. Regarding the implementation of adversarial attacks, we follow the popular ART library v1.15 [27]. All experiments are conducted in Python 3.10 on an Ubuntu 22.04 machine equipped with an Intel® Core™ i7-12700H processor and 16GB of RAM.

A. Evaluation Dataset

In the context of the evaluation results, the 5GC PFCP Intrusion Detection Dataset is utilized. The network traffic can be categorized into different flows based on their characteristics and the nature of the operations involved. The ”Normal” flow represents typical PFCP session establishment, where sessions are set up and managed without malicious intent. The ”Mal_Estab” flow depicts a flood attack during PFCP session establishment, aiming to overwhelm the system. Similarly, the ”Mal_Del” flow involves a flood attack targeting PFCP session deletions, attempting to delete ongoing sessions. The ”Mal_Mod” flow represents a PFCP session modification flood attack that maliciously alters sessions using the DROP Apply Action Field Flags, whereas the ”Mal_Mod2” flow involves a flood attack that modifies sessions using the DUPL Apply Action Field Flag.

The 5GFC dataset enumerates 84 variables, including the column of labels. The training and testing splits, comprising 1302 and 558 samples respectively, are retained as provided by the authors. More specifically, we keep the 1302 training data samples as is, and use them for cross-validation. Subsequently, we split the designated test set equally into i) the actual test set and ii) the left-out test set. Each of the two sets counts 279 test samples. We dedicate the i) actual test set to measure performance degradation after the evasion and the ii) left-out test set to assess performance improvement after retraining.

B. Evaluation Results

Initially, we validate the predictive capacity of the Training and Inference services. Table 6 illustrates the results of the AI models on the validation (5-Fold CV Performance) and test (Test Set & Left-out Test Set Performance) datasets. We observe that LDA and LR perform on par. Those models impose linear hyperplane boundaries between the classes. Performance improvement is significant when jumping onto tree-based models. Decision trees, random forests, and xgboost are very effective with heterogeneous datasets. In line with multiple studies and industry practices, XGB is the best-performing model when properly tuned. The MLP, in contrast, shows only marginal improvement over linear models and fails to achieve superior performance, even with a larger budget for architecture and hyper-parameter tuning. The issue with MLPs lacking behind tree-based methods on tabular datasets is an open research topic, thoroughly discussed and examined in a seminal work [28].

Next, we assess the effectiveness of the adversarial generator by evaluating model performance degradation when subjected on adversarial data. For the type of attack, the first and second columns of Tables 7, 8, and 9 illustrate performance degradation and adversarial data affinity, respectively. Our analysis reveals that the ZOO attack consistently produces adversarial samples that closely mirror the original data. We posit this behavior to the ZOO utilization of probability vectors rather than raw decisions. On the other hand, the HopSkipJump and Boundary attacks generate adversarial data relying solely on the raw model decisions. Despite the lesser information they utilize, HopSkipJump and Boundary attacks still craft adversarial samples which are very close to the actual test set, yet further apart from the ZOO samples. Generating adversarial samples only based on raw decisions poses greater complexity. However, both the HopSkipJump and Boundary attacks diminish security performance, result-

TABLE 6. Comparative results of various common ML/DL methods. This experiment includes no attacks, thus, illustrates performance on the clear data.

Method	5-Fold CV Performance					Test Set Performance					LO Test Set Performance				
(#models)	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC
LDA (8)	0.541	0.554	0.542	0.114	0.824	0.606	0.599	0.606	0.098	0.865	0.527	0.534	0.525	0.118	0.827
LR (16)	0.580	0.581	0.585	0.104	0.846	0.671	0.680	0.674	0.082	0.898	0.595	0.609	0.591	0.101	0.856
TR (12)	0.786	0.787	0.787	0.053	0.891	0.832	0.831	0.831	0.042	0.921	0.814	0.815	0.813	0.046	0.901
RF (18)	0.835	0.834	0.837	0.041	0.965	0.864	0.864	0.864	0.034	0.973	0.842	0.843	0.842	0.039	0.974
XGB (18)	**0.839**	**0.839**	**0.840**	**0.040**	**0.969**	**0.867**	**0.867**	**0.868**	**0.033**	**0.975**	**0.849**	**0.848**	**0.849**	**0.038**	**0.970**
MLP (192)	0.672	0.661	0.669	0.083	0.902	0.632	0.630	0.634	0.089	0.863	0.621	0.620	0.622	0.091	0.842

TABLE 7. Comparative results of various common ML/DL methods. This experiment includes ZOO attacks, thus illustrates performance on the adversarial data.

Method	Test Set Performance - After Evasion					Adversarial - Test Sets Affinity			LO Test Set Performance - Before AT					LO Test Set Performance - After AT				
(#attacks)	Accuracy	F1	TPR	FPR	AUC	Correlation	L1	Wasserstein	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC
LDA (54)	0.222	0.171	0.224	0.194	0.591	0.996	0.000206	0.000205	0.527	0.524	0.525	0.118	0.827	**0.531**	**0.534**	**0.528**	**0.117**	**0.835**
LR (54)	0.448	0.453	0.455	0.138	0.845	0.983	0.000626	0.000614	0.595	0.609	0.591	0.101	0.856	**0.645**	**0.651**	**0.641**	**0.089**	**0.861**
TR (54)	0.571	0.563	0.571	0.108	0.742	0.997	0.000058	0.000056	0.814	0.815	0.813	0.046	0.901	**0.824**	**0.825**	**0.826**	**0.044**	**0.925**
RF (54)	0.710	0.711	0.709	0.062	0.874	0.978	0.001318	0.001258	0.842	0.843	0.842	0.039	**0.974**	**0.860**	**0.859**	**0.860**	**0.035**	0.969
XGB (54)	0.645	0.649	0.647	0.089	0.944	0.984	0.000225	0.000222	0.849	0.848	0.849	0.038	0.970	**0.867**	**0.867**	**0.867**	**0.033**	**0.973**
MLP (54)	0.523	0.512	0.521	0.121	0.826	0.986	0.000203	0.000201	0.621	0.620	0.622	0.091	0.842	**0.632**	**0.629**	**0.632**	**0.088**	**0.870**

TABLE 8. Comparative results of various common ML/DL methods. This experiment includes Boundary attacks, thus illustrates performance on the adversarial data.

Method	Test Set Performance - After Evasion					Adversarial - Test Sets Affinity			LO Test Set Performance - Before AT					LO Test Set Performance - After AT				
(#attacks)	Accuracy	F1	TPR	FPR	AUC	Correlation	L1	Wasserstein	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC
LDA (24)	0.251	0.183	0.253	0.187	0.774	0.925	0.059401	0.058022	0.527	0.534	0.525	0.118	0.827	**0.527**	**0.538**	**0.525**	**0.118**	**0.838**
LR (24)	0.133	0.097	0.134	0.217	0.781	0.969	0.003879	0.003766	0.595	0.609	0.591	0.101	0.856	**0.613**	**0.626**	**0.612**	**0.097**	**0.861**
TR (24)	0.025	0.023	0.025	0.244	0.421	0.947	0.005045	0.004918	0.814	0.815	0.813	0.046	0.901	**0.832**	**0.832**	**0.831**	**0.042**	**0.916**
RF (24)	0.014	0.011	0.015	0.247	0.855	0.589	0.377801	0.351706	0.842	0.843	0.842	0.039	0.974	**0.858**	**0.857**	**0.855**	**0.035**	**0.974**
XGB (24)	0.039	0.041	0.041	0.239	0.627	0.792	0.071934	0.067848	0.849	0.848	0.849	0.038	0.970	**0.857**	**0.856**	**0.856**	**0.036**	**0.973**
MLP (24)	0.095	0.093	0.094	0.223	0.712	0.874	0.092116	0.882110	0.621	0.620	0.622	0.091	0.842	**0.644**	**0.645**	**0.639**	**0.088**	**0.883**

TABLE 9. Comparative results of various common ML/DL methods. This experiment includes HopSkipJump attacks, thus illustrates performance on the adversarial data.

Method	Test Set Performance - After Evasion					Adversarial - Test Sets Affinity			LO Test Set Performance - Before AT					LO Test Set Performance - After AT				
(#attacks)	Accuracy	F1	TPR	FPR	AUC	Correlation	L1	Wasserstein	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC
LDA (24)	0.168	0.099	0.173	0.208	0.748	0.986	0.000256	0.000239	**0.527**	**0.534**	**0.525**	**0.118**	0.827	0.525	0.504	0.524	0.119	**0.845**
LR (24)	0.141	0.101	0.141	0.215	0.776	0.955	0.006220	0.005921	0.595	0.609	0.591	0.101	0.856	**0.602**	**0.614**	**0.599**	**0.099**	**0.861**
TR (24)	0.022	0.019	0.021	0.246	0.411	0.981	0.001879	0.001797	0.814	0.815	0.813	0.046	0.901	**0.818**	**0.818**	**0.818**	**0.042**	**0.911**
RF (24)	0.086	0.084	0.083	0.231	0.864	0.909	0.035919	0.030664	0.842	0.843	0.842	0.039	**0.974**	**0.857**	**0.855**	**0.856**	**0.036**	0.969
XGB (24)	0.097	0.086	0.097	0.225	0.774	0.973	0.003227	0.003133	0.849	0.848	0.849	0.038	0.970	**0.857**	**0.856**	**0.856**	**0.036**	**0.973**
MLP (24)	0.101	0.103	0.101	0.219	0.780	0.942	0.006142	0.006007	0.621	0.620	0.622	0.091	0.842	**0.637**	**0.637**	0.639	**0.082**	**0.902**

TABLE 10. Comparative results of various common ML/DL methods. This experiment compares SMOTE oversampling with the utilized attacks.

Method	LO Test Set - Base					LO Test Set - SMOTE					LO Test Set - BA					LO Test Set Performance - ZOO					LO Test Set Performance - HSJ				
	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC
LDA	0.527	0.534	0.525	0.118	0.827	0.512	0.498	0.509	0.119	0.828	0.527	0.538	0.525	0.118	0.838	0.531	0.534	0.528	0.117	0.835	0.525	0.504	0.524	0.119	0.845
LR	0.595	0.609	0.591	0.101	0.856	0.620	0.631	0.617	0.095	0.861	0.613	0.626	0.612	0.097	0.861	0.645	0.651	0.641	0.089	0.861	0.602	0.614	0.599	0.099	0.861
TR	0.814	0.815	0.813	0.046	0.901	0.803	0.802	0.802	0.049	0.891	0.832	0.832	0.831	0.042	0.916	0.824	0.825	0.826	0.044	0.925	0.818	0.818	0.818	0.042	0.911
RF	0.842	0.843	0.842	0.039	0.974	0.853	0.852	0.852	0.037	0.971	0.858	0.857	0.855	0.035	0.974	0.860	0.859	0.860	0.035	0.969	0.857	0.855	0.856	0.036	0.969
XGB	0.849	0.848	0.849	0.038	0.970	0.842	0.841	0.842	0.039	0.973	0.857	0.856	0.856	0.036	0.973	0.867	0.867	0.867	0.033	0.973	0.857	0.856	0.856	0.036	0.973
MLP	0.621	0.620	0.622	0.091	0.842	0.624	0.623	0.623	0.089	0.851	0.632	0.629	0.632	0.088	0.870	0.644	0.645	0.639	0.088	0.883	0.637	0.637	0.639	0.082	0.902
AVG	0.710	0.710	0.707	0.072	0.895	0.709	0.708	0.708	0.071	0.896	0.720	0.723	0.719	0.069	0.905	0.729	0.730	0.727	0.068	0.908	0.716	0.714	0.715	0.069	0.910

TABLE 11. Comparative results of various common ML/DL methods. This experiment includes white box attacks, thus illustrates performance on the adversarial data.

Method	Test Set Performance - After Evasion					Adversarial - Test Sets Affinity			LO Test Set Performance - Before AT					LO Test Set Performance - After AT				
(#attacks)	Accuracy	F1	TPR	FPR	AUC	Correlation	L1	Wasserstein	Accuracy	F1	TPR	FPR	AUC	Accuracy	F1	TPR	FPR	AUC
LR - FGS (24)	0.315	0.353	0.315	0.172	0.612	0.955	0.002909	0.002733	0.595	0.609	0.591	0.101	0.856	0.659	0.665	0.655	0.085	0.858
MLP - FGS (24)	0.364	0.373	0.366	0.164	0.691	0.979	0.003722	0.003613	0.621	0.620	0.622	0.091	0.842	0.672	0.681	0.671	0.085	0.860
LR - PGD (36)	0.401	0.411	0.401	0.151	0.785	0.988	0.003947	0.003819	0.595	0.609	0.591	0.101	0.856	0.652	0.658	0.648	0.087	0.858
MLP - PGD (36)	0.304	0.316	0.305	0.178	0.604	0.967	0.003289	0.003122	0.621	0.620	0.622	0.091	0.842	0.667	0.671	0.664	0.086	0.859
TR - TBA (10)	0.029	0.030	0.027	0.242	0.421	0.947	0.022264	0.021658	0.814	0.815	0.813	0.046	0.901	0.828	0.828	0.827	0.043	0.937

ing in a predictive accuracy below that of a random classifier. Notably, regardless of which adversarial attack is employed, the performance of the AI classifiers suffers considerable degradation. Similar degradation is observed across all white box attacks as is illustrated in Table 11. This highlights the susceptibility of AI classifiers to maliciously crafted inputs, even in scenarios where the adversarial data closely resembles the original dataset.

Subsequently, we evaluate the performance of our models on the held-out test set subsequent to retraining them on the original training dataset combined with the adversarial data generated by each attack. For the type of attack, the third and fourth columns of Tables 7, 8, 9 and 11 illustrate performance after training and adversarial retraining (AT), respectively, to emphasize the differences. Overall, our findings highlight that all attacks benefit performance, yet note ZOO as the most impactful in augmenting IDS security, showcasing the most substantial improvement in security performance across all models. Regardless of the type of adversarial data employed, we consistently observe substantial performance enhancements. This trend holds true across all types of models and attacks. Interestingly, tree-based models seem to exploit the adversarial data better, due to their hierarchical structure providing the flexibility to create complex, non-linear decision boundaries by recursively partitioning the feature space into regions of varying shapes and sizes. On the other hand, linear models like LDA and LR are inherently limited in their ability to capture complex relationships in the data. Consequently, our analysis of the left-out test set corroborates the contribution of adversarial data in enhancing the 5G network security via the Analytics Service, leading to substantial improvements for multiple models across various performance metrics. Similar behavior is observed for the white-box attacks of Table 11. Performance improvements for the respective models are relatively higher, due to the capacity of white-box attacks to exploit more intricate model information. Last, Table 10 compares the performance of models trained on the plain (first column), the SMOTE-augmented (second column), the Boundary-augmented, the ZOO-augmented and the HopSkipJump-augmented datasets, respectively. We observe that the adversarial attacks produce more informative samples compared to SMOTE and, therefore, contribute more towards performance improvement upon retraining. Thus, this study illustrates that it is the informed nature of adversarial samples that makes ML models more robust and not just the increased data samples, as represented via the synthetic SMOTE instances.

VI. Conclusions

In this paper, we investigate the role of cross-domain storage, communication and analytics services within the architectural design of ETSI ZSM. For this purpose, a specific case study is utilised, considering the deployment of UPF equipped with an AI-powered IDS. Based on this case study, a security game is examined, investigating the potential strategies of the defender(s) and the attacker(s). Next, the design and implementation of the cross-domain storage, communication and analytics services follow with the goal of integrating the potential strategies of the defender(s) and the attacker(s) into the cross-domain analytics services. Finally, evaluation results show that the cross-domain analytics services are able to evaluate the security of the AI models and guide the deployment process.

References

[1] X. Deng, L. Wang, J. Gui, P. Jiang, X. Chen, F. Zeng, and S. Wan, "A review of 6g autonomous intelligent transportation systems: Mechanisms, applications and challenges," *Journal of Systems Architecture*, p. 102929, 2023.

[2] X. Deng, B. Chen, X. Chen, X. Pei, S. Wan, and S. K. Goudos, "A trusted edge computing system based on intelligent risk detection for smart iot," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 1445–1454, 2023.

[3] M. Liyanage, Q.-V. Pham, K. Dev, S. Bhattacharya, P. K. R. Maddikunta, T. R. Gadekallu, and G. Yenduri, "A survey on zero touch network and service management (zsm) for 5g and beyond networks," *Journal of Network and Computer Applications*, vol. 203, p. 103362, 2022.

[4] M. Xevgenis, D. G. Kogias, P. A. Karkazis, and H. C. Leligou, "Addressing zsm security issues with blockchain technology," *Future Internet*, vol. 15, no. 4, p. 129, 2023.

[5] F. Naeem, M. Ali, and G. Kaddoum, "Federated-learning-empowered semi-supervised active learning framework for intrusion detection in zsm," *IEEE Communications Magazine*, vol. 61, no. 2, pp. 88–94, 2023.

[6] G. Chollon, D. Ayed, R. A. Garriga, A. M. Zarca, A. Skarmeta, M. Christopoulou, W. Soussi, G. Gür, and U. Herzog, "Etsi zsm driven security management in future networks," in *2022 IEEE Future Networks World Forum (FNWF)*. IEEE, 2022, pp. 334–339.

[7] Y. Zhou, M. Kantarcioglu, and B. Xi, "A survey of game theoretic approach for adversarial machine learning," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 3, p. e1259, 2019.

[8] S. Sharma, "Game theory for adversarial attacks and defenses," *arXiv preprint arXiv:2110.06166*, 2021.

[9] G. Nakas, P. Radoglou-Grammatikis, G. Amponis, T. Lagkas, V. Argyriou, S. Goudos, and P. Sarigiannidis, "5g-fuzz: An attack generator for fuzzing 5gc, using generative adversarial networks," in *2023 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2023, pp. 347–352.

[10] P. Radoglou-Grammatikis, G. Nakas, G. Amponis, S. Giannakidou, T. Lagkas, V. Argyriou, S. Goudos, and P. Sarigiannidis, "5gcids: An intrusion detection system for 5g core with ai and explainability mechanisms," in *2023 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2023, pp. 353–358.

[11] G. Amponis, P. Radoglou-Grammatikis, T. Lagkas, W. Mallouli, A. Cavalli, D. Klonidis, E. Markakis, and P. Sarigiannidis, "Threatening the 5g core via pfcp dos attacks: the case of blocking uav communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, p. 124, 2022.

[12] G. Amponis, P. Radoglou-Grammatikis, T. Lagkas, S. Ouzounidis, M. Zevgara, I. Moscholios, S. Goudos, and P. Sarigiannidis, "Towards securing next-generation networks: Attacking 5g core/ran testbed," in *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*. IEEE, 2022, pp. 1–4.

[13] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[14] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*. IEEE, 2020, pp. 1277–1294.

[15] H. Qi, F. Ren, L. Wang, P. Jiang, S. Wan, and X. Deng, "Multi-compression scale dnn inference acceleration based on cloud-edge-end collaboration," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 1, pp. 1–25, 2024.

[16] C. Li, L. Chai, K. Jiang, Y. Zhang, J. Liu, and S. Wan, "Dnn partition and offloading strategy with improved particle swarm genetic algorithm in vec," *IEEE Transactions on Intelligent Vehicles*, 2023.
[17] L. Zhao, Y. Han, A. Hawbani, S. Wan, Z. Guo, and M. Guizani, "Media: An incremental dnn based computation offloading for collaborative cloud-edge computing," *IEEE Transactions on Network Science and Engineering*, 2023.
[18] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.
[19] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.
[20] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.
[21] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.
[22] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.
[23] I. J. Goodfellow, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
[27] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig *et al.*, "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.
[28] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" *Advances in neural information processing systems*, vol. 35, pp. 507–520, 2022.