PRIVATEER

Privacy-first Security Enablers for 6G Networks

Deliverable 3.1

Decentralised Robust Security Analytics Enablers Rel. A

DRAFT – Pending approval by the Smart Networks and Services Joint Undertaking (SNS JU)

Co-funded by the European Union

6G SNS

PRIVATEER has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101096110

Space Hellas SA, NCSR "Demokritos, Telefónica I&D, RHEA System SA, INESC TEC, Infili Technologies PC, Ubitech Ltd, Universidad Complutense de Madrid, Institute of Communication and Computer Systems, Forsvarets Forskninginstitutt, Iquadrat Informatica SL, Instituto Politecnico do Porto, ERTICO ITS Europe

PRIVATEER

Deliverable 3.1

Decentralised Robust Security Analytics Enablers Rel. A

Deliverable Type
Report/Other

Month and Date of Delivery
May 31st 2024

Work Package
3

Leader
INFILI

Dissemination Level
Public

Authors
Lampros Argyriou (INFILI)
Antonia Karamatskou (INFILI)

Programme	Contract Number	Duration	Starting Date
Horizon Europe	101096110	36 months	January 2023

Contact Us
info@privateer-project.eu

PRIVATEER has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101096110

Contributors

Name	*Organization*
Dimitris Santorinaios	NCSRD
Ilias Papalamprou, Dimitrios Danopoulos, Dimosthenis Masouros, Dimitrios Soudris	ICCS
Fábio Silva, Ricardo Santos	IPP
Mariana Cunha, João Vilela	INESCTEC
Martin Strand, Gudmund Grov, Markus Asprusten	FFI
Apostolos Garos	SPH

Reviewers

Name	*Organization*
Gudmund Grov	FFI
Martin Strand	FFI
Ricardo Santos, Fábio André Souto Da Silva	IPP

Copyright and Disclaimer

Version History

Version	Date	Modifications
1.0	31/05/2024	First Version

List of Acronyms

Acronym	*Description*
3GPP	Third-Generation Partnership Project
AES	Advanced Encryption Standard
AI	Artificial Intelligence
ART	Adversarial-Robustness Toolbox
CPU	Central Processing Unit
CTI	Cyber-Threat Intelligence
DDoS	Distributed Denial-of-Service
DL	Deep Learning
DP	Differential Privacy
DT	Decision Trees
EDA	Exploratory Data Analysis
FFT	Fast Fourier Transform
FL	Federated Learning
FPGA	Field-Programmable Gate Array
FPR	False-Positive Rate
GAN	Generative Adversarial Neural Network
GPU	Graphics-Processing Unit
IDS	Intrusion-Detection System
IoT	Internet of Things
KPI	Key Performance Indicators
KLD	Kullback-Leibler Divergence
LBS	Location-Based Service
LIME	Local Interpretable Model-agnostic Explanations
LPPM	Location-Privacy-Preserving Mechanism
LSTM	Long Short-Term Memory Networks
MAE	Mean-Absolute Error
ML	Machine Learning
MPC	Multiparty Computation
MSE	Mean-Squared Error
NCSRD	National Centre for Scientific Research Demokritos
NSSF	Network Slice-Selection Function
NWDAF	Network Data-Analytics Function
PCF	Policy-Control Function
PL	Planar Laplace
PPM	Privacy-Preserving Mechanism
PPML	Privacy-Preserving Machine Learning
PoC	Proof of Concept
RDP	Rényi Differential Privacy
RNN	Recurrent Neural Network

ROC	Receiver Operating Characteristic
SBA	Service-Based Architecture
SGD	Stochastic Gradient Descent
SMF	Session-Management Function
SHAP	SHapley Additive exPlanations
SOC	System-On-a-Chip
SVM	Support Vector Machine
TPR	True-Positive Rate
UE	User Equipment
UPF	User-Plane Function
WP	Work Package
XAI	eXplainable Artificial Intelligence
XR	eXtended Reality
ZTA	Zero-Trust Architecture

Executive Summary

The present deliverable documents the first release of the PRIVATEER security analytics modules, which have been researched and developed within Work Package 3 (WP3) "Decentralised Robust Security Analytics". The main objective of WP3 is to develop and integrate cutting-edge technologies to provide robust, privacy-preserving and trustworthy artificial intelligence (AI) security analytics algorithms, which can reliably detect potential cybersecurity threats in decentralised settings, such as Internet-of-Things (IoT) ecosystems. To safeguard the privacy and security of cloud-to-edge devices in a way applicable to the various PRIVATEER use cases, specific requirements and key performance indicators (KPIs) have been elicited. These are now being realised in the design and the architecture of the WP3 components. The use of artificial intelligence (AI) models for decentralised anomaly detection is a particular focus. This report reflects the current work progress in implementing the requirements and the advances achieved during the first period of the project.

In WP3, five main pillars promoting trustworthiness and privacy preservation in AI-based security analytics for decentralised and collaborative settings are exploited. The WP focuses on establishing and delivering robust components based on i) anonymisation of sensitive data, ii) trustworthy and privacy-preserving AI algorithm building, iii) robustness through adversarial hardening, iv) AI explainability, and v) optimal acceleration through appropriate hardware.

Within the 5G setting, which enables highly decentralised environments, the 5G Network Data Analytics Function (NWDAF) is distributed across the network continuum and gives insights into the network data production and consumption. Such data can be used to perform anomaly detection, and, by applying machine learning, provide adaptable and smart, near real-time security analytics. To emulate the context of 5G connectivity, specific NWDAF datasets have been generated by NCSRD and Space Hellas for use within PRIVATEER and its use cases. The streams of data exhibit features and specifications adhering to the 5G NWDAF standard.

In applying AI services in the cloud-edge/IoT continuum, suitable technological approaches for establishing trustworthiness and privacy preservation must be developed. The federated learning (FL) paradigm serves as an ideal starting point for the development of privacy-preserving AI algorithms in WP3. In FL, copies of a global model are sent to multiple clients, which hold their training data locally. A central server coordinates the FL process, collecting and aggregating locally trained models to update the global model without exchanging local input data. This decentralised approach reduces the influence of potential adversaries and preserves privacy by design, making FL suitable for applications in IoT and open networks like 5G and 6G.

Nevertheless, various threats to privacy and security persevere, which must be addressed appropriately through defence strategies to guarantee a secure operation. These risks must be taken into consideration when designing AI services in decentralised environments. Therefore, anticipating potential security vulnerabilities, additional security and privacy measures must be taken.

On the one hand, fruitful privacy-preserving techniques, such as differential privacy and secure multiparty computation, present themselves to protect collaborative computation environments from threats. These can be applied at various stages of the computation or aggregation processes. Such techniques may decrease computational accuracy and, thus, involve balancing utility and privacy, which is a significant challenge. Detecting potential adversarial behaviour relies on changes in client behaviour during downloads from or uploads to the server, distinguishing them from trusted clients. The latest version of the above-mentioned dataset with 5G Radio and Core metrics containing sporadic DDoS attacks has been analysed for this purpose, and an exploratory data analysis of the dataset features has been performed. A deep-learning model for anomaly detection on these time-series data has been developed and investigated. All these topics are part of the work within Task 3.2.

On the other hand, adversarial training can be employed to harden the models against vulnerabilities that can be exploited in various ways. Attacks can target the central server or local clients and lead to manipulation of the global model or inference of private information during the training and prediction phases. This includes membership and attribute inference. Because of the FL architecture and the aggregation mechanism, the effectiveness of certain threats, such as poisoning attacks, depends on the number of affected clients, which must be carefully analysed. The adversarial hardening is dealt with in Task 3.3. To minimise the probability of data breaches and successful data-related attacks, especially when private information is involved, anonymisation pipelines of sensitive or personal information are constructed and implemented, which is part of Task 3.1. Location data have been identified as sensitive and have consequently been anonymised via privacy-preserving methods.

Desirable properties of the AI models creating trust and security include trustworthiness, confidence, accountability, causality or fairness and ethical decisions. Therefore, in Task 3.4, explainability methods are investigated, which can add human-understandable explanation to the results produced by the security analytics. In this way, conclusions regarding the above-mentioned properties can be drawn and an explanation can be provided for the cyber threat intelligence decision that involves machine-learning algorithms. Finally, the edge nodes are equipped with appropriate hardware acceleration mechanisms to optimise the energy and time consumption of the AI models, which is part of Task 3.5.

Contents

1 Introduction

In the evolution from fifth generation (5G) to the forthcoming sixth generation (6G) networks – which promise intelligent and autonomous technologies – security, safety and trustworthiness in artificial intelligence (AI) and related services become increasingly crucial [1]. Both 5G and 6G hold the potential for significant transformation in the way users and stakeholders interact with network technologies by laying the ground for new services through unprecedented data speeds, ultra-low latency, and ubiquitous connectivity. These enable real-time data processing and rapid communication among devices.

However, with these advancements serious challenges arise regarding security and privacy [2]. The expansive network infrastructure, which supports applications such as IoT, edge computing and AI-assisted services for autonomous vehicles and smart cities, creates progressively more complex ecosystems that are susceptible to cyber threats and privacy breaches. The expanding ubiquitous computing capabilities, increased user participation and the growing demand for user and network-usage data raise privacy and security concerns. For these reasons, special attention must be paid to ensure robust security measures and trust in automated AI-driven services that will be integrated into the 6G ecosystem services. Anticipating the challenges described above, the PRIVATEER project tackles the security and privacy threats arising from the collection of large amounts of data by 5G/6G components and services by providing a privacy-first, decentralised approach to machine-learning-based security analytics.

The present report describes the research, findings and implementation of the work done in Work Package 3 (WP3) of PRIVATEER. The document is structured as follows:

Chapter 1 provides an introduction to motivation, privacy and security concerns, and anticipated challenges.

In Chapter 2, we describe the overall structure of WP3 and the development setting. Furthermore, the NWDAF and the dataset generated for PRIVATEER containing a DDoS attack are presented. This dataset is uploaded on Zenodo, however we have prepared a revised version where some parameters have been updated.

Chapter 3 is dedicated to the five components comprising the trustworthy and robust security analytics enablers of PRIVATEER with their functionalities. Each module is described in detail, presenting objectives, state-of-the-art and work plan. Firstly, in Task 3.1 the strategy for the anonymisation pipelines of sensitive data is presented. Secondly, the development of trustworthy federated security-analytics models within Task 3.2 is described. Thirdly, the approach in Task 3.3 for enhancing the robustness of these models against adversarial actions and corresponding hardening techniques is provided. Fourthly, strategies for providing explanations for the models' results and

the corresponding AI-model development mechanisms of Task 3.4 are introduced. Finally, the integration of suitable hardware-based acceleration of the AI models in Task 3.5 for resource- and energy-efficient operations of the federated-learning models is presented. Tasks 3.1 and 3.2, which started in month 9, present a detailed approach and first results in architecture and computation. Chapter 4 concludes by summarising the findings so far and provides an outlook on the WP3 developments within the second phase of the PRIVATEER project.

1.1 Motivation

The extreme interconnectivity that is expected to govern future technologies by linking a vast number of devices requires a detailed analysis of the threat landscape. The analysis of the potential threats and possible mitigation strategies performed within the PRIVATEER project can be found in deliverable D2.1 “6G threat landscape and gap analysis” [3]. Based on this analysis, the technologies that the PRIVATEER use cases will rely on focus on security and privacy preservation. Regarding the development of AI algorithms for security-analytics purposes, the distributed setting of interconnected devices poses challenges, but also holds opportunities. The IoT setting and the consequential potential threat vectors and intrusion points govern the design of secure and robust security-analytics enablers [4].

On the one hand, trustworthiness in AI-driven services within the 5G or 6G landscape requires secure, robust, transparent and accountable decision-making processes, free from errors and biases [5]. On the other hand, to establish trust among stakeholders and users, the privacy of sensitive data stored or transmitted through the networks must be safeguarded as a key priority.

Both above-mentioned strong requirements for the 6G technologies comprise key motivations of WP3. The goal of WP3 is to develop privacy-preserving and trustworthy AI algorithms for adaptable, real-time security analytics of services running on (5G/6G) networks and offering IoT solutions, such as automated driving or smart-home systems. PRIVATEER’s approach relies on the federated-learning (FL) principle [6], which by design federates both data and computation and removes the necessity of duplication or transfer of data. An important cornerstone of the work within WP3 is to anticipate security threats within a decentralised setting characteristic for IoT environments and design suitable defence strategies while maintaining the performance and functionality of the computational setting. Techniques for guaranteeing privacy and security are researched and added to this design, while the models are hardened against potential attacks through adversarial training. The decentralised security-analytics architecture is enriched with data-anonymisation pipelines for sensitive and private data, or data from which private information can be inferred, and, moreover, suitable hardware-acceleration methods. The AI algorithms

are further enhanced with explainability techniques to satisfy important trustworthiness requirements, such as explainability, fairness and robustness.

1.2 Privacy and Security Concerns in AI Services in 6G

In the landscape of 6G networks, AI-based services promise to be a fundamental technology. Through machine learning (ML) and deep learning (DL), the realisation of 6G-connected intelligence moves closer. These technologies will enhance performance, energy efficiency and security across various layers of applications. Distributed storage and processing, along with federated ML are crucial for computational efficiency and will improve security. However, federated AI also opens the door to potential attacks, necessitating constant surveillance, verification and automatic repair mechanisms of AI systems against malicious attacks [7]. To address these challenges, privacy-preserving schemes, including ML-based techniques, aim to safeguard user privacy, especially in the edge and control layers which are prone to data poisoning and attacks on AI models. FL emerges as a prominent privacy-preserving approach, but it also faces potential attack surfaces. Techniques such as differential privacy (DP) [8] and multiparty computation (MPC) [9] hold the potential to enhance privacy within the 6G framework.

In the following, the attack surfaces of FL are described in more detail, since it offers a valuable starting point as a distributed ML architecture.

Being a truly decentralised machine-learning approach, FL preserves privacy by keeping data locally as mentioned above. The model training occurs across multiple local clients which send the model parameters for global aggregation to a central server. Thereby, FL might be susceptible to malicious attacks at different stages, namely through data gathering, training and inference. These attacks can target both the central server and the local clients, compromising the integrity and privacy of the FL process. Therefore, effective measures must be implemented to secure FL across all phases.

On the client side, contaminated data and malicious behaviour can impact the model training and the global model's performance, although the impact of this is reduced due to the federated nature of the computation process. Interception attacks, like eavesdropping during model updates, necessitate secure-transfer methods like encryption or DP. Evasion attacks can alter the outcomes of inference, while privacy-inference attacks extract sensitive information from the model. There exist defence strategies for these attacks, which will be discussed in section 3.3. Poisoning attacks aim to influence the model performance by manipulating local clients which can be mediated through techniques such as DP and secure aggregation. During inference, evasion attacks produce adversarial examples to deceive the model, while privacy-

inference attacks target model inversion, membership inference and model extraction. Obfuscation techniques present defence strategies, while the utility-privacy trade-off must be taken into consideration when applying these techniques. MPC with DP allows collaboration of different parties without revealing individual inputs which creates a trusted aggregator while preserving privacy.

In general, adversarial attacks are techniques where only slight modifications to the input data can cause ML models to make incorrect predictions, even though the changes are imperceptible to humans. These attacks are a significant concern as they can compromise the reliability of AI systems. Adversarial training is one method used to defend against such attacks, where models are trained with adversarial examples to improve their robustness. This will be implemented within Task 3.3 in PRIVATEER. These strategies collectively contribute to enhancing the security and reliability of machine-learning models against adversarial attacks.

Different types of attacks are considered as anticipated threats within the WP3 developments and will be investigated with respect to their success and detectability during adversarial hardening and re-training of the anomaly-detection models in Task 3.2. Moreover, the effect of the obfuscation techniques on the quality and performance of the models will be thoroughly examined. Chapter 3 provides a detailed analysis and results achieved so far.

1.3 Challenges

The rapid advancements in AI-based technologies come with a variety of challenges, both regarding technological and societal aspects [1]. The trustworthiness of software-related services always depends on the technical reliability, but also on the societal acceptance. Especially the implementation of AI-assisted services, which depend on vast amounts of data, relies on trust and robustness. Therefore, providing comprehensibility, fairness and explainability of automated and interconnected services is paramount.

With the proliferation of IoT devices and the anticipated growth in the number of connected devices in 6G networks, the scale and complexity of network infrastructures will increase extremely [10]. Analysing security-related data from such massive networks in real time requires highly scalable and efficient security-analytics solutions.

Another potential challenge lies in the latency requirements, as 5G and 6G networks aim to solve exactly this problem by delivering ultra-low latency communication. Effective security-analytics solutions must meet these latency requirements while providing timely detection and response to security threats. Due to the time-series character of network data, ML-based anomaly detection that relies on sliding windows

or time batches must be adapted carefully to the network-data transfer frequency. This is an important parameter for the model development in Task 3.2.

The most serious challenge presents itself in the security, privacy and data-protection aspects [2]. Decentralised security analytics analyse data from distributed sources, which raises concerns about privacy and data protection. This deliverable deals with a set of measures that are suitable to protect data and privacy in order to guarantee a level of privacy and security for the services running on the network. The privacy-utility trade-off is of central importance in this respect and must be thoroughly balanced to achieve optimal performance along with data accuracy while guaranteeing certain levels of privacy. This will be tackled in WP3 by detailed computational experiments within the second phase of the project.

2 WP Architecture

2.1 WP Structure

WP3 focuses on facilitating security analytics in a decentralised, privacy-preserving manner by delivering all essential components needed from data handling to real-time detection of anomalous behaviour. The key activities include the creation of anonymisation pipelines to selectively anonymise data for AI security analytics in Task 3.1. Additionally, WP3 aims to develop trustworthy AI models optimised for distributed learning and inference, ensuring reliability and integrity across decentralised systems. The evaluation and hardening of AI models against adversarial attacks are of paramount importance and are taking place in a feedback loop between Tasks 3.2 and 3.3. Moreover, on top of the security-analytics models, the implementation of mechanisms for decision support through Explainable AI (XAI) in Task 3.4 enhances the comprehensibility of the AI results. Finally, the integration of hardware-based accelerators in Task 3.5 enhances the efficiency of decentralised, AI-computational operations, optimising energy utilisation for improved performance.

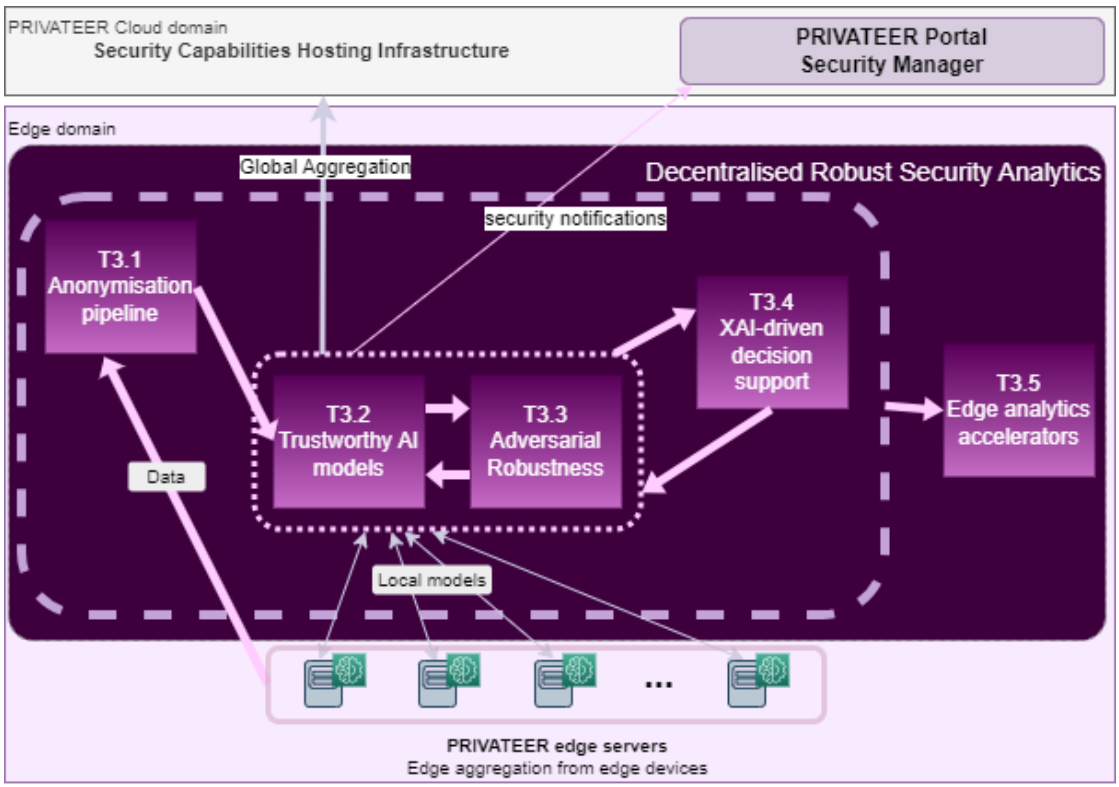

Figure 2.1 WP3 architecture and task interdependence

The architecture of the Decentralised Robust Security Analytics of WP3 is depicted in Figure 2.1. The five technology components corresponding to the five tasks 3.1–3.5 are shown, and their interdependence is underlined. The embedding of the decentralised modules in the edge domain is highlighted by the fact that the data, which are anonymised in Task 3.1 and used as input for the anomaly-detection in Task 3.2, is hosted at the edge nodes. For PRIVATEER's purpose, the edge nodes perform a first aggregation from the user equipment (UE) devices. Local models are computed at the edge nodes and aggregated at the central PRIVATEER server. Task 3.2 focuses on enhancing security analytics through decentralised AI and privacy-preserving techniques, while Task 3.3 tests these models against adversarial attacks using tools like Generative Adversarial Networks (GANs). This synergy creates a robust feedback loop, with Task 3.2 strengthening defences and Task 3.3 refining robustness through continuous testing, ensuring the models are effective and resilient against threats. The resulting algorithms for security analytics are consequently input into the explainability module of Task 3.4. Finally, the hardware acceleration of the algorithms in Task 3.5 takes place for optimal performance. Detected anomalous behaviour is reported via security notifications to the PRIVATEER Security Manager for further analysis or cyber-threat intelligence (CTI) sharing which is developed in PRIVATEER under WP5.

2.2 Development Setting

2.2.1 5G Network Data Analytics Function

The Network Data Analytics Function (NWDAF) was introduced as part of the 3rd Generation Partnership Project (3GPP) Release 15 specification, representing a shift towards data-driven and proactive cellular networks. The NWDAF is designed to aggregate and analyse data from various sources within the 5G network. These data include performance metrics, user behaviour, and network conditions. The insights derived from this analysis are used to optimise network performance, manage resources more efficiently, and improve user experiences.

Figure 2.2 Schema of NWDAF integration into the SBA of 5G

The NWDAF is integrated into the Service-Based Architecture (SBA) of 5G and interacts with other network functions via standardised interfaces, as depicted in Figure 2.2. This integration allows it to collect data from various network elements and provide analytics services to other functions like the Network Slice Selection Function (NSSF) or the Policy Control Function (PCF).

By leveraging advanced data analytics and ML algorithms, the NWDAF can predict network demands, detect anomalies, and provide recommendations for network configuration changes. NWDAF supports a range of use cases, from network security (by identifying and responding to threats) to quality-of-service optimisation (by predicting and mitigating congestion). As a result, it is considered as a significant enabler of advanced 5G use cases like network slicing for diverse service requirements and edge computing for low-latency applications, including extended reality (XR) and automotive use cases.

In PRIVATEER, the NWDAF is instantiated as a multi-container application designed for robust data collection and processing. This application integrates several widely used technologies such as Apache Kafka, ZooKeeper, and InfluxDB, each serving a specialised function within the system.

The application comprises the following components:

- **ZooKeeper** is utilised to manage coordination and configuration for Kafka, a necessity in distributed systems to maintain robustness and reliability. It is isolated within a custom Docker network and opts out of logging to streamline operations.
- **Kafka** serves as the central message broker, handling real-time data feeds essential for dynamic data processing environments. It directly depends on ZooKeeper for operational management and is configured for external communications. Kafka’s environment settings are tailored to facilitate both internal and external communications through specified listeners and to automatically manage topics. This configuration enhances Kafka's integration within the networked application.
- **InfluxDB** is a time-series database ideal for efficient storage and retrieval of time-stamped data. A time-series database is a type of database designed to handle time-stamped data - data that changes over time or is sequentially timestamped. This kind of database manages and stores sequences of values that are tracked, monitored, and aggregated over time. Time-series databases are optimized for handling large volumes of data that are typically written in a chronological order. The InfluxDB maps a local directory to the container's storage location to ensure data persistence.
- **Data Producer and InfluxDB Connector** are additional custom components built to integrate seamlessly with Kafka and InfluxDB. The Data Producer is tasked with feeding data into Kafka, while the InfluxDB Connector is designed

to facilitate data transfer between Kafka and InfluxDB. Both components are crucial for the end-to-end data handling capability of the system, ensuring that data not only flows efficiently but is also effectively processed and stored.

The entire system is interconnected through the Docker network, which supports secure communications between the containers. The configuration is optimised for scalability and reliability in a production environment, focusing on high availability and robust data management.

2.2.2 The NCSRD-DS-5GDDoS Dataset: 5G Radio and Core Metrics Containing Sporadic DDoS Attacks

PRIVATEER Partners have been working on releasing open-source datasets from the 5G+ infrastructure located in NCSRD, which will also be the testbed for the use case scenarios deployment for Release B of the project. The NCSRD-DS-5GDDoS dataset [11] is published on Zenodo and will be updated throughout the project, based on feedback from PRIVATEER partners and external partners with strong interest on AI/ML applications on 5G cybersecurity.

This is a comprehensive dataset recorded in a real-world 5G testbed that aligns with the 3GPP specifications. The dataset captures Distributed-Denial-of-Service (DDoS) attacks initiated by malicious connected User Equipment (UEs). The setup comprises of two cells with a total of nine UEs connected to the same core network. The 5G network is implemented by the Amarisoft Callbox Mini solution, and we further employ a second cell using the Amarisoft Classic, that also hosts the 5G core.

The setup utilises a broad set of UE devices comprising a set of smart phones (Huawei P40), microcomputers (Raspberry Pi 4 - Waveshare 5G Hat M2), industrial 5G routers (Industrial Waveshare 5G Router), a WiFi-6 mobile hotspot (DWR-2101 5G Wi-Fi 6 Mobile Hotspot) and a CPE box (Waveshare 5G CPE Box). All UEs are being operated by subsidiary hosts which are responsible for the traffic generation, occurring from scheduled communications times.

All identifiers are artificially generated and neither represent nor are based on personal data. We identify each UE through its ‘imeisv’ ID, that corresponds to the device in use, due to vendor implementation, that uses the same IMSI for all UEs. There are eight attacker UEs in the testbed and one benign user with imeisv = 8609960480666910. The benign user streams YouTube traffic, while the malicious users are performing two DDoS attacks (UDP floods using hping3); the first attack takes place on 24-01-2024 between 14:48:30-14:58:30 and the second one on 25-01-2014 between 14:05:00-14:10:00. Throughout the recording session, handover events also take place.

The dataset is populated using the data collector previously outlined, which interfaces with the 5G network to collect information about UEs, gNBs, and the Core Network. The data are recorded in an InfluxdB and pre-processed into three separate tabular CSV files for more efficient processing: “amari_ue_data.csv”, “enb_counters.csv” and “mme_counters.csv”.

3 Robust, Decentralised Security-Analytics Components

This chapter introduces the five technical components of the WP which provide PRIVATEER's decentralised security analytics in more detail. Every section corresponds to one task of the WP. We present first research and implementation results along with privacy and security considerations related to the risks and threats described above.

3.1 Anonymisation pipelines

3.1.1 Objectives

The main objective of anonymisation pipelines (Task 3.1) is to provide methods for privacy analysis and protection of sensitive data types considered in PRIVATEER components. In the first stage, this task was focused on gathering information about privacy-sensitive data types and corresponding privacy and utility requirements. Upon the identification of sensitive data types, the focus is on selecting and/or developing appropriate Privacy-Preserving Mechanisms (PPMs) that fulfil the privacy/utility requirements, while considering potential attacks and defining metrics to quantify the attained privacy and utility levels. In order to standardise this process, the anonymisation pipelines will be available as a toolkit of PPMs for heterogeneous data types. This toolkit is designed in a modular manner and can act as a privacy-aware pre-processing stage for data-driven components in the PRIVATEER framework, such as those from Tasks 3.2 and 3.4. In this way, the anonymisation pipelines warrant proper privacy protection for data identified as personal and/or sensitive before making it available to the security analytics models.

3.1.2 State of the Art

Although privacy has been recognised as a human right since 1948, there is a lack of a standardised and universal privacy definition. The growing collection of large amounts of data has been raising serious privacy concerns, which is a call for Privacy-Preserving Mechanisms (PPMs) [12].

In the context of 6G technologies, all sorts of localisation-based analytics methods envisioned reveal potential threats related to location exposure and position tracking [3]. To address these risks, location privacy has become an emerging topic of research with the proposal of several Location Privacy-Preserving Mechanisms (LPPMs). The

existing LPPMs are commonly divided into the following approaches: anonymisation, obfuscation, reducing location sharing, and cryptography-based. The anonymisation approaches typically rely on traditional notions (e.g. k-anonymity) to protect the users' identity, whereas the obfuscation approaches are mainly focused on protecting the position of the users. To do so, various methods have been proposed either to obfuscate locations in terms of spatial, temporal, and spatiotemporal characteristics or to perturb the locations by adding noise (e.g. geo-indistinguishability).

Due to the simplicity of implementation, efficiency, and effectiveness, geo-indistinguishability has been considered the state of the art by the research community. Geo-indistinguishability is a formal notion based on Differential Privacy (DP) that guarantees that any two points within a given radius centred at the user's location are statistically indistinguishable independently of the adversary's background information [13]. For a detailed explanation on DP, see Section 3.2.2.3. The Planar Laplace (PL) mechanism was the first proposed mechanism to satisfy the notion of geo-indistinguishability applied to the context of Location-Based Services (LBSs). The PL mechanism consists of adding 2-dimensional Laplacian noise centred at the exact user location with a probability density function that satisfies ε-geo-indistinguishability. This probability of generating an obfuscated location from any two points x, x' is bounded by the distance between these two points factored by the privacy budget ε, commonly set by the user. This privacy parameter ε is typically defined as $\varepsilon = l/r$, being r, the radius defined by the user, where the privacy level l is guaranteed.

However, preserving a certain level of privacy can come at the expense of data utility, which makes the selection of the proper PPM quite challenging. Despite the necessity of identifying the data type that needs to be protected, privacy protection can occur at different times of the data lifecycle [14]: data collection, data publishing, data distribution, and at the output of data mining. Furthermore, each stage might have specific constraints (e.g. data collection can be either sporadic or continuous) that should be taken into consideration by PPMs. To fulfil the identified requirements, PPMs might then require a trusted third party or may not.

In addition to the selection of the PPM, a proper configuration of the mechanisms is also crucial since a misconfiguration can lead to an ineffective privacy/utility level [15]. Thus, there is a need to define the appropriate metrics to assess the attained privacy and utility levels. The development of the privacy toolkit within Task 3.1 aims at standardising this process of selecting, configuring, and evaluating PPMs to warrant appropriate privacy protection for data that is identified as personal and sensitive, before making it available to the security-analytics models.

3.1.3 Work Plan

3.1.3.1 Identification of Threats

The ubiquitousness of smart and mobile devices has led to a growing collection of enormous amounts of data. Although the benefits that advent from collecting data both to the users and to the service providers, exposing sensitive information might pose serious threats to privacy. Within the context of 6G networks, the collection of user information, including location data, is relevant to predict user mobility and reduce latency of edge-services handover, as well as behaviour tracking/prediction to find abnormal patterns in user equipment of traffic behaviour. Despite the enhancements in connectivity and in the location precision, location exposure and position tracking are considered privacy challenges that need to be addressed [3].

Location privacy is an emerging topic of research due to the sensitive nature of this data type. The potential threats of exposing location data go beyond physical security since location data can reveal users' identity, routines, habits, or even health conditions [16]. Thus, this task will be focused on protecting location data, identified as a sensitive data type within the PRIVATEER project.

3.1.3.2 Approach and Architecture

The lack of a standardised method for privacy analysis as well as the demand for protecting sensitive types considered in PRIVATEER components led to the development of a toolkit of PPMs [17]. This toolkit is designed to follow a modular and extensible approach, thus allowing to act as a privacy-aware pre-processing stage for data-driven components in the PRIVATEER framework.

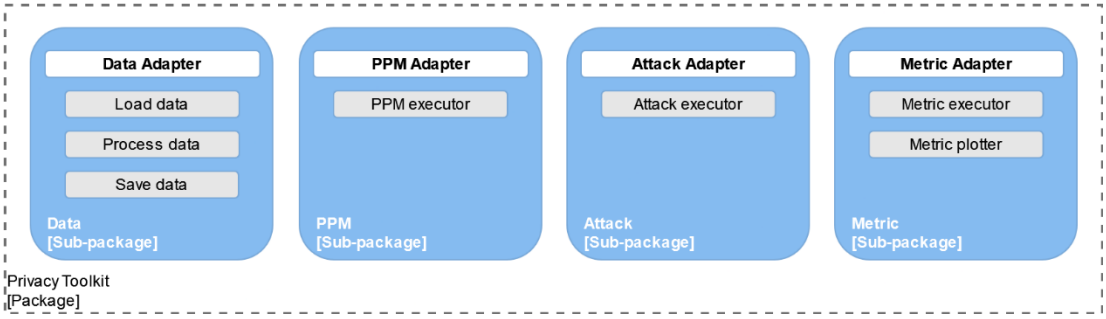

Figure 3.1: Architecture overview of the privacy toolkit.

The main objective of this toolkit is to apply PPMs, test configurations, and assess mechanisms according to the attained privacy and utility levels of data. Similarly to other well-known scientific toolkits, this toolkit will be available as an open-source Python package. Figure 3.1 provides an overview of the package architecture. Commonly, an anonymisation pipeline is composed of the following steps: input data, PPM, attack, privacy/utility metrics, and output data. Concerning the input data step,

beyond loading, data might need to be processed before applying the PPM. The PPM is then properly configured and applied to the resulting input data, generating an anonymized version of the data as output. The assessment of the achieved privacy/utility level relies on the output data, suitable privacy/utility metrics and, optionally, on an attack whose goal is to estimate the original data through the anonymized version.

To systematise this process, each step of the mentioned pipeline constitutes a sub-package as represented in Figure 3.1. Each sub-package contains an adapter corresponding to an abstract class that can be extended by implementing the abstract methods (i.e., relevant methods for the component). These adapters foster the implementation of new features (e.g., new PPMs or metrics) while maintaining the source-code structure and readability. The sub-packages presented in Figure 3.1 can be briefly described as follows. The data sub-package is responsible for handling a data type by providing methods such as the ones represented in grey: load data, process data, and save data. For adding new data types, the Data Adapter can be extended with the implementation of the corresponding methods and other methods of relevance for the new data type. The remaining sub-packages follow a similar approach by defining the corresponding adapter. Thus, for adding a new PPM, attack, or metric, the respective adapter is extended, the executor is implemented, as well as other desired methods.

Within the PRIVATEER framework, location data was identified as a sensitive data type that needs to be protected. Towards this goal, the toolkit is designed to include implementations of appropriate PPMs, attacks, and metrics in this context. Nevertheless, due to its extendibility, the privacy toolkit is expected to support heterogeneous data types, as well as different types of PPMs, attacks, and metrics that are suitable for the needs of the PRIVATEER framework.

3.1.3.3 Current Status and Next Steps

Recalling the objectives of Task 3.1, this task started with the identification of sensitive data types from PRIVATEER, which has been accomplished in accordance with the current version of the dataset. From the performed analysis, location data was identified as a sensitive data type and will be the focus of data protection.

As mentioned before, the anonymisation pipelines will be standardised through a toolkit of PPMs. This toolkit is designed in a modular and extensible manner [17], including PPMs implemented to protect data, as well as metrics, and attacks. The current version of the toolkit implements location data as data type by extending the Data Adapter, the Planar Laplace (geo-indistinguishability) as a PPM, and the quality loss as a metric to assess the utility level achieved after applying the PPM. This

assessment is quantified by the distance between the obfuscated location that was generated by the PPM and the exact user location.

To foster the anonymisation pipelines in a wide manner, this toolkit is also accessible through a Web application. The Web version facilitates the selection and configuration of PPMs through a more interactive interface, while providing a better understanding of the privacy parameters under configuration. The performed configurations can then be downloaded to be executed locally in a larger sample of data. For illustration, Figure 3.2 demonstrates the configuration and application of the Planar Laplace (PL) mechanism in an interactive manner, which allows us to modify the privacy parameter ε, while showing the corresponding impact on the obfuscation radius. In addition, recalling that PL is designed to single queries and, hence, that the privacy level linearly scales with the number of queries [13], this example also allows us to visualize the privacy degradation that occurs when applying the mechanism to traces. This privacy degradation comes from the multiple applications of the protection mechanism.

The next steps of this task are the identification and analysis of sensitive features, and the development of PPMs to protect such sensitive features. Furthermore, this toolkit can also be integrated with other project components to provide a pre-processing stage for the data-driven components and enable assessment of the impact of privacy-protection methods.

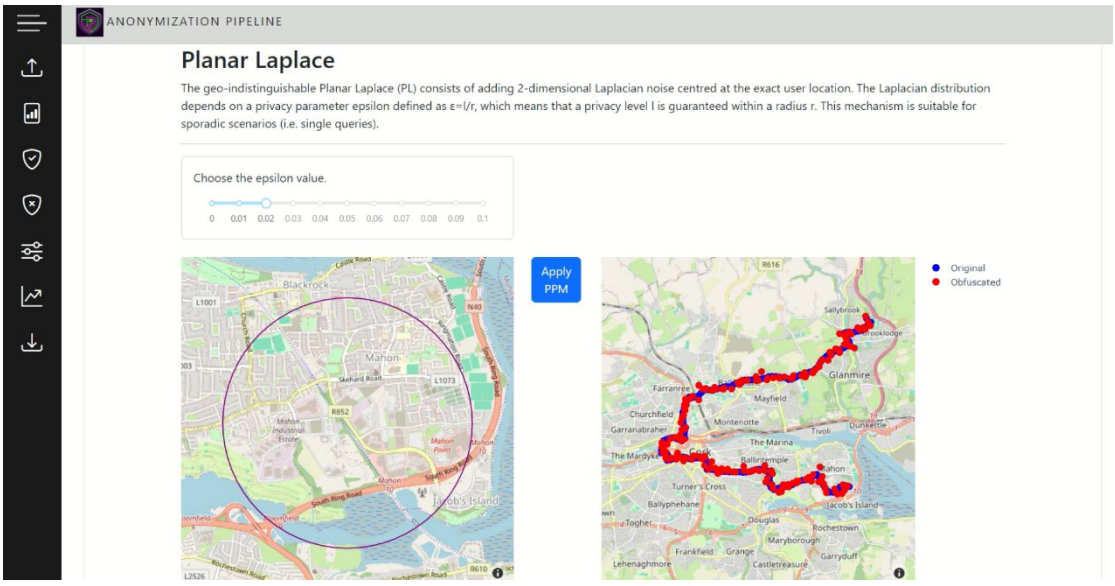

Figure 3.2: Web application screenshot for the application of Planar Laplace privacy-preserving mechanism.

3.2 Trustworthy AI model building

3.2.1 Objectives

Task 3.2 aims to enhance security analytics using decentralised AI, focusing on anomaly detection and threat classification. It leverages Privacy-Preserving Machine Learning (PPML) techniques to safeguard user data, incorporating data from the 5G NWDAF function for improved insights. The task seeks a balance between privacy, performance and fairness in AI training and emphasises robustness through adversarial training.

The following outlines the specific objectives of Task 3.2:

- Develop sophisticated decentralised artificial intelligence models specialised in anomaly detection and threat classification to augment the security-analytics capabilities of the PRIVATEER framework.
- Implement advanced PPML techniques during the training phase of these AI models to ensure privacy guarantees for PRIVATEER users. This approach aims to secure sensitive data against unauthorised access or inference, providing a solid privacy framework within the analytics operations.
- Address the privacy-performance trade-off by devising optimisation strategies that ensure the efficient processing of data while upholding strict privacy standards, ultimately aiming to maximise the effectiveness of security analytics without compromising the privacy of PRIVATEER users.
- Integrate results from Task 3.3 regarding adversarial-training paradigm to reinforce the security and trustworthiness of the AI models. This involves training models to withstand malicious attempts to deceive or bypass the security analytics, thereby enhancing the resilience and reliability of the threat detection mechanisms.
- Engineer and execute an effective alarm management framework capable of aggregating, analysing, and refining preliminary security alerts with corroborative information from external sources. This strategy aims to optimise the accuracy and relevance of security notifications, reducing false positives and ensuring timely response to genuine threats.

In the following, we present the current state of the art for technologies that have been identified as crucial to the PRIVATEER project. Within PRIVATEER these technologies will be combined and tailored to the needs of preserving privacy and security within decentralised settings reflected in the PRIVATEER use cases.

3.2.2 State of the Art

3.2.2.1 Network Anomaly Detection

In the era of increasing digital interconnectivity, spanning traditional networks to the next generation of Software-Defined Networks and the Internet of Things (IoT), network security has become a paramount concern. The interconnected nature of these digital systems amplifies the potential impact of cyber-attacks, exposing a vast array of systems, services, and stakeholders. Despite these risks, security experts often find it challenging to stay ahead of the rapidly evolving landscape of cyber threats, as criminals devise more sophisticated attacks [18].

While traditional security measures such as firewalls, encryption methods, and anti-virus software remain essential, they alone are not sufficient to address the novel and complex threats that characterize today's security landscape [19]. Intrusion-Detection Systems (IDS), which serve as an additional line of defence by monitoring and alerting on suspicious activities, traditionally rely on signature-based detection. However, the relentless evolution of cyber threats necessitates constant updates to IDS signature databases, a process that is both time-consuming and increasingly ineffective. On top of that, the widespread use of encryption means most traffic bypasses traditional deep packet inspection, further reducing the effectiveness of these signatures.

In response to these challenges, the industry is increasingly adopting zero-trust architectures (ZTA), which require continuous verification and adaptive defence mechanisms. Complementing this, advanced machine learning-based detection techniques are utilized to enhance ZTA by providing swift and precise identification of potential security threats. These innovations aim to not only accelerate the detection and mitigation of threats but also to improve the overall robustness of network systems against the sophisticated cyber-attacks of the modern era.

The taxonomy of ML-based IDS is quite varied. Such advanced IDS encompass a range of approaches, from supervised-learning techniques that utilise labelled datasets for training models to recognise known threats, to unsupervised learning algorithms adept at detecting new, previously unseen anomalies by analysing patterns in network traffic [20].

Traditional approaches take ML as a common practice, with k-means clustering, isolation forests and one-class support vector machines (SVM) being the main approaches [21].

Apart from traditional ML algorithms, DL models have been extensively used for anomaly-based intrusion detection. Based on recent research, various DL architectures have shown promising results in network-anomaly detection, each with unique strengths and challenges [22, 23]. Generative Adversarial Networks (GANs), for example, utilise a dual-network architecture comprising a generator and a

discriminator. The generator creates data-mimicking normal traffic, while the discriminator works to distinguish between this synthetic data and real network data. This setup allows GANs to effectively identify anomalies as the generator evolves to produce increasingly accurate synthetic data. However, they face challenges with training stability and high computational demands.

Additionally, Long Short-Term Memory Networks (LSTMs) and other Recurrent Neural Networks (RNNs) are effective for data that involve sequences, such as time series or continuous network traffic. These networks excel at learning dependencies and patterns over time, which is crucial in dynamic environments where anomaly patterns can evolve and change.

Lastly, autoencoders are employed for unsupervised-learning tasks – ideal in situations where labelled data are scarce. They detect anomalies by reconstructing input data and identifying instances where the reconstruction error is unusually high, signalling an anomaly. This method is particularly advantageous in environments where anomalies are rare and not well-defined, allowing for detection without prior labelling of data.

Together, these DL methods form a comprehensive toolkit for network-anomaly detection, each contributing differently to address the complexities of monitoring and securing modern networks.

The rapid evolution and adoption of 5G networks have gained considerable attention due to their enhanced speed, lower latency, greater capacity, improved reliability, and reduced energy consumption. This technological leap is also influencing the realm of AI-based malicious-traffic detection, with a growing focus on adapting these models to the unique architecture and scale of 5G networks. Unlike its predecessors, 5G represents a highly flexible infrastructure, organised around various functional modules, presenting new challenges not previously encountered. Among these, the most pressing issues include the need for high-performance processing capabilities within detection models to handle increased network-traffic volumes, ensuring data security and privacy, and maintaining robust detection capabilities in a more complex network environment. As 5G networks enhance throughput and capacity, AI models must not only manage larger volumes of data but also match this performance in real-time processing scenarios.

These challenges underline the need for innovative solutions that enhance the speed and efficiency of anomaly detection in the evolving landscape of 5G networks.

3.2.2.2 Federated Learning

As described above, federated learning (FL) which was introduced by McMahan et al. in 2017 [6] serves as a good starting point for privacy-preserving and decentralised

computation. The technique allows each node to train a model locally in contrast to submitting the complete dataset to a central node. After training, each party submits their updated model to an aggregator, which averages all models into a unified model, which is again distributed to the parties. The algorithm was termed *federated averaging* (FedAvg). The server receives weights w_{t+1}^k from K clients, and computes the update

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k . \tag{3.1}$$

This process continues until the model converges. Note that this is a linear combination of the client weights with the client dataset sizes n_k and total size n.

McMahan et al. suggested FL as a privacy-preserving effort, as private data would no longer leave the data owner. However, while the information is obscured, FL – or indeed ML itself – gives no formal guarantee that the model does not leak private information.

3.2.2.3 Differential Privacy

To get such guarantees, we need to turn to dedicated definitions. The concept of differential privacy (DP) [8, 24] parametrises the acceptable privacy loss, and it is well-known how to instantiate mechanisms that satisfy the definition for high levels of privacy. We first give the mathematical definition, and then explain the intuition it conveys.

Let *M* be an algorithm that takes a dataset *D* as input, and let *S* be a subset of all possible output from *M* (known as the *image* of M and denoted $\mathrm{im}\, M$). Let D_1 and D_2 be two datasets that differ in a single datapoint. The algorithm *M* gives (ε, δ)-DP if for all pairs D_1, D_2 and all $S \subseteq \mathrm{im}\, M$,

$$\Pr[M(D_1) \in S] \leq \exp(\varepsilon) \Pr[M(D_2) \in S] + \delta.$$

The original definition of ε-DP used $\delta = 0$, which makes it easier to grasp. The definition then states that if you choose a set *S* that corresponds to a particularly interesting property, then the probabilities of *M* producing a model that reveals that property is approximately the same regardless of whether one uses the dataset D_1 or D_2, where the former may contain a datapoint with this property whereas the latter may not.

This again implies that the property will only be visible from the model if sufficiently many datapoints contribute to it: if the model suggests that a house could be painted with zebra stripes, then you can be sure that this is not only because *your* house has this decoration.

A smaller ϵ gives better privacy but may make the model less accurate. On the other hand, the presence of DP may also reduce a tendency to overfitting [25].

Experience shows that the original ϵ-DP definition is too strong in the case of non-likely events. Therefore, the *δ-approximate* definition has been suggested [26], which relaxes the definition slightly.

One way to apply DP is for each participant to ensure that all vectors have a bounded norm and clip any that are too large [27, 28].

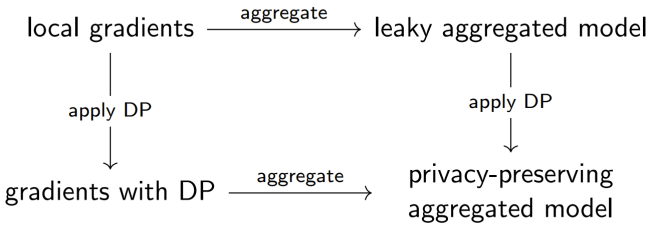

Figure 3.3: Two ways of creating a privacy-preserving aggregated model from local gradients.

This gives us a way to create a privacy-preserving aggregated model: each participant applies DP to their gradients, and the central server aggregates these models. This corresponds to starting in the top left corner of Figure 3.3, and following the arrow down and then right. However, this has a substantial drawback. As each participant adds noise to the model, this noise will grow during aggregation, which results in a less accurate model.

3.2.2.4 Multiparty Computation for Secure Aggregation

We will now explore the other path in Figure 3.3: First aggregate, then apply DP. The benefit of applying DP after the aggregation is that the noise is only added once, and we should, therefore, get a more accurate model.

Clearly, we can't delegate this to the untrusted aggregation service, as the gradients and the resulting model would leak private information. To solve this problem, we let (a subset of) the distributed parties cooperate to emulate the aggregator. This is feasible using secure multiparty computation (MPC) [9].

Theoretically, any function can be computed using MPC. The security guarantee is that any adversary may only learn as much as he would from their input and output. In this sense, MPC can be compared to a perfect pre-programmed black box to which all participants privately send their input. Eventually, the box will output the result of the computation.

In practice, MPC favours "simple" computations. Sums and scalar multiplications are essentially free of additional cost, whereas multiplications may require additional

rounds of communications. More advanced operations, such as divisions or checking (in)equalities require even more work and communication.

Recall Equation 3.1 and the observation that it essentially is a linear combination. This gives us three possibilities:

1. The parties wish to keep their updates private. However, they are willing to disclose the size n_k of their datasets. Notice that $n = \sum_{k=1}^{K} n_k$. Hence, we only need to perform K scalar multiplications and sums in MPC to aggregate, which is essentially free.
2. The parties wish to keep their dataset size private as well. Now we need to perform a secure multiplication for each summand, but these can be done in parallel, so the computation is still cheap. This is known as an *inner product.* Additionally, we compute n separately with K sums, and reveal the value prior to division.
3. In the most private scenario, the parties are unwilling to leak even the complete dataset size n across all parties. In this case, we again compute n privately, but instead of revealing it, we must compute a single secure division.

In a simplified scenario, one can assume that all parties use the same number of data points.

Unsurprisingly, this has already been covered in the existing research literature [29, 30, 31]. Lately, researchers have also explored securing other aggregation methods than FedAvg [32, 33].

Applying DP using MPC is more complex, but still feasible. Mohamad et al [34] have surveyed the state of secure aggregation, and list a number of techniques and ideas based on different primitives, like FL and DP. Combining FL with DP is an active research field, to which both academia and many of the large technology companies contribute to this effort [35, 36].

We finish this section with a brief discussion of some of the options that MPC may provide us.

We assume K clients in the FL scenario. However, not all need to participate in the computations. Instead, they may appoint a low number, say N, of trustees. Each of the K then create N *shares* of their data and distribute the shares to the trustees. These perform the computation on behalf of the clients.

A share is the technical term for how data in MPC is distributed. We briefly describe two common methods to illustrate.

1. Given a secret number a, choose N random numbers $a_1, a_2, \dots, a_N$ such that they sum to a. Send the share a_i to player i. If all do the same, then the

computing players may easily sum and scale the shares. Multiplications intuitively require more communication to compute the cross-terms.

2. Notice that we need two points to uniquely determine a line. This generalises to us needing $n + 1$ points to uniquely determine a polynomial of degree n. By choosing a random polynomial p of degree $t - 1$, and setting the constant term to our secret a, we can distribute the share $(i, p(i))$ to player i, for arbitrarily large i [37]. One can show that it is easy to sum shares, how to multiply, and that any t players suffice to *reconstruct* the secret.

Generally, fewer players give more efficient protocols, and much of the current MPC research is oriented towards efficient 2-party or 3-party protocols.

MPC also gives the user a choice of security models. Generally, we assume the existence of an adversary that controls all dishonest parties. We must assess the capabilities of the adversary:

- **Honest**: This adversary would not look at plaintext data and always behave according to the protocol. That would imply that no security was needed and should, therefore, be ignored in this context.
- **Semi-honest**: This adversary is also called *honest-but-curious* or *passive*. It will follow the protocol but use any available data to learn secrets.
- **Covert**: This adversary may deviate from the protocol, but not in such a way that it would be detected.
- **Dishonest or active**: This adversary will deviate from the protocol at will.

Defending against a dishonest adversary requires more work from the protocol than defending from a semi-honest adversary. One must assess the correct level per application.

We must also determine the reach of the adversary. Common choices are assumptions that the adversary may corrupt less than a third of the players, less than half, or more than half. As before, stronger adversaries require more from the protocol.

As a result, several protocols have been devised and implemented by cryptographers. These protocols can be used to run generic programs. The MP-SPDZ software [38] provides a nice benchmarking and testing platform for such protocols and programs.

3.2.3 Work Plan

3.2.3.1 Approach and Tools

Based on the 3GPP Technical Specification 23.700 (Rel. 17), NWDAFs can be deployed across various areas within a large Public Land Mobile Network. These are distributed

to handle local data analytics close to the network functions they are associated with, such as the User-Plane Function (UPF) or Session-Management Function (SMF). In the specification, FL is proposed as a possible solution to handle issues such as data privacy and security, and model-training efficiency in this setup. For example, with a multiple-level NWDAF architecture, NWDAFs may be co-located with a 5G Core Network Function (e.g., UPF, SMF), and the raw data cannot be exposed due to privacy concerns and performance reasons. In such case, FL will be a good way to let a Server NWDAF coordinate with multiple localised NWDAFs to execute an ML algorithm.

In the context of our development, the local NWDAF instances will be positioned near the gNodeBs, which provide communication for the UEs, in proximity to the end users, whereas the NWDAF aggregator will be situated either on the service provider's side or hosted in the cloud.

Regarding the development of FL, Flower[1] will be used. Flower is an open-source Python library designed for federated learning, offering a flexible and modular framework that seamlessly integrates with popular ML tools like TensorFlow and PyTorch. It supports efficient model updates and aggregation through a lightweight communication protocol, reducing overhead in data transmission between the server and numerous client devices. Flower's architecture is built for scalability, capable of handling large-scale deployments and distributed processing. It also enhances privacy and security, incorporating techniques such as DP and secure aggregation to protect data integrity. Moreover, Flower allows for high customisation in aggregation strategies and client behaviour, making it a robust choice for our experiments that require precise and adaptable FL strategies.

Opacus[2] is a PyTorch library specifically designed to facilitate the training of machine learning models while ensuring DP and safeguarding user data. This library seamlessly integrates into existing PyTorch workflows with minimal required adjustments and supports a wide variety of PyTorch models and training setups.

At the core of Opacus's architecture is the PrivacyEngine, which orchestrates the DP mechanisms. The engine alters the training process to include critical features such as per-sample gradient computation, gradient clipping, and the addition of noise. These features ensure that the model training adheres strictly to DP standards. Furthermore, Opacus employs Privacy Accountants, particularly using the Rényi DP (RDP) framework, to meticulously track the privacy budget. The RDP accountant method enables the accumulation of privacy costs associated with each operation, ensuring that the total privacy expenditure remains within set limits.

[1] https://flower.ai/
[2] https://opacus.ai/

We have not yet finalised our choice of an MPC framework for our FL secure-aggregation process. This decision, which will be made in the coming months following a thorough analysis of the specific threat model and protocol that best suits our project's needs, will be documented in Rel. B of D3.1.

3.2.3.2 Experiments, Evaluation and Next Steps

Building upon the Key Performance Indicators (KPIs) defined in deliverable D2.1 "6G threat landscape and gap analysis" [3], we must develop an experimental setup tailored to rigorously evaluate these KPIs across various model configurations. This setup will involve comparative analyses across two primary dimensions: centralised versus federated models, and private versus non-private models.

In the centralised versus federated dimension, we will compare the performance of models trained in a traditional centralised manner against those trained using our FL approach. This comparison will help us measure the impact of distributing the learning process across multiple nodes on metrics such as accuracy loss, detection time, and the ability to handle adversarial workers.

For the private versus non-private dimension, experiments will focus on comparing models that implement privacy-preserving techniques to those that do not. This will allow us to assess the trade-offs between privacy and utility, specifically how privacy measures affect model accuracy, sensitivity to data changes, and the model's resilience against adversarial privacy attacks.

This dual comparative approach ensures a comprehensive evaluation of the models across all critical aspects mentioned in the KPIs, providing a clear picture of how our security and privacy enhancements affect overall system performance.

3.2.4 Current status and next steps

3.2.4.1 Explanatory Data Analysis of 5G Core Network Data

The models developed in the context of “Trustworthy AI model building”, are consuming the 5th version of NCSRD-DS-5GDDoS dataset described in section 2.2.2. The dataset is composed of multivariate time series-data, which includes multiple variables recorded in regular time intervals. Each series in this dataset consists of various metrics related to 5G radio and core networks, capturing the performance and behaviour of network components during normal operation and sporadic DDoS attacks. This format enables us to observe the dynamics of network traffic and security incidents over time, making it suitable for temporal analysis and anomaly detection.

The temporal resolution of the dataset is granular, with metrics and events logged on every 5 seconds. The dataset encompasses a concentrated observational period with specified time frames, highlighting the onset and end of network traffic, as well as the precise start and end times of DDoS attacks. In the 5th version of the dataset, coverage includes:

- A three-day window from March 20, 2024, to March 23, 2024, where normal traffic was generated and monitored between all 5G devices and two endpoints within the network.
- Two distinct DDoS attack events on March 23, 2024, and March 24, 2024, each lasting one hour.

To facilitate an in-depth analysis and enhance our understanding of the dataset, we developed an interactive dashboard utilising Streamlit[3] for the application framework and Plotly[4] for the interactive visualisations. The dashboard is designed with a multi-select feature for metrics, allowing simultaneous visualisation of various metrics to compare their behaviours. Moreover, a rolling average can be applied to the selected metrics, with window-size options of 60, 120, 180, 240, 300, and 360 time steps, respectively. This functionality is used for smoothing the time series data, thereby mitigating the effects of short-term fluctuations and revealing underlying trends in metrics that exhibit high variability.

Figure 3.4 showcases the dashboard's utility, specifically showcasing the interface for the '`ul_bitrate`' metric. It provides a clear visualisation of the uplink bitrate over time

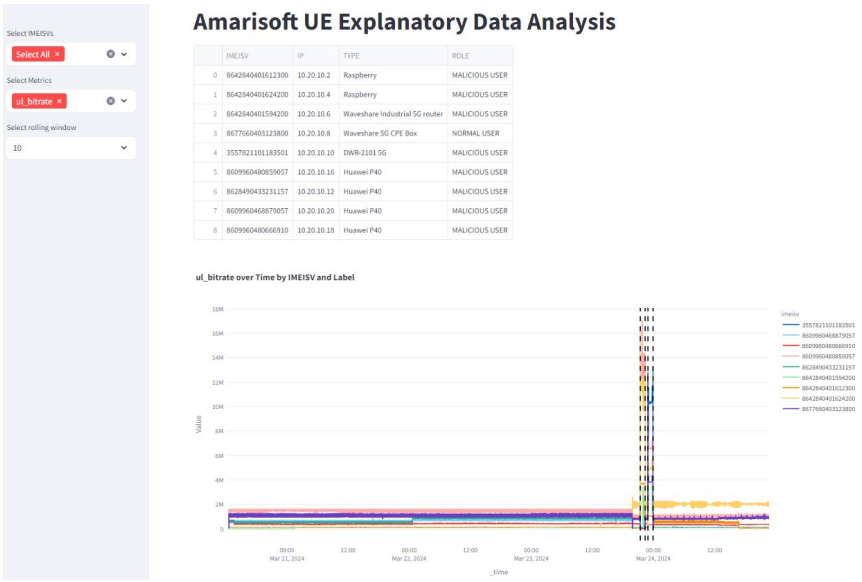

Figure 3.4: Screenshot of Streamlit application for explanatory data analysis

[3] https://streamlit.io/
[4] https://plotly.com/

across multiple devices. The vertical dotted lines within the visualisation distinguish the time intervals of DDoS attacks, offering immediate visual cues to the temporal correlation between the attacks and network behaviour.

The results of the Exploratory Data Analysis (EDA) highlighte some deficiencies in data generation. Specifically, the following observations were made:

- Devices of `imeisv 3557821101183501, 8609960480666910, 8628490433231157, 8642840401594200, 8677660403123800` were unsuccessful in conducting the first attack.
- Devices of `imeisv 3557821101183501, 8642840401594200, 8642840401612300, 8642840401624200` were inactive during the period of benign operation.
- There was a disconnection for all devices from 2024-03-23 18:38:00 PM to 21:54:00 PM.

For training the AI model, we used benign traffic data from active devices prior to their disconnection. For testing, we utilised benign traffic data collected after the second attack.

3.2.4.2 Deep-Learning Model

High-Level Overview of Model Architecture
The model we have developed is an LSTM autoencoder, as shown in Figure 3.5, a type of neural network architecture that combines the capabilities of LSTM networks and autoencoders. This model is particularly suited for time-series anomaly detection due to its ability to capture temporal dependencies and learn data representations. LSTM is a type of RNN that is capable of learning long-term dependencies in sequential data. LSTM networks have a memory cell that can maintain information for an arbitrary amount of time, allowing them to process sequences of arbitrary length. They are particularly useful for tasks such as language translation, speech recognition, and time-series forecasting.

Autoencoders are a type of neural network architecture used to learn efficient data encodings in an unsupervised manner. They work by encoding an input into a latent-space representation and subsequently reconstructing the output from this representation, hence learning a compact representation of the data.

LSTM autoencoders are a good fit for the time-series data due to their ability to handle sequential information over longer periods and their proficiency in reconstructing a learned sequence, making them ideal for identifying anomalies that manifest as deviations from the learned sequences [39]. The combination of LSTMs and autoencoders has been demonstrated to be effective in various time-series applications, from speech-signal processing to the prediction of complex industrial processes.

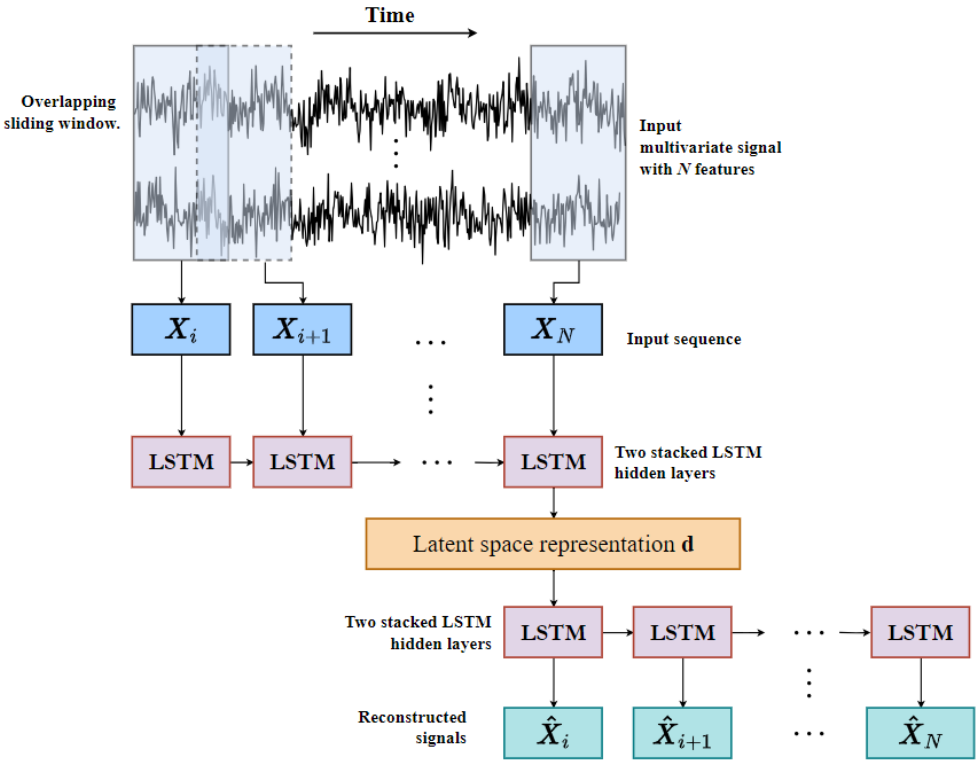

Figure 3.5: Architecture of an LSTM-based autoencoder for time series analysis, showing the flow from input multivariate signals, through encoding by two stacked LSTM layers, to reconstruction of the output signal.

Model Input

The LSTM network requires a three-dimensional input array. The first dimension represents the batch size, the second dimension represents the time steps, and the third dimension represents the number of units in one input sequence. To construct this three-dimensional input, we used a method of rolling overlapping windows on the multivariate time-series data. For each time step, we consider a fixed window of previous time steps and a fixed window of future time steps, with some overlap between the windows. This method segments the data into sequences where each sequence contains a predefined number of time steps. By sliding the window size over the original data by a certain number of steps less than the window size, each new sequence slightly overlaps with the previous, ensuring no temporal continuity is lost.

The input is structured as a tensor with shape `(batch_size, sequence_length, num_of_features)`, where `batch_size` is the number of sequences in a batch, `sequence_length` is the number of time steps in each sequence, and `num_of_features` is the number of features considered in each time step.

Model Output & Reconstruction Loss

The fundamental task of an autoencoder, including our LSTM autoencoder, is to perform data reconstruction. These models try to learn two functions: an encoding

function that transforms the input data, and a decoding function that recreates the input data from the encoded representation. Thus, although the processes of the hidden layers of both the encoder and decoder networks may change the dimensionality of the layers' output, the final output of the model must match the dimensionality of the original input.

To measure the performance of the model in reconstructing the input data, reconstruction loss is used. The reconstruction loss is a measure of the difference between the input data and the output of the model. By minimising the reconstruction loss, we can train the model to learn a representation of the input data that can be used for anomaly detection.

There are several reconstruction losses that can be used, depending on the specific requirements of the task. Some common reconstruction losses include:

- **Mean-Squared Error (MSE) also known as L2**: This is a common loss function in ML which measures the average-squared difference between the input and output.
- **Mean-Absolute Error (MAE) also known as L1**: This loss function measures the average absolute difference between the input and output. It is less sensitive to outliers than MSE but may not capture the full range of differences between the input and output.
- **Kullback-Leibler Divergence (KLD)**: This is a measure of the difference between two probability distributions. It is often used in generative models, where the input and output are probability distributions.

Model Training & Experimentation

To fine-tune the LSTM Autoencoder for optimal performance, we employed a systematic hyperparameter-tuning approach using Grid Search. This method involves defining a comprehensive search space and systematically evaluating the model's performance across all possible combinations of hyperparameter values. The process ensures that we identify the set of hyperparameters that yields the best results according to predefined metrics.

The search space for our model was defined as follows:

- **Window Size**: The number of time steps per sequence in the input data. We explored window sizes of 60, 90, and 120 to determine the optimal temporal context for modelling.
- **Model-Architecture Configuration**: Variations in the model's architecture were considered by adjusting the dimensions of the hidden layers. Configurations ranged from smaller setups (25 units in the first hidden layer and 25 or 50 units in the second) to larger ones (50 units in the first and 100

units in the second), exploring the trade-offs between model complexity and performance.

- **Layer Normalisation**: We tested the model both with and without layer normalisation to investigate its impact on training stability and performance.
- **Dropout rate**: Dropout rates of 0.2 and 0.3 were tested to prevent overfitting by randomly omitting subsets of features at each training step.
- **Loss Function**: The model was evaluated using both L1 Loss and Mean-Squared Error Loss, to compare their effectiveness in the reconstruction task.
- **Learning Rate**: Learning rates of 0.0001 and 0.001 were examined to optimise the speed and convergence of training.

Window size, model architecture configuration and layer normalization hyperparameters concern the details of the model architecture, while the rest concern the training process.

The window size in an LSTM network, determining the amount of temporal context available per sequence, is crucial for capturing long-term dependencies essential for understanding patterns in time-series data. A larger window size can enhance the model's ability to integrate and learn from extended historical data, thus potentially improving its accuracy in scenarios where past events influence future outcomes. However, it also increases model complexity and the computational load, which may lead to longer training times and higher memory usage. Additionally, while a larger window might help in smoothing out noise and focusing on underlying trends, it could dilute important short-term signals and delay the model's response time in real-time applications. Conversely, a smaller window size might make the model more sensitive to noise and less capable of detecting patterns or anomalies that occur over longer periods. Therefore, selecting an optimal window size involves balancing these aspects to align with the specific characteristics of the dataset and the operational constraints, aiming to optimise both the detection capabilities and computational efficiency of the model.

Model-architecture configuration concerns the internals of the model. We have utilised a regularised overcomplete autoencoder, which is a type of autoencoder that has been regularised to learn useful features from the data distribution. It is overcomplete because the code dimension (h) is greater than the input dimension (x). This means that the autoencoder has more capacity than necessary to copy the input to the output, forcing it to learn meaningful representations of the data. Regularisation and dropout are used to prevent overfitting and encourage the model to learn substantial characteristics in a proper way.

Dropout is a regularisation technique where randomly selected neurons are ignored during training, meaning their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to

the neuron on the backward pass. By introducing dropout, we effectively reduce the likelihood of overfitting by ensuring that the model does not become overly dependent on any single neuron or pattern in the training data.

Regularisation has been added to the model via the AdamW [40] optimiser, which is a variation of the Adam optimiser that corrects the way weight decay is applied in the original Adam [41] algorithm. The main difference between Adam and AdamW lies in the handling of the weight-decay regularisation term. In traditional stochastic gradient descent (SGD), weight decay works by shrinking the weights by a small factor during each update, effectively imposing an L2 penalty. In the original Adam optimiser, the weight decay is added directly to the gradients, much like in SGD. However, Adam also scales the gradients by a running average of the magnitude of recent gradients. Because of this scaling, the effect of weight decay is also scaled, which can lead to an inconsistent application of weight decay. AdamW decouples the weight decay from the gradient updates. Instead of applying weight decay to the gradients, AdamW applies weight decay directly to the weights after the optimiser step.

Regarding the loss function we decided to experiment using L1 Loss and MSE Loss, which are the most dominant reconstruction losses in the literature. As mentioned above, MSE Loss is highly sensitive to outliers. This sensitivity is beneficial for tasks requiring precise reconstruction, where large errors are especially problematic. Conversely, L1 Loss calculates absolute differences and is less affected by outliers, offering a more robust metric in environments with anomalies that do not necessarily signal model failures.

The learning rate is a tuning parameter in an optimisation algorithm that determines the step size at each iteration while moving toward a minimum of a loss function. It essentially controls how much the weights of the model are adjusted with respect to the loss gradient. A suitable learning rate ensures efficient convergence to a minimum loss, balancing the speed and stability of the learning process.

Choosing different values for the learning rate can significantly impact the model's training dynamics. Higher learning rates can lead to faster convergence but may overshoot the minimal loss points, causing the training process to be unstable or even diverge, while lower learning rates promote more stable and reliable convergence by taking smaller steps. However, they risk slowing down the training process, potentially leading to long training times and sometimes getting trapped in local minima. Experimenting with learning rates of 0.0001 and 0.001 allows us to find an optimal balance that maximises learning efficiency without compromising the stability and accuracy of the model's performance.

Feature Selection

After conducting the EDA, we identified eight features that could potentially be most relevant in identifying a DDoS attack on the network. Those are the following:

- `dl_bitrate` **(Downlink Bitrate)**: Measures the rate at which data is received by a user's device from the network. Measured in bits per second (bps).
- `ul_bitrate` **(Uplink Bitrate)**: Measures the rate at which data is sent from a user's device to the network. Measured in bits per second (bps).
- `dl_retx` **(Downlink Retransmissions)**: Counts the number of times packets are re-transmitted on the downlink. Re-transmissions occur when packets are lost or corrupted during transmission. High downlink re-transmission rates can indicate network congestion or signal-quality issues.
- `dl_tx` **(Downlink Transmissions)**: This represents the total number of packets transmitted in the downlink direction.
- `ul_tx` **(Uplink Transmissions)**: This represents the total number of packets transmitted in the uplink direction.
- `ul_total_bytes` **(Uplink Total Bytes)**: Measures the total amount of data sent from the user devices to the network over a given period.
- `dl_total_bytes` **(Uplink Total Bytes)**: Measures the total amount of data received by the user devices from the network over a given period.

Inference
During anomaly detection with autoencoders, the inference process begins by passing test data through the trained autoencoder. The model calculates the reconstruction loss for each data point. A predefined threshold is set for this loss, and data points exceeding this threshold are classified as anomalies. This approach is based on the premise that the autoencoder, trained solely on normal data, should reconstruct such data with minimal error. Consequently, significant deviations in reconstruction loss indicate anomalies, effectively identifying unusual or atypical data patterns.

To calculate the optimal threshold for anomaly detection, we used a method based on the Receiver Operating Characteristic (ROC) curve. The ROC curve is a plot of the true-positive rate (TPR) against the false-positive rate (FPR) at various threshold values. It provides a visualisation of the performance of a binary classifier, such as our autoencoder, in terms of its ability to distinguish between normal and anomalous data.

The process involves first calculating the FPR and TPR for a range of threshold values. Next, we calculate the distances between the TPR and FPR values at each threshold. The threshold with the minimum distance is considered the optimal threshold, as it represents the balance between the TPR and FPR.

By using this method, we can automatically determine the optimal threshold that maximises the detection of anomalies while minimising false positives.

Best Model & Results

The dataset was divided into three subsets: training, validation, and test sets. The test set consists of user equipment (UE) traffic recorded from the end of the second attack to the conclusion of the recording period. The remaining benign activity was split for training and validation, with 80% of the data used for training and 20% for validation. The model configuration that yielded the best performance on the test set is the following:

- **Window Size**: 120
- **Hidden Dim of 1st LSTM Layer (both encoder & decoder)**: 50
- **Hidden Dim of 2nd LSTM Layer (both encoder & decoder)**: 100
- **Layer Normalisation**: False (no layer normalisation applied)
- **Dropout rate**: 0.2
- **Loss Function**: L1 Loss
- **Learning Rate**: 0.001

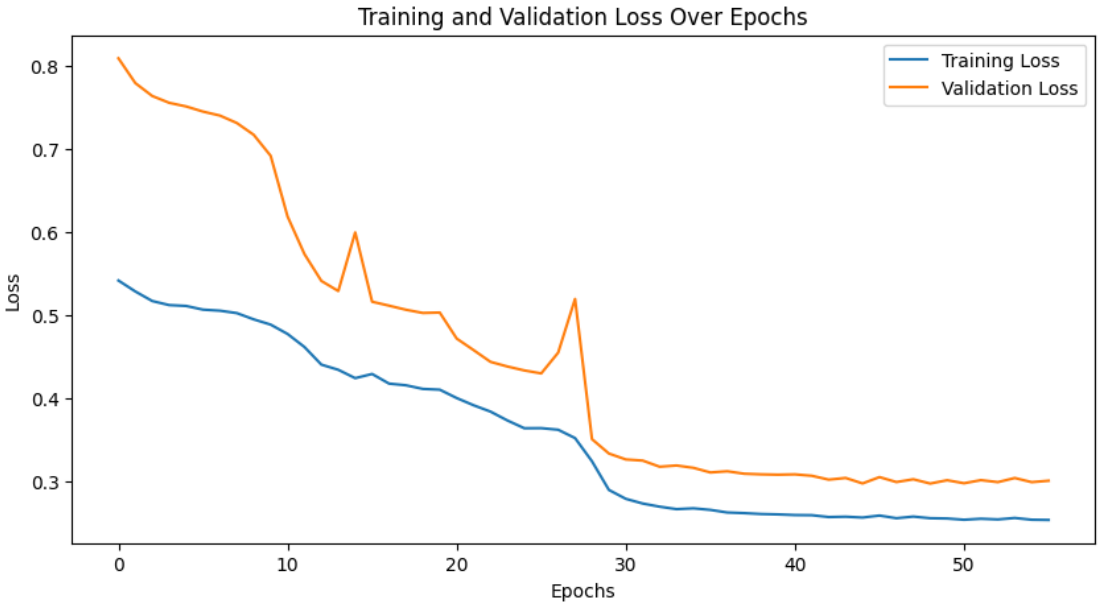

Figure 3.6: Training and validation L1 Loss during training epochs.

Figure 3.6 depicts the training process of the model with the above configurations in terms of training and validation loss.

Throughout the training phase, the model demonstrates a consistent decrease in training loss, indicating effective learning and model fitting to the training data. Notably, the validation loss largely mirrors the downward trend of the training loss, suggesting that the model is generalising well to unseen data. There are occasional spikes in validation loss, which could be attributed to variations in the validation dataset; however, the overall trajectory remains downward. This convergence of both

training and validation loss is indicative of a stable learning process, with no significant signs of overfitting, as the model's performance improves steadily over the epochs.

The ROC curve depicted in Figure 3.7, reveals an area under the curve of 0.98. This large area indicates an excellent level of discrimination; the model is highly capable of identifying true anomalies while maintaining a low rate of false positives. The curve's proximity to the top left corner of the graph underscores the model's effectiveness, as it reflects a high TPR and a low FPR, which are desirable characteristics for reliable anomaly-detection models.

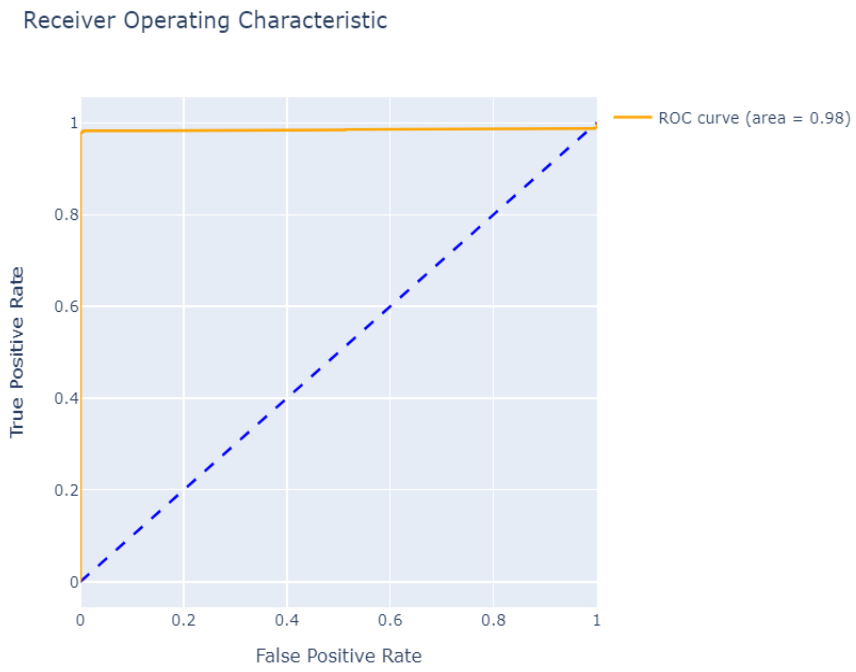

Figure 3.7: Receiver Operating Characteristic plot after model evaluation on test data.

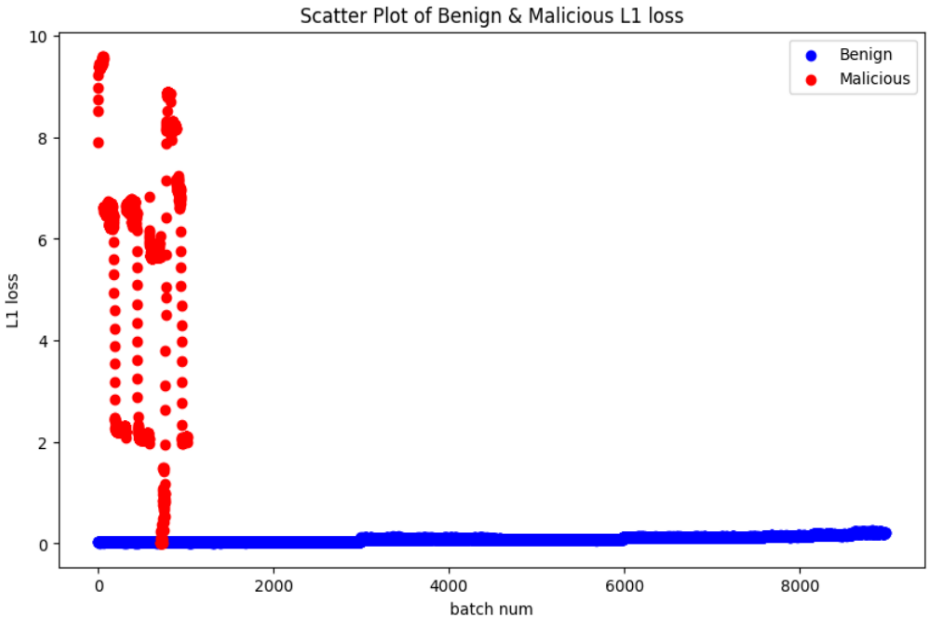

Figure 3.8: Scatterplot of L1 loss for benign & malicious data points.

The scatterplot in Figure 3.8 visualises the reconstruction loss, measured by the L1 loss, for test data labelled as benign (blue) and malicious (red). Trained solely on benign data, the model has learned to reconstruct such behaviour accurately, resulting in the dense cluster of blue points near the origin, indicative of low reconstruction loss. Conversely, the malicious data points, shown in red, are scattered predominantly higher on the loss axis. This elevated loss of malicious data underscores the model's inability to reconstruct anomalous behaviour accurately, which it has not encountered during training. The distinct separation between the reconstruction losses for benign and malicious data validates the model's utility in recognising and flagging deviations from the learned benign patterns as anomalies.

The detection metrics of the model evaluated on the test set are the following:

- **Accuracy**: 0.9943
- **Precision**: 0.9632
- **Recall**: 0.9813
- **F1 Score**: 0.9722
- **True Positives (TP)**: 995 (98.13%)
- **True Negatives (TN)**: 8937 (99.58%)
- **False Positives (FP)**: 38 (0.42%)
- **False Negatives (FN)**: 19 (1.87%)

The performance metrics of our model demonstrate its high effectiveness in anomaly detection. Recall, or the true positive rate, stands at 0.9813, signifying that the model successfully captures 98.13% of all actual anomalies. The F1 Score, which balances precision and recall, is at a robust 0.9722, further underscoring the model's reliability.

In absolute terms, out of the anomalies present in the data, 995 were correctly identified as TP, and 8937 TN were accurately recognised, resulting in very few false alarms and missed anomalies, 38 and 19 respectively.

Interpretation of the Model's Behaviour

In assessing the robustness of our model, we undertook a comprehensive analysis to understand the behaviour it has learned. One significant observation is the clear difference in absolute values and variance between benign and malicious time series. Our analysis focused on whether the DL model successfully extracted meaningful patterns that could differentiate these two distinct behaviours.

Figure 3.9 highlights the distinctive patterns of uplink bitrate for various active devices during normal period and attack period. The upper charts, depicted in blue, represent the benign operation while the lower charts, in red, illustrate periods of malicious activity.

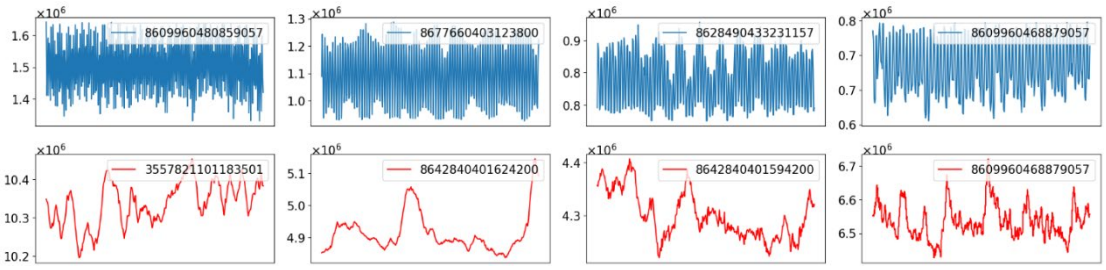

Figure 3.9: Uplink bitrate for different devices during benign and malicious activity.

To clarify the behaviour of the model, we aimed to quantify the disparity between data inputs for each class. This was achieved by applying L1-norm to each data sample, facilitating a comparative analysis of their distributions. The same metric was then employed to measure the distribution of the model's outputs. Figure 3.10 showcases the results of this analysis.

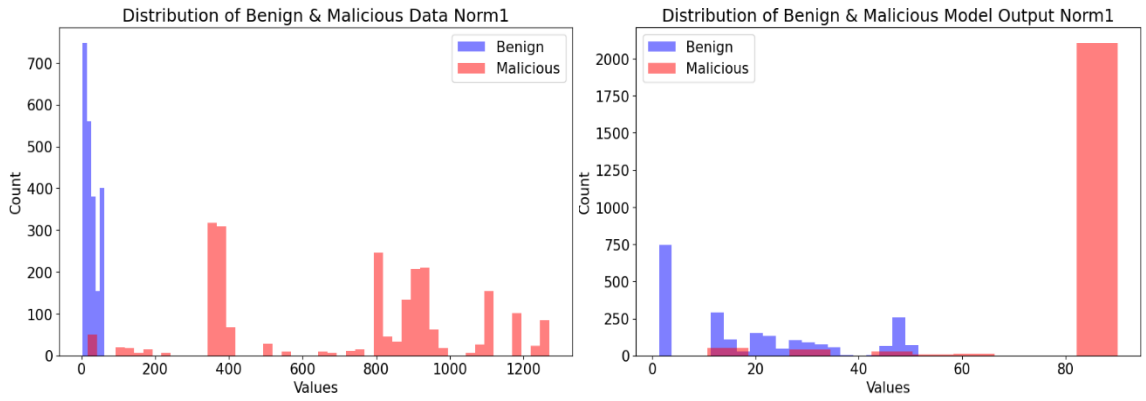

Figure 3.10: Histograms of L1-norm (Norm1) for benign and malicious input data and model outputs.

The benign data points are characterised by lower L1-norm values, mostly concentrated under 60, indicating consistency in the benign class's behaviour. On the other hand, malicious data points are distributed over a broader range of norm values, indicating more varied behaviour with the potential for high norm values. When processed by the autoencoder, the resulting data points for both classes are bounded within a reduced norm range (<150), with malicious data points being heavily concentrated near the upper bound. The norm distribution for the benign data points has not altered significantly. This suggests that the autoencoder, which was trained exclusively on benign data, has learned a reconstruction function optimised for low-norm vectors, effectively reducing reconstruction errors for benign time series.

The second test we conducted to understand the behaviour of our model involved mitigating the influence of higher absolute values of malicious time series data during model inference. To achieve this, we applied distinct scalers to the benign and malicious segments of the time-series data. Then we tested our model on the scaled malicious data. The detection capacity of the model significantly deteriorated.

This can also be illustrated by comparing the original and reconstructed time series from a device that exhibited both benign and malicious behaviour. This pattern is depicted in Figure 3.11.

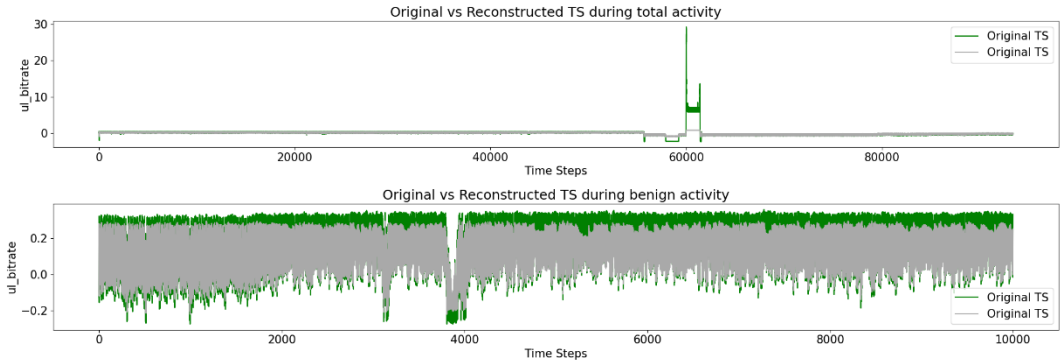

Figure 3.11: Original vs Reconstructed Time Series

During benign operations, the model accurately captured the time series. In contrast, for the malicious activity, the model failed to replicate the high values observed. Instead, it consistently outputs a stable threshold value throughout the malicious phase.

Overall, the LSTM-autoencoder model effectively distinguished malicious from benign traffic. This success was primarily achieved by focusing on the elevated values of relevant features, a behaviour that is expected during DDoS attacks. Interestingly, the model did not derive significant additional value from the intercorrelation of multiple features. Instead, it efficiently identified spikes in the absolute values within the

feature space, a strategy that, while somewhat simplistic in only considering DDoS attacks, proved to be quite effective. This simplicity invites further exploration to assess the model's potential against other types of attacks, suggesting an interesting avenue for future research to repeat the process and identify how the model performs under different malicious conditions.

The next steps involve implementing the privacy-preserving techniques outlined in the previous section on top of the developed model.

3.3 Adversarial-Robustness Evaluation and Feedback

3.3.1 Objectives

Task 3.3 aims to use adversarial tools and techniques to test, evaluate and increase the adversarial robustness of AI models. The AI models' robustness against different types of attacks will be evaluated. This will then feed into an adversarial training pipeline to increase the robustness of the AI models against these types of attacks. Task 3.3 has the following objectives:

- Apply adversarial tools to assess the adversarial robustness of AI models from Task 3.2. This approach should evaluate the AI models' robustness to attacks and indicate areas for improvement.
- Explore techniques to increase the adversarial robustness of AI models through adversarial training, regularisation, and other methods. This will then be used to create an adversarial-training paradigm that will enhance the robustness of the AI models.
- Evaluate the quality loss of the AI models for each method in the paradigm to ensure that the methods do not degrade the model beyond usefulness.

3.3.2 State of the Art

3.3.2.1 Evasion Attacks/Adversarial Examples

An attack based on adversarial examples perturbs the input to the models in a way that may force the model to misclassify. The changes to the input need to be so small that it still behaves in the same general way [42]. A typical example of this involves adding an imperceptible layer of noise to an image forcing the model to classify it as a different object, even though the image still looks like the original object to the human eye [43]. The exact mechanism for how adversarial examples work is not entirely understood and it is not always necessary to have the exact model to create adversarial examples that fool the model. If two models are trained to solve the same general problem, chances are that an adversarial example created to fool one of them may

fool both [44]. As this is an attack in the test phase of the model, the effects and defences against it should be similar, both inside and outside an FL context.

3.3.2.2 Membership-Inference Attacks

Membership-inference attacks are a group of attacks in which an attacker is able to deduce whether a data point was part of the training set of a trained model. This may implicitly leak sensitive information. For instance, an attacker can check whether a specific person was part of the training set of a model designed to predict the probability of cancer within a patient [45].

3.3.2.3 Attribute-Inference Attacks

Attribute-inference attacks are similar to membership-inference attacks but differ slightly in that an attacker with partial knowledge about a data point may be able to learn unknown values about that data point. This can be used to infer sensitive values used in training [45].

3.3.2.4 Poisoning Attacks

In poisoning attacks, an attacker with access to the training dataset may be able to manipulate the training set in order to influence a model to behave in a certain undesirable way. An attacker can use this to create a backdoor into the model for evasion or similar. For instance, an attacker could force a security model to classify certain malicious activity as benign [46].

3.3.2.5 Adversarial Training and Robustness

Adversarial training is one of the most common defensive strategies against adversarial attacks, where the model is trained against adversarial examples constructed to fool the model [43, 44, 47].

A Generative Adversarial Network (GAN) architecture [48], as shown in Figure 3.12, can be used for such adversarial training. A GAN network contains two interconnected networks: a generator and a discriminator. The generator network generates adversarial examples based on a randomly drawn vector from the latent space, while the discriminator network is trained to correctly classify both the original and the constructed examples [44]. The latent vector can be seen as a reduced dimensionality sample, as the generator uses the latent vector to generate a full sample.

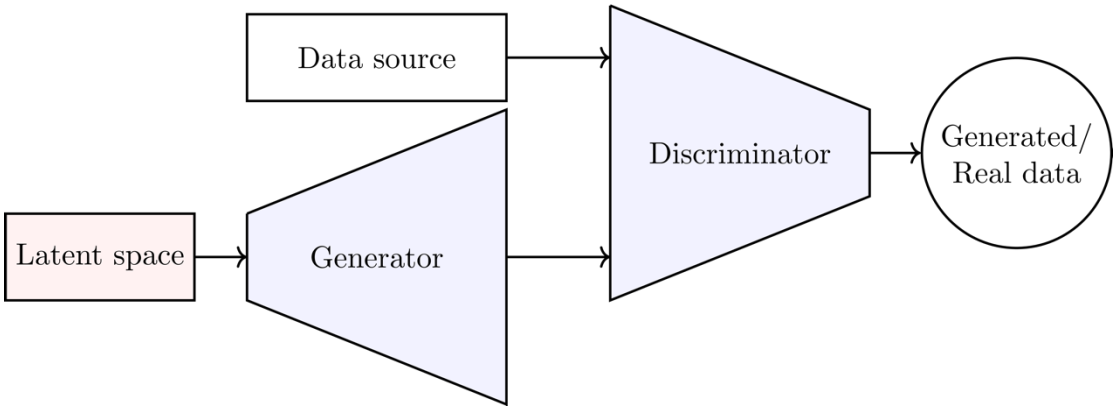

Figure 3.52: Example of a GAN Architecture

Other defences include using provable robust ML models, input transformation, using DP, and using detection mechanisms for adversarial examples [44]. Using provable robust ML models may provide the strongest defence but may provide a lower accuracy and may suffer scalability issues with complex neural networks [43]. Training a model using DP techniques may help to make the model more robust against adversarial attacks as they introduce "benign" perturbations into the training set [49]. DP shows many similarities to adversarial training in both method and effects. It can be shown that adversarial training can achieve a good trade-off between privacy and model accuracy [50, 51, 52].

Input transformation is a model-agnostic method that attempts to improve robustness by removing any adversarial elements from the input. This can be done, e.g., by adding random noise to the input to try to counteract the small adversarial perturbations [44, 47]. A GAN-based de-noising technique has also been proposed [47, 49].

Detecting adversarial examples before they are presented to classifier model can effectively eliminate the problem of misclassifying adversarial examples. This can be done by observing how a neural network behaves when faced with adversarial examples. For example, dimensional properties, feature attribution scores, and distances between adjacent classes may behave differently with adversarial examples [49]. The use of a variational autoencoder has also been proposed for detecting adversarial examples [49].

3.3.2.6 Robustness Frameworks

There are several different robustness frameworks available for machine learning models. The Adversarial Robustness Toolbox (ART) is a Python-based open-source framework created by IBM for generating adversarial attacks and defences [53]. It provides a variety of different attacks for evasion attacks, poisoning attacks, and inference attacks. It also provides defences such as adversarial example detection, poisoning detection, general adversarial training, and data preprocessing defences

[53]. In addition, it also provides several robustness metrics, such as Empirical Robustness, Loss Sensitivity and CLEVER [53, 54].

Foolbox is another Python-based open-source framework and provides adversarial attacks, defences and robustness metrics [55].

There are other frameworks available, such as Cleverhans, though this is focused mainly on computer vision models [56].

3.3.3 Selected Attacks

This task will start looking into membership inference attacks. Next, the task will investigate poisoning attacks. The task may then investigate attribute inference, evasions attacks and model extraction attacks.

3.3.4 Work Plan

The work plan of the task can be divided into three phases.

Phase one will start with threat modelling of the attack scenario (membership-inference attacks). The attack scenario will be used for initial experiments using ART [53], and testing on simplified datasets and models. The attacks used will feature adversarial examples and evasion attacks using GANs and GAN-like methods. ART will mainly be used as it is an actively developed framework implementing both attacks and defences in addition to robustness metrics. Other frameworks may also be tested to complement or replace the ART framework wherever it seems beneficial.

Phase two will consist of applying the selected attacks on the LSTM-model and dataset from Task 3.2. This will be evaluated, and adversarial examples will be provided for re-training the model. If possible, other mitigation methods to improve robustness of the models from Task 3.2, such as gradient masking, regularisation, and adversarial example detection, will also be explored.

Phase three of the task will consist of evaluating the newly trained models and assessing their robustness against the selected attacks. In addition, the quality loss of the model will be evaluated.

Phase two and phase three will be repeated for at least poisoning attacks, and, time permitting, attribute inference, evasions attacks and/or model extraction attacks.

3.3.5 Evaluation Set Up

In general, the evaluation will compare the number of attacks achieved before and after mitigations are applied and assess the quality loss of the model after mitigations are applied. In addition to evaluating the success rate of attacks, several robustness metrics of the models will be evaluated. This will include Empirical Robustness, Loss-sensitivity and CLEVER [53, 54].

3.4 XAI-driven decision support

3.4.1 Objectives

Task 3.4 aims to provide insight in a human-understandable manner in the decentralised AI built for security analytics in Tasks 3.2 and 3.3. The primary objective is to offer explanations on the decision-making process through eXplainable AI (XAI) techniques, particularly focusing on aiding anomaly detection and threat classification.

Through the utilisation of XAI methodologies, the objective is to provide insights into how the security-analytics models arrive at their decisions. In an ideal implementation, it could include model and decision interpretability for a given threat and anomaly detection. Additionally, techniques such as counterfactual information can assess how confidently a decision was made by the model and if particular data points alter the decisions significantly.

Explanations require the input data derived from the processed 5G NWDAF, as established in Tasks 3.1 and 3.2, and leverage the anomaly-detection models from Tasks 3.2 and 3.3, which identify and categorise threats within the security-analytics platform. From the outputs of these tasks, XAI models are developed to provide explanation either by model-agnostic implementations such as SHAP or LIME, or white models such as Decision Trees or Naïve Bayes that provide a clear understanding of their decision-making process.

The outcomes of this task are expected to encompass the integration of XAI approaches within a federated system, ensuring that explanations are provided both at the core and edge systems compatible with FL and anonymisation pipelines. Next, the task will design and implement a specialised decision-support framework for PRIVATEER's use cases. This framework will incorporate a variety of XAI algorithms tailored to complement objectives such as interpretability and explainability at both local and global levels. It is imperative that this decision-support framework accommodates state-of-the-art DL algorithms trained with private anonymised data, all within a federated and distributed environment.

3.4.2 State of the Art

Explainability in Artificial Intelligence (XAI) can be defined as an approach aimed at enhancing the transparency and comprehensibility of AI decision-making processes. This methodology, while not aggressively assertive, strives to simplify the inherent complexity of ML models, facilitating a more accessible understanding without compromising performance significantly. XAI aims to strike a balance between technical complexity and the human ability for comprehension and trust in intelligent-system decisions.

There are key concepts when exploring XAI models to explain model decision. One is explainability which pertains to the XAI model's capacity to elucidate the reasons and mechanisms behind a specific prediction or its internal behaviour. Another one is interpretability which refers to the human user's capability to comprehend the explanations offered.

Apart from the previous definition, XAI models have a set of desirable properties such as transparency, interpretability, trustworthiness, fairness, transferability, bias detection, robustness, and domain understanding. Each of these properties are important to define and characterise XAI models which are reliable for human interpretability [57].

3.4.3 XAI Models and Taxonomy

XAI models are used to explain decisions made by traditional known black-box models. In the literature, there is a clear distinction between white- and black-box models fundamentally based on a human ability to be able to review and understand how an ML model created a decision [58].

Our interest is in the exploration of instance-based decisions. This is, evaluating the contribution of each input feature to the output of a model for generalisation purposes.

One model-agnostic XAI model is the Local Interpretable Model-agnostic Explanations (LIME) algorithm [59]. LIME [59] operates by building a simplified, understandable model that mimics the classification-model's behaviour. This simplified model is trained using local data points and is then used to explain the classification model’s decision. The process encompasses the following stages:

- Selection of an instance for explanation.
- Perturbation of the instance to generate a dataset comprising similar instances.
- Assignment of weights to similar instances based on their resemblance to the instance being explained.

- Training of a local, interpretable model using the weighted dataset.
- Utilisation of the local model to provide explanations for the complex model's decision.

A different model-agnostic approach can be found in SHAP (Shapley Additive Explanations) [60]. SHAP is a technique designed to explain individual predictions, leveraging the game-theoretically optimal Shapley values. Shapley values, derived from cooperative game theory, offer favourable attributes. In this context, the feature values of a data instance are analogous to players in a coalition, and the Shapley value represents the average marginal contribution of a feature value across all potential coalitions.

Yet another model-agnostic approach to XAI for DL algorithms is Integrated Gradients [61]. It is a model-agnostic approach aimed at assigning significance to each input feature within a deep neural network. Its objective is to evaluate the contribution of each input feature to the network's ultimate prediction by assessing the extent to which the prediction alters when varying that particular feature.

3.4.3.1 Counterfactual Explanations

Counterfactual explanations refer to hypothetical instances or scenarios that are generated to explain model predictions. One strategy for explanation is through counterfactuals, which reveal what changes in an instance could lead to a different outcome [62]. By presenting these hypothetical scenarios, users can gain a better understanding of the decision-making process of the AI model and identify potential biases or areas for improvement.

There are different strategies to produce counterfactual explanations:

- **Optimisation**: Counterfactual explainers utilising optimisation strategies formulate a loss function incorporating desired properties and employ existing optimisation algorithms to minimise it;
- **Heuristic-Search Strategy**: Counterfactual explainers employing heuristic search strategies aim to discover counterfactuals by making local and heuristic choices at each iteration to minimise a specific cost function;
- **Instance-Based**: Instance-based counterfactual explainers generate counterfactuals by selecting the most similar examples from a dataset;
- **Decision Tree**: Counterfactual explainers based on DT approximate the behaviour of the black-box model with a decision tree and then utilise the tree structure to identify counterfactual explanations.

There are other proposals for explainability methods, as proposed in [63]. This strategy aims to classify strategies based on their functioning approach:

- **Local Perturbations**: The modification of the inputs of a model to assess confidence in the decision, assert feature importance and robustness of a decision;
- **Leveraging Structure**: Leveraging the internal structure of models to identify and build explanations. In the field of DL, it is common to look at internal gradients to ascertain feature importance;
- **Meta-Explanation**: Usage of alternative XAI methods to create explanations based on the decision model. It might involve dedicated heuristics developed for a specific use case;
- **Architecture Modification**: Altering the decision model, trying to make it simpler and thus more understandable;
- **Examples**: Use past examples as explanations for current classifications.

Yet another alternative is proposed by by McDermid et al. [64]. This classification is influenced by the result of an explanation to create its categories:

- **Feature Importance**: identification of the most important features for the explanation of a decision based on a model and instance of input;
- **Surrogate Models**: usage of simpler, self-explanatory models to approximate the decision process of more complex models;
- **Examples**: a strategy that aims to explain decisions based on similar examples from previous inputs.

Apart from categories, explanations also should take into consideration XAI scope and stage [65]. Regarding scope, it can be either local, restricted to the instance being used or global, considering the whole model. The stage is related to whether the explanation is devised before or after the application of an ML algorithm.

3.4.3.2 Time Series, XAI and FL

Techniques used for analysing time-series data are often adapted from computer vision and natural-language processing domains [66]. Their aim is to emphasise the specific signal components that receive the most focus from the model during classification. However, these methods may overlook certain characteristics inherent to time-series data, such as recurring spatio-temporal patterns and correlations among multiple channels or sensing modalities [67].

In the case of time series data, XAI models employ different strategies based on XAI theory [68], such as:

- **Time Points-Based Explanations**: Assign a relevance score or weight to every time point of a time series;
- **Subsequence-Based Explanations**: Identify sub-parts of a time series responsible for the classification outcomes;

- **Instance-Based Explanations**: Rely on the whole time series instance to express the reasons for the classification.

Time series have implications in problems and algorithms used. XAI requires specialised approaches to capture the temporal data dependencies between instances. Nevertheless, approaches using simple XAI processes although able to be used with relative success, fail to link temporal relationships.

Implementations of federated XAI are available in the literature [69]. They follow the same strategy as federated learning. This means that XAI models may be trained and maintained at the edge node, using the federated mechanism to merge different XAI models for the same domain on a central server [69]. A critical concept here is the ability to merge XAI models in a convergent path able to produce quality explanations both at the central node and the local nodes.

3.4.4 Work Plan

The current work plan defined for Task 3.4 is to review the outputs from the previous tasks and understand the data and algorithms used for threat classification and anomaly detection. In the current approach, an LSTM model will be employed together with time-series data. With this reality, we aim to employ XAI models as digital twins for the black box models in the security analytics platform.

In order to achieve the results, our work plan involves an initial stage of experimentation and selection of a relevant XAI model for the PRIVATEER security-analytics platform. These initial tests aim to assess and produce proofs of concept (PoCs) that explain decisions on threat identification and anomaly detection. At this stage, the initial PoC, based on explanation dashboards, will aim to support the classification model with XAI explanations ante-hoc and post-hoc and both global and local in nature.

A second stage aims to specialise the XAI models to the problem domain, namely time-series telecommunication data. Time series are a special subset of problems and require special treatment in order to produce quality explanations.

At a third stage, the federated nature of the problem will be taken into consideration. Preparation for potential extension of the XAI models for the federated scenario will be specified and the corresponding theoretical framework will be devised.

At a fourth and final stage, an XAI decision toolkit will be implemented for the use cases selected, with the previous work from the latter stages. This toolkit should demonstrate the application of XAI for the security-analytics platform in the PRIVATEER project and offer modularity, different strategies, and extensibility in the implementation of XAI models.

3.5 Edge analytics accelerators

3.5.1 Objectives

The objective of Task 3.5 is to facilitate the implementation of hardware-accelerated data-analytics applications at the edge, aiming for both low latency and energy efficiency while having a secure operation.

Initially, the focus is on accelerating AI/ML methodologies for anomaly-detection tasks such as threat identification and classification. Additionally, optimisation and approximation techniques will be applied in order to maximise energy efficiency (i.e., by reducing the precision of computations).

The required input data and anomaly-detection AI model will derive from Tasks 3.1 and 3.2. aiming to classify threats within the security-analytics platform. Hardware accelerators such as FPGAs or GPUs will be examined with the ultimate goal of enhancing the energy efficiency of the given AI model on the inference process when compared with the execution on a general-purpose processor (i.e., CPU).

3.5.2 State of the Art

The aim of this paragraph is to assess the current state of development regarding hardware accelerators for various devices, including GPUs, FPGAs, and several approximation techniques.

For Task 3.5, we opt to examine Xilinx FPGAs or Nvidia GPUs as representative device types, along with their associated tools. This choice is motivated by their current market dominance, widespread adoption by major cloud providers, and the potential impact of these novel compute elements on edge systems.

Hardware accelerators, including devices like the Xilinx Alveo or MPSoC FPGA family, as well as Nvidia GPU devices, are commonly interfaced with CPUs via PCI-Express or integrated as complete system-on-a-chip (SoC) solutions with peripherals [70, 71, 72]. The hardware or tool manufacturers provide an interface layer, such as a shell design in the FPGA or a driver for the specific GPU, along with corresponding software components comprising necessary libraries executed on the server(s) or host side. Xilinx Vitis environment includes a comprehensive core development kit to seamlessly build accelerated applications for both server and SoC Xilinx FPGAs. Vitis AI development environment is a separate specialised development environment for accelerating AI inference on Xilinx platforms in a more automated way. Regarding the GPU development, Nvidia Cuda Toolkit is a development environment for building GPU-accelerated applications, including libraries, debugging and optimisation tools, a C/C++ compiler, and a runtime library. Also, Nvidia Cuda-X which is built on top of

Nvidia Cuda, is a collection of libraries, tools, and technologies that deliver higher performance in multiple application domains in a more automated way.

Applications ranging from accelerating basic signal processing functions such as Fast Fourier Transform (FFT) [73], data encryption such as Advanced Encryption Standard (AES) [74] or even DL applications [75, 76, 77] are constantly being developed using emerging technologies from both software and hardware domains. Wherever low latency and power efficiency are vital, FPGA or GPU devices can be leveraged to offload the computation of the CPU and process the most compute-intensive part of the applications while the CPU is free to perform other important tasks.

3.5.3 Work Plan

The work plan of Task 3.5 is to deliver solutions for accelerating security analytics workloads. Specifically, within this task, an optimised hardware kernel will be developed, considering balancing energy versus performance trade-offs for hardware/software co-design targeted for anomaly-detection application. Also, approximation-computing techniques will be applied to enhance the energy efficiency of the algorithm compared to CPU software execution.

Specifically, below is the work plan in distinct steps:

1. Identification of target kernel for acceleration: Extensive discussions with the AI model providers from Task 3.2 have taken place in order to determine the appropriate ML or DL model that will be targeted for hardware acceleration. The AI model has parallelisation potential; thus, it will benefit from being deployed on a hardware-accelerator platform such as FPGA or GPU.
2. Development of the first version of accelerated AI model: High-Level Synthesis will be utilised for developing the model for FPGAs or CUDA programming model for developing the model for GPUs. Also, advanced tools for automating the creation of firmware for hardware, particularly in the context of DL algorithms, will be examined if needed.
3. Approximation techniques will be examined to further enhance the energy efficiency on the hardware-accelerated models. This might include any of the following: quantisation, precision scaling, loop perforation or approximate function memorisation.
4. The kernels will be integrated with the rest of the security-analytics application.

Below in Figure 3.13 is the custom development flow for both FPGAs and GPUs when targeting the trained AI analytics model that will result from Task 3.2. The output

kernel will be able to perform inference of the model using FPGA or GPU hardware, achieving higher energy efficiency than CPU.

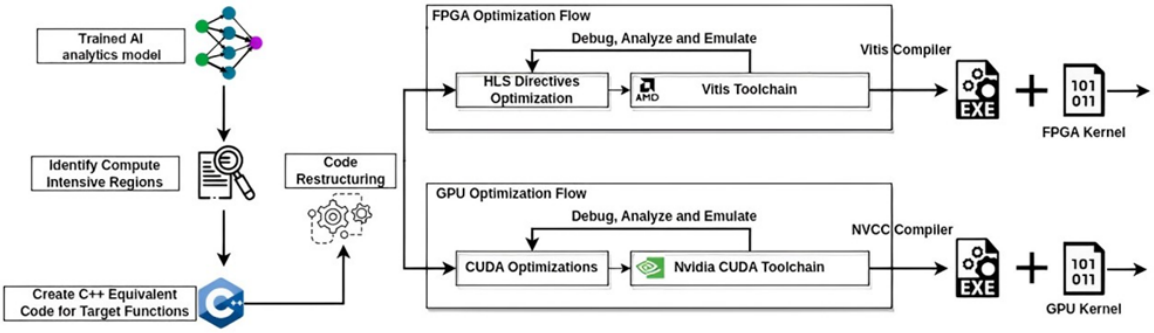

Figure 3.13: Custom development flow for FPGAs and GPUs.

4 Conclusion

In summary, all five tasks of WP3 have started successfully. A detailed work plan has been provided for all technical tasks within the WP3. Their interdependencies have been analysed and are respected in the WP advancements for optimal collaboration and multitasking.

As described in the present report, a thorough data and feature analysis has been performed regarding the 5G NWDAF dataset containing a DDoS attacks. Following the EDA and general considerations for the 5G landscape regarding anonymisation needs and data suitability for ML purposes, within the Tasks 3.1 and 3.2 anonymisation pipelines for location data and a security-analytics DL model have been constructed and implemented. These constitute the basis for further developments within Tasks 3.2, 3.3, 3.4 and 3.5 during the second phase of the PRIVATEER project.

Crucial next steps will involve the adversarial hardening against specific attacks in Task 3.3, explainability methods in Task 3.4, and HW acceleration in Task 3.5. In Task 3.2, several privacy and security settings will be explored by employing DP and MPC, and the privacy-utility trade-off will be investigated in depth for real-life use cases and, in particular, use cases 1 and 4 of PRIVATEER as elaborated in deliverable D2.2 [78]. Systematic federated DL experimentation set ups will be created for the anomaly-detection algorithms. Computational experiments will be conducted in all significant parameters related to security and privacy, such as the privacy budget or the stage of applying DP, in order to reach conclusions regarding an optimal privacy-utility balance for PRIVATEER's use cases. Secure-aggregation mechanisms will be investigated with respect to their effectiveness and efficiency, in order to be implemented into the federated security-analytics framework.

References

[1] P. Meena, M. B. Pal, P. K. Jain and R. Pamula, "6G communication networks: introduction, vision, challenges, and future directions," *Wireless Personal Communications,* vol. 125, no. 2, pp. 1097-1123, 2022.

[2] V.-L. Nguyen, P.-C. Lin, B.-C. Cheng, R.-H. Hwang and Y.-D. Lin, "Security and privacy for 6G: A survey on prospective technologies and challenges," *IEEE Communications Surveys & Tutorials,* vol. 23, no. 4, pp. 2384-2428, 2021.

[3] F. Scaglione, C. Petrollini and F. Manti, "PRIVATEER Deliverable D2.1: 6G threat landscape and gap analysis," *10.5281/zenodo.7994961,* 2023.

[4] S. A. A. Hakeem, H. H. H. and H. Kim, "Security Requirements and Challenges of 6G Technologies and Applications," *Sensors (Basel),* vol. 22, p. 1969, 2022.

[5] V. Ziegler, P. Schneider, H. Viswanathan, M. Montag, S. Kanugovi, A. Rezaki and e. al., "Security and Trust in the 6G Era," *IEEE Access,* vol. 9, pp. 142314-142327, 2021.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, 2017.

[7] P. Liu, X. Xu and W. Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," *Cybersecurity,* vol. 5, no. 4, 2022.

[8] C. Dwork, "Differential Privacy," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, 2006.

[9] Y. Lindell, "Secure Multiparty Computation (MPC)," *IACR Cryptol. ePrint Arch.,* p. 300, 2020.

[10] R. Kumar, S. Gupta, H.-C. Wang, C. Kumari and S. Korlam, "From Efficiency to Sustainability: Exploring the Potential of 6G for a Greener Future," *Sustainability,* vol. 15, no. 23, p. 16387, 2023.

[11] National Centre of Scientific Research "Demokritos", Space Hellas (Greece), "NCSRD-DS-5GDDoS: 5G Radio and Core metrics containing sporadic DDoS attacks," Zenodo, 10.5281/zenodo.10671494, 2024.

[12] M. Cunha, R. Mendes and J. P. Vilela, "A Survey of Privacy-Preserving Mechanisms for Heterogeneous Data Types," *Computer Science Review,* vol. 41, no. 100403, 2021.

[13] M. E. Andrés, N. E. Bordenabe, C. K. and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications security,* pp. 901-914, 2013.

[14] R. Mendes and J. P. Vilela, "Privacy-Preserving Data Mining: Methods, Metrics, and Applications," *IEEE Access,* vol. 5, pp. 10562-10582, 2017.

[15] R. Mendes, M. Cunha and J. P. Vilela, "Impact of frequency of location reports on the privacy level of geo-indistinguishability," *Proceedings on Privacy Enhancing Technologies,* 2020.

[16] B. Baron and M. Musolesi, "Where You Go Matters: A Study on the Privacy Implications of Continuous Location Tracking," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,* vol. 4, no. 4, pp. 1-32, 2020.

[17] M. Cunha, G. Duarte, R. Andrade, R. Mendes and J. P. Vilela, "Privkit: A Toolkit of Privacy-Preserving Mechanisms for Heterogeneous Data Types," *ACM Conference on Data and Application Security and Privacy (CODASPY),* p. 6, 2024.

[18] A. J. G. de Azambuja, C. Plesker, K. Schützer, R. Anderl, B. Schleich and V. R. Almeida, Artificial Intelligence-Based Cyber Security in the Context of Industry 4.0—A Survey, vol. 12, Multidisciplinary Digital Publishing Institute, 2023, p. 1920.

[19] M. Rodríguez, Á. Alesanco, L. Mehavilla and J. García, Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection, vol. 22, 2022, p. 9326.

[20] A survey of deep learning-based network anomaly detection | Cluster Computing.

[21] M. Almseidin, M. Alzubi, S. Kovacs and M. Alkasassbeh, Evaluation of machine learning algorithms for intrusion detection system, IEEE, 2017, pp. 277-282.

[22] G. Pang, C. Shen, L. Cao and A. v. d. Hengel, Deep Learning for Anomaly Detection: A Review, vol. 54, 2022, pp. 1-38.

[23] M. Zhu, K. Ye and C.-Z. Xu, Network Anomaly Detection and Identification Based on Deep Learning Methods, M. Luo and L. Zhang, Eds., Springer International Publishing, 2018, pp. 219-234.

[24] C. Dwork, F. McSherry, K. Nissim and A. D. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," *J. Priv. Confidentiality,* vol. 7, p. 17–51, 2016.

[25] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold and A. L. Roth, "Preserving Statistical Validity in Adaptive Data Analysis," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, 2015.

[26] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov and M. Naor, "Our Data, Ourselves: Privacy Via Distributed Noise Generation," in *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, 2006.

[27] R. C. Geyer, T. Klein and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective," December 2017.

[28] X. Zhang, X. Chen, M. Hong, Z. S. Wu and J. Yi, "Understanding Clipping for Federated Learning: Convergence and Client-Level Differential Privacy," June 2021.

[29] X. Cao, M. Fang, J. Liu and N. Z. Gong, "FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping," December 2020.

[30] C. Brunetta, G. Tsaloli, B. Liang, G. Banegas and A. Mitrokotsa, "Non-interactive, Secure Verifiable Aggregation for Decentralized, Privacy-Preserving Learning," in *Information Security and Privacy - 26th Australasian Conference, ACISP 2021, Virtual Event, December 1-3, 2021, Proceedings*, 2021.

[31] G. Tsaloli, B. Liang, C. Brunetta, G. Banegas and A. Mitrokotsa, "DEVA: Decentralized, Verifiable Secure Aggregation for Privacy-Preserving Learning," in *Information Security - 24th International Conference, ISC 2021, Virtual Event, November 10-12, 2021, Proceedings*, 2021.

[32] M. G. Belorgey, S. Dandjee, N. Gama, D. Jetchev and D. Mikushin, "Falkor: Federated Learning Secure Aggregation Powered by AES-CTR GPU Implementation," *IACR Cryptol. ePrint Arch.,* 2023.

[33] T. Gehlhar, F. Marx, T. Schneider, A. Suresh, T. Wehrle and H. Yalame, "SafeFL: MPC-friendly Framework for Private and Robust Federated Learning," in *2023*

IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, May 25, 2023, 2023.

[34] M. Mansouri, M. Önen, W. B. Jaballah and M. Conti, "SoK: Secure Aggregation Based on Cryptographic Schemes for Federated Learning," *Proc. Priv. Enhancing Technol.,* vol. 2023, p. 140–157, 2023.

[35] Apple, "pfl: Python framework for Private Federated Learning simulations," [Online]. Available: https://apple.github.io/pfl-research/index.html. [Accessed 11 04 2024].

[36] M. Raykova, *Advances (and Challenges) in Secure Aggregation, PPML 2023,* Santa Barbara, 2023.

[37] A. Shamir, "How to Share a Secret," *Commun. ACM,* vol. 22, p. 612–613, 1979.

[38] M. Keller, "MP-SPDZ: A Versatile Framework for Multi-Party Computation," in *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, 2020.

[39] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal and G. Shroff, LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection, arXiv, 2016.

[40] I. Loshchilov and F. Hutter, Decoupled Weight Decay Regularization, arXiv, 2019.

[41] D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, arXiv, 2017.

[42] X. Yuan, P. He, Q. Zhu and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 9,* pp. 2805-2824, Sept. 2019.

[43] I. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," *Proc. Int. Conf. Learn. Represent. (ICLR),* 2015.

[44] J. Zhang and C. Li, "Adversarial examples: Opportunities and challenges," *IEEE transactions on neural networks and learning systems 31.7,* pp. 2578-2593, 2019.

[45] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu and X. Zhang, "Membership Inference Attacks on Machine Learning: A Survey," *ACM Computing Surveys, Volume 54, Issue 11s,* pp. 1-37, September 2022.

[46] Z. Tian, L. Cui, J. Liang and S. Yu, "A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning," *ACM Comput. Surv. 55, 8, Article 166,* 2022.

[47] S. Zhou, C. Liu, D. Ye, T. Zhu, W. Zhou and P. S. Yu, "Adversarial Attacks and Defenses in Deep Learning: From a Perspective of Cybersecurity," *ACM Computing Surveys 55.8,* pp. 1-39, 2022.

[48] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial networks," *Commun. ACM 63, 11,* pp. 139-144, November 2020.

[49] P. Samangouei, M. Kabkab and R. Chellappa, "Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models," in *6th International Conference on Learning Representations*, 2018.

[50] M. Nasr, R. Shokri and A. Houmansadr, "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning," in *IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2019.

[51] M. Nasr, R. Shokri and A. Houmansadr, "Machine Learning with Membership Privacy using Adversarial Regularization," in *ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, New York, NY, USA, 2018.

[52] J. Hamm, "Minimax filter: learning to preserve privacy from inference attacks," *The Journal of Machine Learning Research, Volume 18, Issue 1,* pp. 4704-4734, Jan. 2017.

[53] M.-I. Nicolae, M. Sinn, M. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy and B. Edwards, "Adversarial Robustness Toolbox v1.2.0," *CoRR,* 2018.

[54] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh and L. Daniel, "Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach," in *International Conference on Learning Representations*, 2018.

[55] J. Rauber, R. Zimmermann, B. Matthias and W. Brendel, "Foolbox Native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX," *Journal of Open Source Software,* vol. 5, p. 2607, 2020.

[56] I. Goodfellow, N. Papernot and P. D. McDaniel, "cleverhans v0.1: an adversarial machine learning library," *Computing Research Repository, abs/1610.00768,* 2016.

[57] A. Das and P. Rad, "Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey," *arXiv,* 2020.

[58] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila and F. Herrera, "Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible," *AI Inf. Fusion,* vol. 58, pp. 82-115, 2020.

[59] M. T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?," in *Why Should I Trust You?*, New York, USA, 2016.

[60] S. M. Lundeberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[61] M. Sundararajan, A. Taly and Q. Yan, "Axiomatic Attribution for Deep Networks," in *Proceedings of the 34th International Conference on Machine Learning, PMLR 70: 3319-3328*, 2017.

[62] R. Guidotti, "Counterfactual explanations and how to find them: literature review and benchmarking," *Data Min. Knowl. Discov.,* 2022.

[63] W. Samek and K. R. Müller, "Towards Explainable Artificial Intelligence," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 11700 LNCS, pp. 5-22, 2019.

[64] J. A. McDermid, Y. Jia, Z. Porter and I. Habli, "Artificial intelligence explainability: the technical and ethical dimensions," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 379, no. 2207, p. 20200363, 10 2021.

[65] T. Speith, "A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods," *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency,* p. 2239–2250, 2022.

[66] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin and N. Díaz-Rodríguez, "Explainable Artificial Intelligence (XAI) on TimeSeries Data: A Survey," 4 2021.

[67] F. D. Martino and F. Delmastro, "Explainable AI for clinical and remote health applications: a survey on tabular and time series data," *Artif. Intell. Rev.,* vol. 56, p. 5261–5315, 2023.

[68] A. Theissler, F. Spinnato, U. Schlegel and R. Guidotti, "Explainable AI for Time Series Classification: A Review, Taxonomy and Research Directions," *IEEE Access,* vol. 10, pp. 100700-100724, 2022.

[69] A. Renda, P. Ducange, F. Marcelloni, M. C. F. D. Sabella and G. S. A. V. D. M. D. R. L. G. B. G. Nardini, "Federated Learning of Explainable AI Models in 6G Systems: Towards Secure and Automated Vehicle Networking.," *Information,* vol. 13, 2022.

[70] M. Asiatici, N. George, K. Vipin, S. A. Fahmy and P. Ienne, "Virtualized Execution Runtime for FPGA Accelerators in the Cloud," *IEEE Access,* vol. 5, pp. 1900-1910, 2017.

[71] A. Li, S. L. Song, J. Chen, J. Li, X. Liu, N. R. Tallent and K. J. Barker, "Evaluating Modern GPU Interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect," *IEEE Trans. Parallel Distrib. Syst.,* vol. 31, p. 94–110, January 2020.

[72] C. Kachris, B. Falsafi and D. Soudris, Hardware Accelerators in Data Centers (1st. ed.), Springer Publishing Company, Incorporated, 2018.

[73] J. Sheng, C. Yang, A. Sanaullah, M. Papamichael, A. Caulfield and M. C. Herbordt, "HPC on FPGA clouds: 3D FFTs and implications for molecular dynamics," in *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, 2017.

[74] C. Luo, Y. Fei, P. Luo, S. Mukherjee and D. Kaeli, "Side-channel power analysis of a GPU AES implementation," in *2015 33rd IEEE International Conference on Computer Design (ICCD)*, 2015.

[75] T. Geng, T. Wang, A. Sanaullah, C. Yang, R. Xu, R. Patel and M. Herbordt, "FPDeep: Acceleration and Load Balancing of CNN Training on FPGA Clusters," in *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2018.

[76] H. Sharma, J. Park, N. Suda, L. Lai, B. Chau, J. K. Kim, V. Chandra and H. Esmaeilzadeh, "Bit Fusion: Bit-Level Dynamically Composable Architecture for Accelerating Deep Neural Network," in *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 2018.

[77] D. Danopoulos, G. Zervakis, K. Siozios, D. Soudris and J. Henkel, "AdaPT: Fast Emulation of Approximate DNN Accelerators in PyTorch," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 42, pp. 2074-2078, 2023.

[78] A. Pastor, H. Ramon and D. Lopez, "PRIVATEER Deliverable D2.2: Use cases, requirements and design report," https://zenodo.org/records/10027073, 2023.

Consortium

Space Hellas
www.space.gr

NCSR Demokritos
www.demokritos.gr

Telefonica I&D
www.telefonica.com

RHEA SYSTEM SA
www.rheagroup.com

INESC TEC
www.inesctec.pt

Infili Technologies PC
www.infili.com

UBITECH LTD
www.ubitech.eu

IQUADRAT R&D
www.ucm.es

ICCS
www.iccs.gr

FORSVARETS FORSKNINGSINSTITUTT
www.ffi.no

UNIVERSIDAD COMPLUTENSE DE MADRID
www.ucm.es

INSTITUTO POLITÉCNICO DO PORTO
www.ipp.pt

ERTICO ITS EUROPE
www.ertico.com

Contact Us
info@privateer-project.eu

PRIVATEER has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101096110

Co-funded by the European Union
6GSNS