5Genesis

5TH GENERATION END-TO-END NETWORK, EXPERIMENTATION, SYSTEM INTEGRATION, AND SHOWCASING

[H2020 - Grant Agreement No. 815178]

Deliverable D3.14

5G Security Framework (Release B)

Editor	A. Priovolos (SHC)
Contributors	SHC, INF
Version	1.0
Date	March 31st, 2021
Distribution	PUBLIC (PU)

List of Authors

SHC	Space Hellas (Cyprus) Ltd.
A.Priovolos, D. Lioprasitis, G. Gardikis	
INF	INFOLYSIS P.C.
V. Koumaras, G. Theodoropoulos	

© 5GENESIS Consortium

Disclaimer

The information, documentation and figures available in this deliverable are written by the 5GENESIS Consortium partners under EC co-financing (project H2020-ICT-815178) and do not necessarily reflect the view of the European Commission.

The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. The reader uses the information at his/her sole risk and liability.

© 5GENESIS Consortium

Copyright

Copyright © 2021 the 5GENESIS Consortium. All rights reserved.

The 5GENESIS Consortium consists of:

NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS"	Greece
AIRBUS DS SLC	France
ATHONET SRL	Italy
ATOS SPAIN SA	Spain
AVANTI HYLAS 2 CYPRUS LIMITED	Cyprus
AYUNTAMIENTO DE MALAGA	Spain
COSMOTE KINITES TILEPIKOINONIES AE	Greece
EURECOM	France
FOGUS INNOVATIONS & SERVICES P.C.	Greece
FON TECHNOLOGY SL	Spain
FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V.	Germany
IHP GMBH – INNOVATIONS FOR HIGH PERFORMANCE MICROELECTRONICS/LEIBNIZ-INSTITUT FUER INNOVATIVE MIKROELEKTRONIK	Germany
INFOLYSIS P.C.	Greece
INSTITUTO DE TELECOMUNICACOES	Portugal
INTEL DEUTSCHLAND GMBH	Germany
KARLSTADS UNIVERSITET	Sweden
L.M. ERICSSON LIMITED	Ireland
MARAN (UK) LIMITED	UK
MUNICIPALITY OF EGALEO	Greece
NEMERGENT SOLUTIONS S.L.	Spain
ONEACCESS	France
PRIMETEL PLC	Cyprus
RUNEL NGMT LTD	Israel
SIMULA RESEARCH LABORATORY AS	Norway
SPACE HELLAS (CYPRUS) LTD	Cyprus
TELEFONICA INVESTIGACION Y DESARROLLO SA	Spain
UNIVERSIDAD DE MALAGA	Spain
UNIVERSITAT POLITECNICA DE VALENCIA	Spain
UNIVERSITY OF SURREY	UK

This document may not be copied, reproduced or modified in whole or in part for any purpose without written permission from the 5GENESIS Consortium. In addition to such written permission to copy, reproduce or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

Version History

Rev. N	Description	Author	Date
1.0	Release of D3.14	A. Priovolos (SHC)	30/03/2021

© 5GENESIS Consortium

LIST OF ACRONYMS

Acronym	Meaning
API	Application Programming Interface
CA	CONSORTIUM AGREEMENT
DARE	Data Analysis and Remediation Engine
GA	GRANT AGREEMENT
GUI	Graphical User Interface
HDFS	Hadoop Filesystem
HW/SW	Hardware / Software
LSTM	Long Short-Term Memory
MEC	Mobile Edge Computing
ML	Machine Learning
NFV	Network Functions Virtualisation
RAN	Radio Access Network
SDN	Software-Defined Networking
YARN	Yet Another Resource Negotiator

© 5GENESIS Consortium

Executive Summary

5G comes with a new rich set of features and capabilities, which, in addition to their obvious technical and business value, are challenged with various side-effects. One such side-effect is the drastic increase in the attack surface, compared to legacy cellular network infrastructures.

One of the objectives of 5GENESIS is to facilitate and enhance the security of 5G networks via its Security Framework that is built around a Security Analytics platform. 5G Security Analytics refers to the collection and joint analysis of massive amounts of heterogeneous data from multiple points of the 5G infrastructure utilized for integrated monitoring. The ultimate aim is the detection and classification of anomalies associated with security incidents, using state-of-the-art ML techniques.

The ambition of 5GENESIS is to realise an effective and efficient 5G Security Analytics framework by:

- Ingesting and adapting heterogeneous data coming from multiple sources
- Selecting and adapting the proper Machine Learning algorithms, also combining them into complex workflows
- Enabling intuitive visualization and query for facilitating security operations normally carried out by the network operator/5GENESIS provider.

5GENESIS builds on well-established technologies from the Big Data / Machine Learning realm, such as Apache Hadoop, Spark and Kafka, and heavily relies on the capabilities of Apache Spot project on cybersecurity analytics.

A key baseline for the 5GENESIS developments has been the DARE (Data Analysis and Remediation Engine) platform, developed in the frame of the EU SHIELD project, which in turn is based on Apache Spot. The 5GENESIS Security Analytics platform operates in three stages, referred to as (a) Data Acquisition, Transformation and Storage; (b) Data Analysis; and (c) Visualisation and Export.

Overall, the Security Framework includes two independent pipelines, which both employ Machine Learning/Deep Learning approaches to infer suspicious behaviours:

- The network flow (NetFlow) processing pipeline ingests and analyses NetFlow streams at various points in the network and focuses on network-level attacks. This was the main component of the Release A of the framework (as described in Deliverable 3.13)
- The metrics processing pipeline analyses infrastructure metrics from various nodes of the infrastructure (RAN, core, network and compute elements) and identifies anomalies which might point to breaches to the compute infrastructure and/or the virtual network functions (VNFs). The metrics processing pipeline is tightly integrated with the 5GENESIS monitoring framework, directly ingesting metrics from there. The metrics processing pipeline was developed during the second phase of the activity and is now integrated in Release B.

Both pipelines have been fully deployed in the 5GENESIS Limassol platform (and is under deployment in the Athens platform), while their performance against a series of emulated

© 5GENESIS Consortium

attacks has been quite promising. The software of the Security Framework has been released as open-source under the GPLv3 license, at https://github.com/5genesis/Security-Framework

© 5GENESIS Consortium

Table of Contents

© 5GENESIS Consortium

© 5GENESIS Consortium

1. INTRODUCTION

1.1. Purpose of the Document

The emergence of 5G networks is admittedly accompanied by a significant increase of the attack surface of the telecom infrastructure. New vulnerabilities identified are associated in particular with the capabilities related to 5G network softwarisation and slicing.

In this context, among its other features, 5GENESIS includes a Security Analytics platform as a contribution towards hardening the security of next-generation networks. This document presents the technical approach for the 5GENESIS Security analytics platform, as well as the relevant work that has been done under WP3/Task 3.7.

1.1.1. Document dependencies

This document is based on specifications, requirements and assumptions as discussed in the Architecture related deliverables. The table below summarizes the relevance towards the deliverables produced by WP2, as well as the first edition of this Deliverable (D3.13)

id	Document title	Relevance
D2.1 [1]	Requirements of the Facility	The document sets the ground for the first set of requirements related to supported features at the testbed for the facilitation of the Use Cases.
D2.4 [2]	Final report on facility design and experimentation planning	The 5GENESIS facility architecture is defined in this document. The list of functional components to be deployed in each testbed is defined. Testing and experimentation specifications that influence the testbed definition, operation and maintenance are defined.
D3.13 [3]	5G Security Framework – Release A	The first version of the 5G Security Framework is presented.

1.2. Structure of the Document

The document is structured as follows:

- Chapter 1, *Introduction* (the present section)
- Chapter 2, *Release A and B summary*, provides and overview of the main features developed during the first phase (Release A) and second phase (Release B) of this activity
- Chapter 3, *Foundation Technologies*, refers to the key technologies that are used to implement the 5GENESIS security framework.

© 5GENESIS Consortium

- Chapter 4, *Overview of the 5GENESIS Security Analytics Platform,* presents the high-level architecture of the security platform to be used in the project.
- Chapter 5, *Testing and Evaluation,* presents the tests conducted and the quantitative evaluation of the developed mechanisms.
- Finally, Chapter 6, *Conclusions*, concludes the document.

1.3. Target Audience

The document is initially targeted to the 5GENESIS project team, towards establishing a common understanding of the architecture and functionalities of the Security Analytics platform and identifying the technical steps needed for its smooth integration in the 5GENESIS Facility.

However, since most of the document is not specifically tied to the project, it can also be addressed to the wider 5G community and stakeholders. Since the outcome of this activity has been open-sourced, this public deliverable can be used by essentially anyone who either wishes to "replicate" the entire 5GENESIS architecture, or just plans to individually exploit and integrate the Security Analytics platform in another 5G system.

1.4. Motivation and Scope

1.4.1. Security aspects in 5G

5G comes with a new rich set of features and capabilities, which, in addition to their obvious technical and business value, as expected, are accompanied with various side-effects, one of the most important of which is the drastic increase in the attack surface, compared to legacy cellular network infrastructures. Some of these 5G-specific capabilities, which, under certain circumstances, may introduce new vulnerabilities and increase the probability of a security incident, are:

- Software-defined infrastructures. The dynamic nature of service-oriented infrastructures introduced in the 5G landscape (be it SDN/NFV, containerised services and applications, etc.) makes it easy to deploy and configure services at the click of a button. In addition, the heterogeneity of HW/SW components and technologies, combined with the fast development cycles associated with the agile paradigm campaigned by most commercial adopters, can drastically increase attack surfaces and the related security/privacy risks of individual components as a whole. Software-based virtual appliances and SDN rules can be subject to malicious modifications, associated with various risks, from data breach to interruption of critical services and applications.
- Slicing and multi-tenancy. The vision of 5G is to enable end-to-end virtualization and true multi-tenancy by dividing the network to slices to be provided to multiple users. However, while in theory slices are expected to be fully isolated, in practice this isolation will be much weaker as expected, since slices share the same physical and often logical resources both in the control as well as the data plane. This implies the potential of either accidental or malicious inter-slice incidents, while operations and events in one slice affect the neighbouring ones.

© 5GENESIS Consortium

- **Multi-actor service paradigms.** In addition to multi-tenancy, 5G promises the further decoupling of service, network and infrastructure providers, allowing multiple actors to be engaged in the 5G ecosystem and share resources, often in a dynamic manner. This has obvious security implications, mostly associated with privacy.
- **Complex, multi-tier architectures.** Apart from the traditional core/backhaul/RAN segmentation, 5G networks introduce additional architectural blocks, such as NFV/SDN Management and Orchestration, in-network compute resources (for NFV/MEC enablement), integration of heterogeneous backhaul/access technologies (such as satellite), cross-domain management functions etc. This increased complexity of the data and, most importantly, the control plane, naturally exacerbates the overall system vulnerabilities.

In 5G, not only the probability of a security incident increases, but also the expected impact and severity. The connection of more and more devices in a 5G network, many of which track personal data, while others support critical operations (as in Intelligent Transport Systems/connected cars or e-health), implies that security incidents can lead to severe privacy breach and/or even life-threatening situations.

It is evident that the emergence of 5G calls for significantly stricter security controls compared to legacy network. A more thorough investigation of the 5G security landscape is out of the scope of the present document and can be found in white papers and reports such as the one [5] produced by the Security Working Group of the 5G PPP, which 5GENESIS is attending and closely following. ENISA has also conducted an exhaustive survey of 5G threats and vulnerabilities [6].

However, at the same time, as a counterbalance to increased risks, 5G technologies also provide new capabilities to establish mitigation measures and contingencies. A key capability is network softwarisation (through SDN and NFV) which can be at the same time a strength and a weakness (as explained above). While SDN/NFV indeed increases the attack surface, at the same time it offers the capability to dynamically deploy virtual security appliances in the network, at the core and also at the edge, and selectively divert traffic through them for enhanced detection/prevention. This capability has been promoted via several projects which leverage software-based networks for security, including the EU SHIELD project[7] and its follow-up, the EU PALANTIR project [8].

Another interesting opportunity stems from the fact that 5G network components are highly heterogeneous and distributed across the network, thus creating an enormous amount of diverse data (mostly logs and monitoring information), whose timely analysis can lead to effective inference of security incidents. This is the concept of 5G security analytics, which is targeted in 5GENESIS and detailed in the next subsection.

1.4.2. 5G Security Analytics

5G Security Analytics refers to the collection and joint analysis of massive heterogeneous data from multiple points of the 5G infrastructure for integrated monitoring, with specific focus on detecting and classifying anomalies associated with security incidents.

Big Data and Machine Learning (ML) technologies are the ideal foundations towards this goal. Big Data infrastructures enable the scalable ingestion, storage and analysis of massive data,

© 5GENESIS Consortium

even in real-time. At the same time, state-of-the-art ML algorithms enable the identification of incidents, which will go unnoticed using traditional rule-based detection. This enables

i) the detection of zero-day attacks, whose exact digital fingerprint is unknown and
ii) the proactive identification of threats even at their very early stages, where detection thresholds of traditional methods have not been crossed.

ML techniques can currently counter Cyber-attacks [9] in two different ways: first, with anomaly detection, where unsupervised algorithms are trained to learn the trusted behaviour in order to detect any irregular ones; second, with threat classification, where supervised learning is used to classify known attacks. The former aims to detect 0-day attacks while the latter is more effective for known attacks. Nowadays, existing cybersecurity solutions combine both techniques: first, they apply anomaly detection, looking for malicious flows, and then use threat classification to identify and classify the type of attack. Current unsupervised anomaly detection algorithms can be sorted in three categories [10]:

- Reconstruction: is based on data compression and posterior reconstruction. Behaviours reconstructed with low error rate are considered “normal” while high error rate ones are considered anomalous.
- Boundary: focuses on finding boundaries around the normal behaviour and every behaviour outside the defined range is considered as anomalous.
- Density Estimation: estimates the probability density function of the training data, it discriminates anomalous behaviours using a threshold value.

The most used supervised classification algorithms in cybersecurity are Support Vector Machines (SVM), Decision trees and Naïve Bayes classifiers. More specifically, [11] has shown that SVMs can be successful in the task of traffic classification, and [12] shows that tree-based methods exhibit very high accuracy measures, while also reducing the need of feature pre-processing. Algorithms based on neural networks are also being considered, but their use seems to be mostly restricted to data of high dimensionality.

The ambition of 5GENESIS has been to realise an effective and efficient 5G Security Analytics framework by:

- Ingesting and adapting heterogeneous data coming from multiple sources
- Selecting and adapting the proper ML algorithms, and also combining them into complex workflows
- Enabling intuitive visualization and query for facilitating security operations.

Overall, the 5GENESIS Security Framework includes two independent pipelines, both of which employ Machine Learning/Deep Learning approaches to infer suspicious behaviours:

- The network flow (NetFlow) processing pipeline ingests and analyses NetFlow streams at various points in the network and focuses on network-level attacks.
- The metrics processing pipeline analyses infrastructure metrics from various nodes of the infrastructure (RAN, core, network and compute elements) and identifies anomalies which might point to breaches to the compute infrastructure and/or the virtual network functions (VNFs). The metrics processing pipeline is tightly integrated with the 5GENESIS monitoring framework, directly ingesting metrics from there.

© 5GENESIS Consortium

2. RELEASE A AND RELEASE B SUMMARY

2.1. Release A Summary

The first phase of the activity (M7-M15) resulted in Release A of the Security Analytics. This focused mainly on the flow processing pipeline and was implemented using as baseline Apache Spot as well as results of the SHIELD project. The focus was on additional features to further optimize its operation.

In order to make the solution more efficient by optimizing the ingest chain, a distributed collector framework was created. The collector is responsible for data ingestion inside Spot and can be hosted in a virtual machine where all the traffic is directed to, to perform the collection. The distributed collector on the other side, can be hosted in several VMs and perform the same task. Having the distributed collector, the traffic does not have to be directed in a specific machine for capturing and it enables the collection of traffic from various sources. Since the distributed collector was created, the data transformation workers had to be modified as well, to "listen" to all collectors and gather data to be sent in Kafka for publishing in HDFS.

In addition, also to facilitate the mitigation of privacy aspects in its application, an add-on feature for IP anonymization was implemented. Due to restrictions, it may not be possible for an IP address to be visible to all users. The IP anonymizer transforms every IP of the collected data into a random IP, there is a one to one mapping between the two IP's, and the whole procedure can continue accordingly with no further interruption. Moreover, we exploited the dashboard offered from Apache Spot to visualize results and be able to observe any anomalies / threats that may occur.

Finally, yet importantly, an internal trial was conducted, also in collaboration with the SHIELD project, in order to validate the current performance of Apache Spot in a pre-operational environment.

Release A was integrated in Release A of the 5GENESIS Limassol platform.

2.2. Release B Summary

The second phase of the activity (M16-M33) focused on two aspects. First, the flow processing pipeline was extended to include anomaly detection based on an Autoencoder algorithm which yielded significantly improved results compared to LDA, as presented in the tests of Section 5. The algorithm was trained using external public NetFlow datasets.

Second, the metrics processing pipeline was developed from scratch. This was built on the existing 5GENESIS monitoring framework (based on the Prometheus monitoring system). Bespoke exporters (collectors) were developed to collect metrics from the 5G RAN and the 5G edge computing infrastructure. An ML processing engine was developed, based on Autoencoder/LSTM (Long Short-Term Memory) model, which was trained using data from "normal" traffic and emulated security incidents, collected in the Limassol platform.

© 5GENESIS Consortium

The results of the algorithm were integrated in the Grafana dashboard, which now combines the monitoring metrics with the detected anomalies, so that the platform operator can have a holistic view of the status of the infrastructure.

Last but not least, extensive tests were carried out to evaluate the performance of both pipelines under emulated security incident scenarios, with quite promising results.

Release B of the Security Analytics framework is being integrated in the Limassol and Athens platforms.

The following table summarises the features of the two releases.

Table 1. Comparison of features per release

Feature	Release A (M15)	Release B (M33)
Network flow collection & ingest	✓	✓
IP anonymisation	✓	✓
RAN metrics collection & ingest		✓
Core (compute) metrics collection & ingest		✓
Scalable storage	✓	✓
ML anomaly detection – Latent Dirichlet Allocation	✓	✓
ML anomaly detection – Autoencoder		✓
ML anomaly detection – Autoencoder/LSTM		✓
Flow processing pipeline visualisation	✓	✓
Metrics processing pipeline visualisation		✓

© 5GENESIS Consortium

3. FOUNDATION TECHNOLOGIES

3.1. Apache Spark, Kafka and the Hadoop ecosystem

3.1.1.1. Hadoop Ecosystem

The Apache Hadoop [13] software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than relying on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly available service on top of a cluster of computers, each of which may be prone to failures. There are four major elements of Hadoop i.e. *HDFS, MapReduce, YARN and Hadoop Common*. Along with these elements, there are extra tools that support/supplement them and together they constitute Hadoop Ecosystem, as shown in Figure 1. Combined together, these tools provide services such as absorption, analysis, storage and maintenance of data etc.

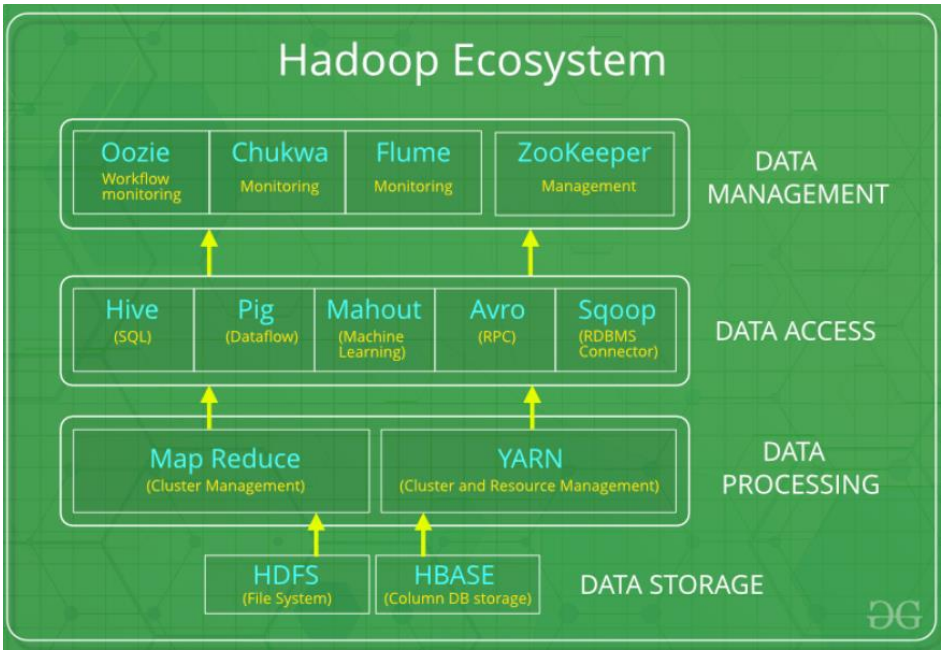

Figure 1 Hadoop Ecosystem (source: https://www.geeksforgeeks.org/hadoop-ecosystem/)

Hadoop is based on a storage component, called Hadoop Distributed File System (HDFS), and on a processing part which follows the MapReduce programming model. Operational wise, Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

© 5GENESIS Consortium

The 5GENESIS security analytics platform is based on Hadoop/HDFS (v.5.7 and above) for scalable and distributed storage of data (see Sec. 4.2.4).

3.1.1.2. Apache Kafka

Apache Kafka [14] is a distributed streaming platform, used for building real-time data pipelines and streaming apps. Kafka publishes and subscribes to streams of records, similar to a message queue or enterprise messaging system. It also stores streams of records in a fault-tolerant durable way and processes streams of records as they occur. Kafka can also connect to external systems (for data import/export) via Kafka Connect and provides Kafka Streams, a Java stream processing library. It can be used in two main categories of applications:

- Real time streaming data pipelines that get data between systems or applications.
- Real-time streaming applications that transform or react to the streams of data.

Kafka runs on cluster of one or more servers than can span multiple datacenters and stores streams of records in categories called topics. Each record is consisted of a key, a value and a timestamp.

Kafka has four core APIs:

- *Producer API*: allows an application to publish a stream of records to one or more Kafka topics.
- *Consumer API*: allows an application to subscribe to one or more topics and process the stream of records produced to them.
- *Streams API*: allows an application to act as a stream processor, consuming an input stream from one or more topics and producing an output stream to one or more output topics, effectively transforming the input streams to output streams.
- *Connector API*: allows building and running reusable producers or consumers that connect Kafka topics to existing applications or data systems. For example, a connector to a relational database might capture every change to a table.

As already discussed, Kafka topics are categories where streams of records are stored. Topics are always multi-subscriber; meaning that a topic can have zero, one, or many consumers that subscribe to the data written to it.

Two important APIs of Kafka that need to be explained further are producers and consumers.

- *Producers*: publish data to the topics of their choice and are responsible for choosing which record to assign to which partition within the topic. This can be done in a round-robin fashion to balance load or it can be done according to some semantic partition function (say based on some key in the record).
- *Consumers*: assign themselves a consumer group name and each record published to a topic is delivered to one consumer instance within each subscribing consumer group. Consumer instances can be in separate processes or on separate machines. If all the consumer instances have the same consumer group, then the records will effectively be load balanced over the consumer instances. If all the consumer instances have different consumer groups, then each record will be broadcast to all the consumer processes.

© 5GENESIS Consortium

The 5GENESIS security analytics platform, (see Sec. 4.2.3) uses Kafka to ingest data and transform/normalize them prior to their analysis, which is done using Spark (next section). In the project, Kafka is used as streaming platform in the sense that after the data collection, Kafka's role is to transfer these data into the HDFS for storage. Producers publish collected data to topics and then consumers are responsible for delivering each of these topics to a consumer group.

3.1.1.3. Apache Spark

Apache Spark [15] is a distributed and highly scalable cluster – computing framework. It provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. There are *four main modules*, which all are interoperable, meaning that data can pass between them:

- *MLib*: offers machine learning functionality.
- *GraphX*: offers big data in memory graph processing fast.
- *SQL*: provides the ability to process data in tabular forms and with tabular functions. It is integrated with Parquet and JSON formats in order for data to be represented in better formats and integrate with external systems. Moreover, Spark can be used by Hive as a processing engine.
- *Streaming*: data in Spark are processed as streams and cover a variety of topics such as transformations, output operations, etc.

The 5GENESIS security analytics platform, as detailed in Chapter 4, uses Spark to i) more quickly process and transform incoming data and ii) perform the core analytics functionality, detecting and classifying incidents. For the purposes of our work we mainly exploit SQL and Streaming modules. Version 2.1.0 and above of Spark is required.

3.2. Apache Spot

Apache Spot [16] is an open source software that leverages information from flow and packet analysis. It promotes threat detection and remediation using Machine Learning and integrates all security data into a comprehensive dashboard based on open data models. This ecosystem of ML-based applications can run simultaneously either on a single or a shared data set to provide enterprises analytic flexibility and thus the ability to discover suspicious connections and unseen attacks.

Spot architecturally is consisted of three parts, as shown in Figure 2:

- Data Sources: all the possible sources that provide data, either a streaming source or batch files.
- Spot GUI: all the services that constitute Spot's GUI and are responsible for visualizing results.
- Spot Landscape: the back-end part, which is responsible for operations like ingestion, transformation, machine learning, etc.

© 5GENESIS Consortium

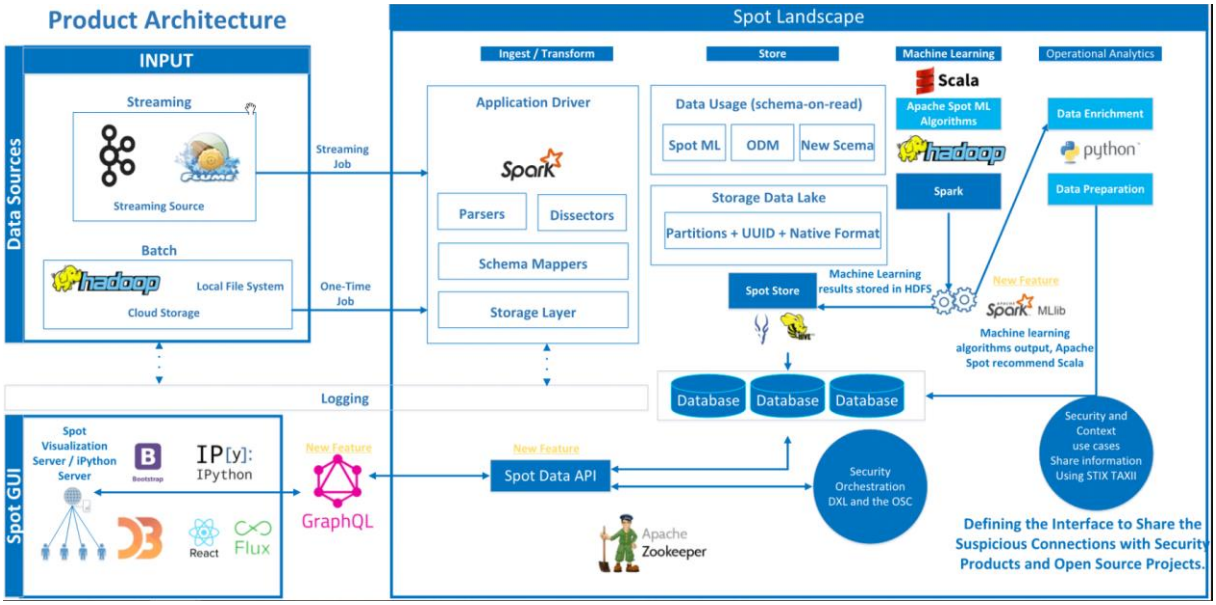

Figure 2: Architecture of Apache Spot[1]

In turn, Spot is based on Hadoop, Kafka and Spark (see previous sections) for data storage, ingestion and processing. As explained in more detail under Chapter 4, 5GENESIS inherits most of the architectural concepts of Spot and re-uses its data model, basic ML algorithm (Latent Dirichlet Allocation, LDA), as well as GUI for basic visualization in the flow processing pipeline. It also extends it with an additional algorithm, based on Autoencoder model.

3.3. Prometheus, InfluxDB and Grafana

This is the open-source technology stack used by the 5GENESIS Monitoring and Analytics framework, and has also been exploited for the metrics processing pipeline of the Security Framework.

Prometheus [18] is a platform for monitoring, focused on time-series data. It uses a pull model to collect metrics from agents/exporters.

InfluxDB [19] is a scalable database specifically tailored for time-series data.

Finally, Grafana [20] is a powerful visualisation platform for monitoring metrics.

The three tools are extensively presented in Deliverable D3.5 [21], as part of the 5GENESIS [22] Monitoring and Analytics stack.

[1] https://spot.apache.org/get-started/architecture/

4. Overview of the 5GENESIS Security Analytics framework

4.1. Overall architecture

The 5GENESIS 5G Security Analytics platform consists of a central data analytics engine and a distributed set of data collection components. Following a Big Data approach, the data value elicitation is divided in three categories, as shown in Figure 3 below:

1. Data acquisition, transformation and storage
2. Data analysis
3. Visualisation and export

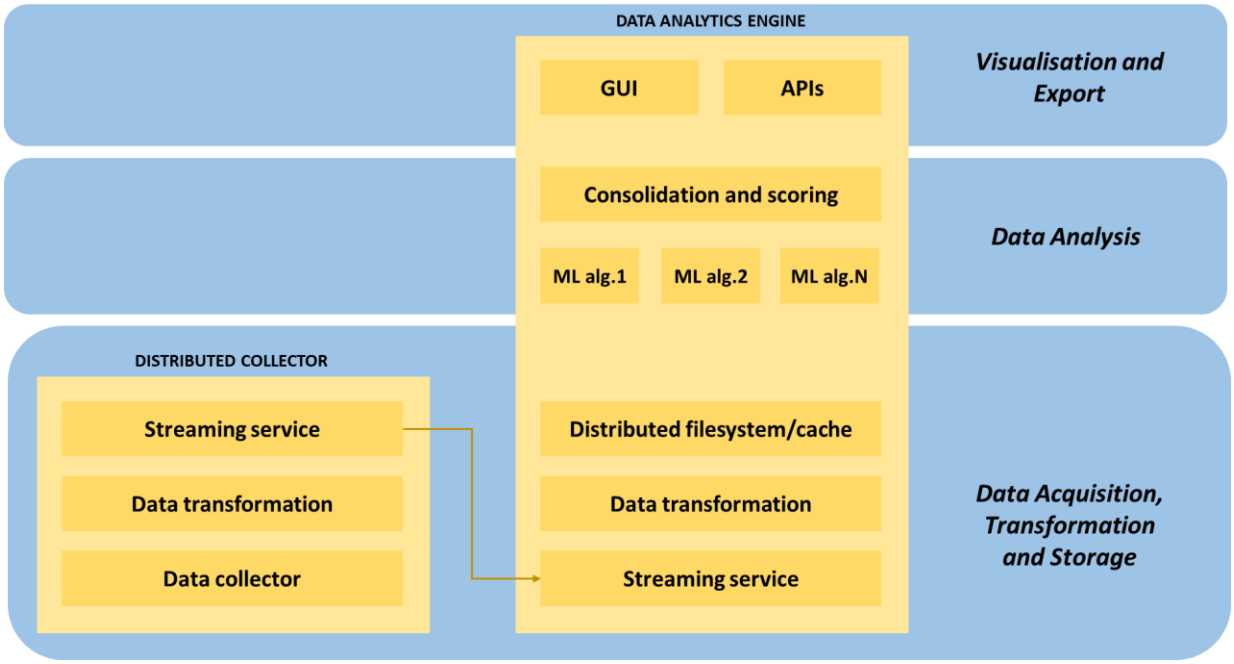

Figure 3: Functional components of the 5GENESIS Security Analytics platform

The three stages are common for the two pipelines (flow and metrics), yet the technologies employed vary. Below is a brief description of the involved subcomponents of the 5GENESIS Security Analytics framework, separate for each of the two pipelines.

4.2. Flow processing pipeline

4.2.1. Data collection

In the flow processing pipeline, the data collector is responsible for acquiring network flow (NetFlow) data generated from remote endpoints (VNFs, network elements, etc.). NetFlow

© 5GENESIS Consortium

data is either captured locally (e.g. in a VNF using dedicated collector agents), or received from a network element (e.g. firewall, router), which is capable of transmitting such information; most medium/high-end models found in the market already support this feature.

4.2.2. Data transformation

The data transformation stage is responsible for transforming the format-specific data into a structured, generic format. Data collection (see Sec. 4.2.1) and transformation can follow two approaches:

- Option 1 – Centralised collection and transformation: only the collection of the data is distributed, while all the other functionalities are centralised in the Data analysis phase.
- Option 2 – Distributed collection and transformation: the data collection and the data transformation are distributed at the collector agent and hence, the data is sent to the central engine in a standard format (e.g. CSV).

Data collection and transformation follows the procedures supported by Apache Spot, mainly for collecting NetFlow data and converting them to a structured format (CSV, tables). However, while the centralized architecture (Option 1) is the only method supported by Apache Spot, in 5GENESIS we take advantage of the distributed architecture (Option 2), as initially developed in the SHIELD project and further extended in 5GENESIS. It has been shown that the distributed approach can achieve, up to 10-fold decrease in processing time, compared to the centralized one.

4.2.3. Streaming service

The streaming service sends the information from the monitoring vNSF to the data analytics central engine, assuring reliability on the communication. The streaming service is based on Apache Kafka (Sec. 3.1.1.2.), thus, achieving both scalability and versatility.

The streaming service splits the network data into smaller specific topics and smaller partitions, while creating a data pipeline for each topic. The use of Kafka also allows the streaming stage to be reliable and fault tolerant for ensuring the integrity of the data and their quality in further processing steps.

4.2.4. Distributed File System/Cache

The Distributed File System / Cache is responsible for storing the collected data for both, batch (i.e. hard disks) or real-time (i.e. cache) processing. Once the network data has been transformed, the input is stored in a distributed file system in both the original and modified formats (in the case of centralized processing) or only the modified/preprocessed (in the case of distributed processing). The distributed file system is responsible for storing the collected data and making them available, so that it can be accessible by search queries.

For storing data, the Hadoop filesystem (HDFS) (See Sec. 3.1.1.1.) is used, achieving scalability, integrity, and better performance for bulk data exchanges. Raw data are saved directly in the

© 5GENESIS Consortium

filesystem, while structured (preprocessed) information are stored in Hive tables. Hive[2] uses the Hadoop filesystem and facilitates reading, writing, and managing large datasets residing in distributed storage using SQL.

4.2.5. Machine Learning algorithms

The Data Analysis phase features cognitive and analytical functionalities capable of detecting network anomalies that are associated with specific vulnerabilities or threats. The processing and analysis of large amounts of data is carried out by using Big Data analytics and machine learning techniques. By processing data and logs from probes/collectors deployed at specific strategic locations of the network, the data analytics framework can link traffic logs that are part of a specific activity in the network and detect any possible anomaly. In case malicious activity is detected, the relevant alarms are produced.

The set of ML algorithms is responsible for the detection of anomalies in network traffic that will lead to the prevention or mitigation of potential threats. The machine learning engine works not only as a filter for separating bad traffic from benign, but also to characterise the unique behaviour of network traffic. It contains routines for performing suspicious connections analytics on data (currently NetFlow) gathered from the Data Acquisition phase and the built-in Distributed storage system subcomponent. These analytics consume a collection of network events to produce a list of the events that are considered to be the least probable, and these are considered the most suspicious.

4.2.5.1. Latent Dirichlet Allocation

The statistical model that is used by Spot for discovering abstract topics of these events and ultimately discovering normal and abnormal behaviour is a topic modelling algorithm called Latent Dirichlet Allocation [17]. LDA is a generative probabilistic model used for discrete data that is applied to network traffic by converting network log entries into words through aggregation and discretisation, to discover hidden semantic structures. LDA is the built-in algorithm of Apache Spot (See Sec. 3.2). Spot executes LDA routines using a Scala Spark implementation from MLlib, Apache Spark's scalable machine learning library. It should be noted that Spot's current capabilities do not include any anomaly classification algorithms that would interpret the detected outliers as specific threats/attacks, thus such an algorithm will be originally developed to meet this requirement. The module will exploit Spot's existing batch processing capabilities, coupled with the development of streaming analytics functionalities currently missing from Spot, to achieve real-time (or near real-time) visibility for threat detection.

4.2.5.2. Autoencoder

In addition to LDA, in 5GENESIS the capabilities of Spot are extended with a new algorithm. A new method of anomaly detection has been proposed in order to improve the accuracy of Spot's built-in algorithm, reducing the time needed for prediction as much as possible. This is a Deep Learning model following an Autoencoder architecture. Autoencoders are suitable in

[2] http://hive.apache.org/

© 5GENESIS Consortium

cases, where anomalies must be found. They usually are trained with normal only traffic or with normal traffic which contains a small number of anomalies. Autoencoders compress the given input into a smaller representation and then they try to decompress it and reproduce the input. The error between the given input and its output is very small under normal circumstances. If the error is above a threshold the input will be considered as anomaly. In this case the given connections with a big reconstructing error will be marked as suspicious.

The proposed Autoencoder has been integrated in Apache Spot's pipeline, replacing built-in LDA algorithm. It takes as input network flows, using 18 features from the flow's metrics. These metrics are flow's protocol & duration, existing flags, source & destination ports, and number & size of both incoming and out coming packets. Its architecture is very simple, in order to be as lightweight as possible and to reduce the training time. It consists of three fully connected layers. The first layer has 18 neurons to retrieve the input, the second layer has 12 neurons in order to compress the given input and the final layer has also 18 neurons in order to decompress and recreate the input. As activation function *tanh* has been selected and Adam has been selected as optimizer. The model has been trained for 90 epochs. In order to reduce the needed resources for both training and time a Spark library, Elephas[3], has been selected so both training and predicting are using Spot's deployed Spark cluster for their operations. Also, with the usage of Spark all functions of the model are parallelized, so that the existing resources are better initialized.

4.2.6. GUI

For the visualization of information from the flow processing pipeline, 5GENESIS uses the GUI of Apache Spot.

The main view of the GUI is depicted in Figure 4 below and aims at the visualization of the suspicious flows. The latter are displayed as a list (with the most suspicious ones at the top) as well as in a graph.

The operator is also allowed to manually score the detected flows, thus helping the algorithm to learn and eventually minimize false positives.

[3] https://github.com/maxpumperla/elephas

© 5GENESIS Consortium

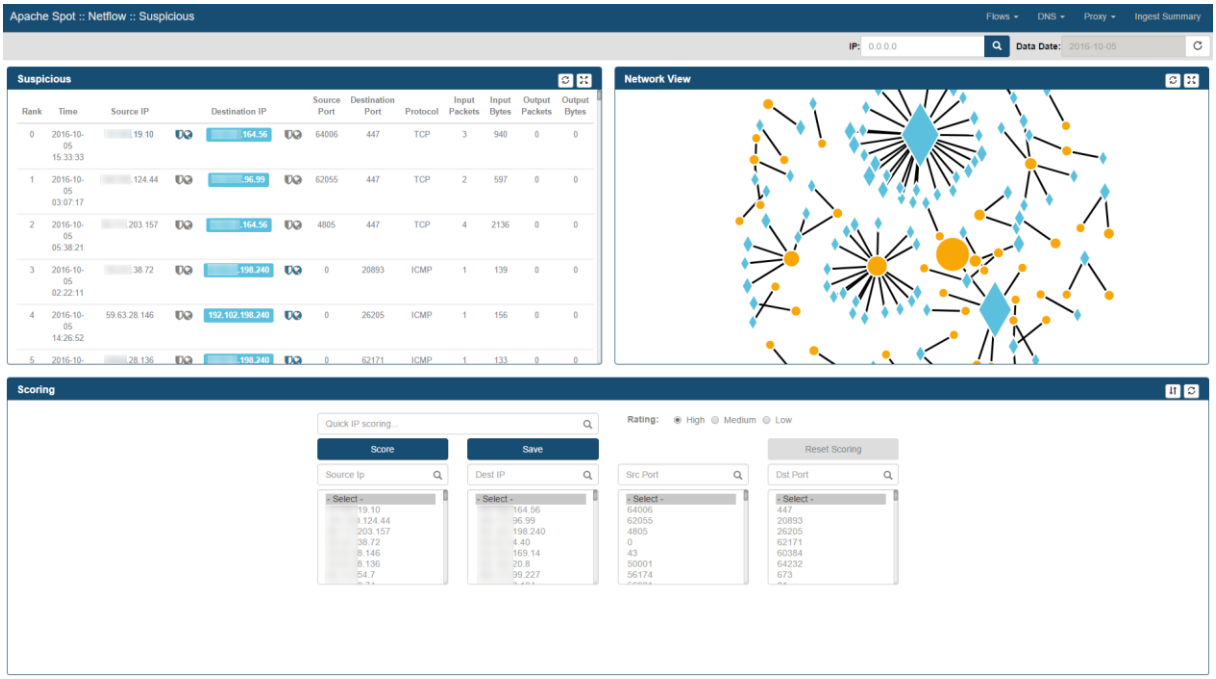

Figure 4. Apache Spot GUI (Suspicious connects view)

4.3. Metrics processing pipeline

4.3.1. Data collection

The Metrics Processing Workflow collects metrics from two different data sources (currently RAN and edge compute node), predicting which of these metrics are part of an anomaly and visualizes the detected anomalies in a web interface.

It is currently deployed across three main components: 5G RAN (based on Amarisoft Callbox model in our configuration), Edge Compute Node (based on a Dell Edge Gateway platform) and Core DC, as shown in Figure 5.

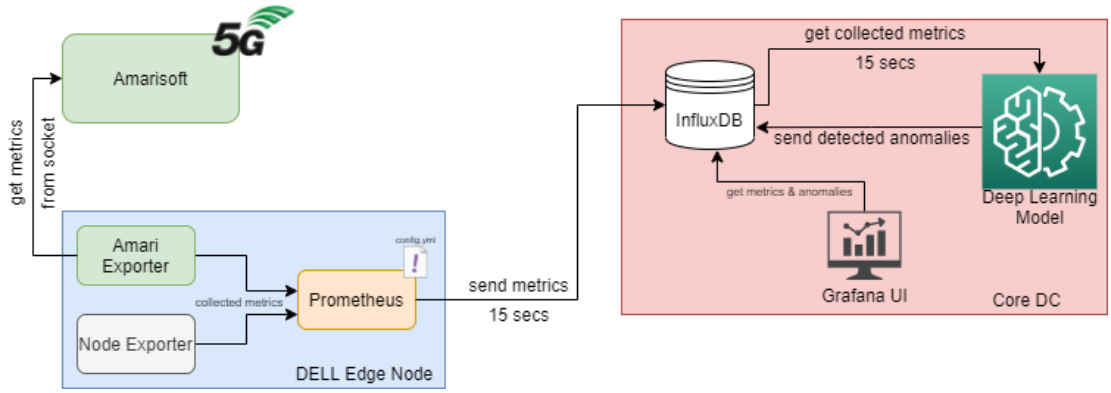

Figure 5 Metrics Processing Workflow

In Metrics Processing Workflow there are two main data collectors: Node Exporter and Amari Exporter. These data collectors are responsible for collecting metrics from the Edge Node and RAN respectively. Collected metrics from both exporters are saved in Prometheus, which is running in the Edge Node. Alternatively, Prometheus can be deployed at the core.

Other data collectors can also be used as long as they send collected data in appropriate format to Prometheus. For 5G metrics, the Anomaly Detection algorithm can work with other vendors as well, provided that a compatible exporter is developed in order to send collected metrics to Prometheus in expected format. The Node Exporter can also be deployed in any other Linux machine, because it only collects system metrics from this node. In case that there are more than one edge nodes, the Node Exporter can be deployed in all of them, but the data that will be sent in Prometheus must be aggregated.

4.3.1.1. Node Exporter

The Node Exporter is running in the Edge Node. It is a Prometheus component developed in Go programming language for collecting Linux system metrics[4]. It collects system metrics, which can be useful in identifying attacks in 5G edge nodes. It provides some information about the node running such as OS name and version and system architecture. Its main focus is to collect system metrics about node's CPU, the available and used memory, the number of available bytes from node's file system and swap partition and network metrics, such as the number of transmitted and received bytes from all network interfaces in the machine.

4.3.1.2. Amari Exporter

The Amari Exporter also is running in the Edge Node. It was developed for the needs of 5GENESIS, in Go programming language and its main focus is to collect 5G metrics from Amarisoft Callbox RAN. Amarisoft RAN and Amari exporter are communicating through a Go implementation of Websocket protocol[5]. Some of the collected metrics are downlink & uplink bitrate, RX/TX CPU usage and RX-TX delay. Amari Exporter uses Prometheus library[6] in order to transform and send collected metrics to appropriate format for Prometheus.

4.3.1.3. Prometheus

Prometheus must be deployed with a YAML file, which contains required configurations, such as data sources for data, targets where the collected data will be sent, the time interval for sending the data to targets, etc.

4.3.2. Data transformation

Different data sources can be used with the Metrics Processing pipeline. Different 5G radios can be used as long as the collected metrics are transformed to a specific format. Also, more than one edge nodes can be monitored if the collected metrics from all of these nodes are

[4] https://github.com/prometheus/node_exporter
[5] https://github.com/gorilla/websocket
[6] https://github.com/prometheus/client_golang/

© 5GENESIS Consortium

aggregated. Use of Prometheus is not mandatory for Anomaly Detection algorithm to run, but it is required for Grafana UI. The Anomaly Detection algorithm fetches records from InfluxDB. Currently, the fields illustrated in Figure 6 are supported.

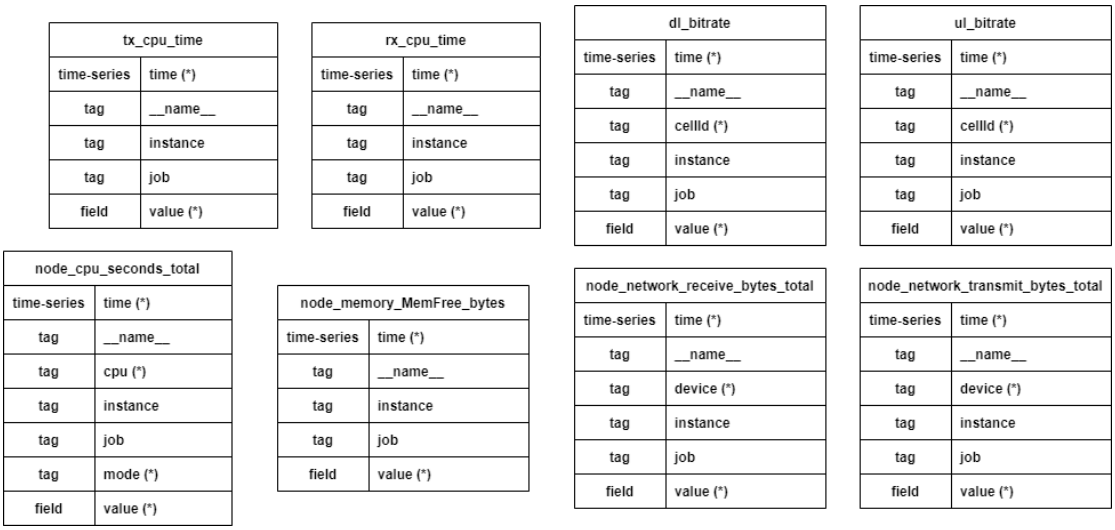

Figure 6 InfluxDB fields

From the collected metrics from the two exporters some new features are created, that will be used for both training and testing the model. The collected metrics are resampled every 15 seconds, in order to synchronize them, because there may be a small difference in timestamps used by the two exporters. Before the model is trained, a dataset of collected data with normal traffic is preprocessed in order to extract the required features for the model to be effective. The metrics and the extracted features that used are the following:

- *CPU seconds from edge node:* this metric is collected as a counter, which increases in each record. This counter is separate for each core and for each CPU mode. Using the time difference between the two records and the counter difference, the percentage of CPU used in user mode can be extracted for each core separately. Getting the mean value of all cores, the mean CPU percentage in user mode is calculated. From this metric, the rate of mean percentage of CPU usage in user mode is extracted.
- *Free Memory bytes:* using free bytes and total bytes of memory, the percentage of memory that is used can be calculated. Using the calculated percentage the rate of percentage of used memory can be defined.
- *Network receive and transmit bytes total:* these metrics are also collected as counter. For every interface there is a counter for received and transmitted bytes. Converting these counters to bitrate, the average upload and download bitrate are calculated. From the calculated mean bitrate, the difference between two records is used.
- *RX/TX CPU time:* these metrics are stored as percentage of usage for RX CPU usage and TX CPU usage respectively. From these metrics, the CPUs percentage of usage rate is extracted and used.
- *5G Uplink/Downlink bitrate:* these metrics describe the bitrate of 5G Uplink and Downlink radio data channels.

The extracted features are normalized using Min-Max Normalization. The values used for normalization are saved in a JSON file in order to normalize real time incoming data. After normalization, the dataset is split in sequences, with four steps per sequence, where the first

© 5GENESIS Consortium

three records will be used to predict the fourth. When the model is running for detecting anomalies, the same data transformations is used, in batches of incoming data.

Prometheus leverages the InfluxDB API for writing data into InfluxDB. Prometheus sends POST requests to the /write endpoint, including the time-series data gathered from the exporters. For every metric, a measurement is created on the InfluxDB side. Influx uri, database and credentials are required for the remote write and are part of the prometheus.yml config file in the following format :

```
 # Remote write configuration for Influxremote_write:
  -                                                                    url:
"http://10.10.X.X:8086/api/v1/prom/write?db=database_name&u=username&p=pass
word"
```

4.3.3. Data streaming

Every 15 seconds a new record is fetched from Influx DB. This record is added to an array, which holds the fetched records that are split in sequences and used for prediction. For implementing this array, a sliding time window is used, which keeps only the last n records (n was set to 30 but it is configurable). From the new record, the required features are extracted and normalized.

4.3.4. Deep Learning (DL) model

4.3.4.1. DL Model Training

For training, the architecture of an Autoencoder has been used. The designed architecture, as shown in Figure 7, has 9 Bidirectional LSTM layers and one Dense layer at the end. The choice of LSTMs has been made due to the nature of the data. The incoming data are multiple time series, where both previous and next values of the series are relative. For this reason, the Bidirectional variance of LSTMs has been chosen.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_1 (Bidirection (None, 4, 128)            37376
_________________________________________________________________
bidirectional_2 (Bidirection (None, 4, 64)             41216
_________________________________________________________________
bidirectional_3 (Bidirection (None, 4, 32)             10368
_________________________________________________________________
bidirectional_4 (Bidirection (None, 4, 16)             2624
_________________________________________________________________
bidirectional_5 (Bidirection (None, 4, 8)              672
_________________________________________________________________
bidirectional_6 (Bidirection (None, 4, 16)             1088
_________________________________________________________________
bidirectional_7 (Bidirection (None, 4, 32)             4224
_________________________________________________________________
bidirectional_8 (Bidirection (None, 4, 64)             16640
_________________________________________________________________
bidirectional_9 (Bidirection (None, 128)               66048
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 1032
=================================================================
Total params: 181,288
Trainable params: 181,288
Non-trainable params: 0
_________________________________________________________________
INFO:root:None
```

Figure 7 Deep Learning Model Architecture

As activation function ReLu has been used. As optimizer SGD (Stochastic Gradient Descent) with Nesterov momentum has been selected with learning rate equal to 0.01. The model is trained for 50 epochs. The dataset with only normal traffic, which is used for training, contains 8734 records. 10% of training dataset is used for evaluating the model during training operation.

4.3.4.2. DL Model Testing

The trained model fetches the new data in real time, as explained in 4.3.3, and extracts the required features, as described in 4.3.2. After the feature extraction, the new data are normalized using Min-Max normalization, with the stored min and max values for each feature from training. Then the batch of data is split into sequences of 4 records, where the first three are used to predict the 4th. The trained model will use the first three records from each sequence, in order to predict the 4th record. If the sequence consists of records of normal traffic the root mean squared error (RMSE) between the predicted and actual values will be very small. If the RMSE is above a threshold, then the sequence will be marked as anomaly. The threshold for each feature and an aggregated threshold for all features can be set either by the user directly or it can be generated through the training process. In the last case, two datasets containing attack records will be used for evaluation and for generating some anomaly thresholds. In either case, the thresholds must be saved in a JSON file, and they are loaded during the execution of the algorithm.

4.3.5. GUI

A UI for metrics processing workflow has been deployed using Grafana. The provided UI can be used for both monitoring some useful metrics and also list all detected anomalies, as shown in Figure 8.

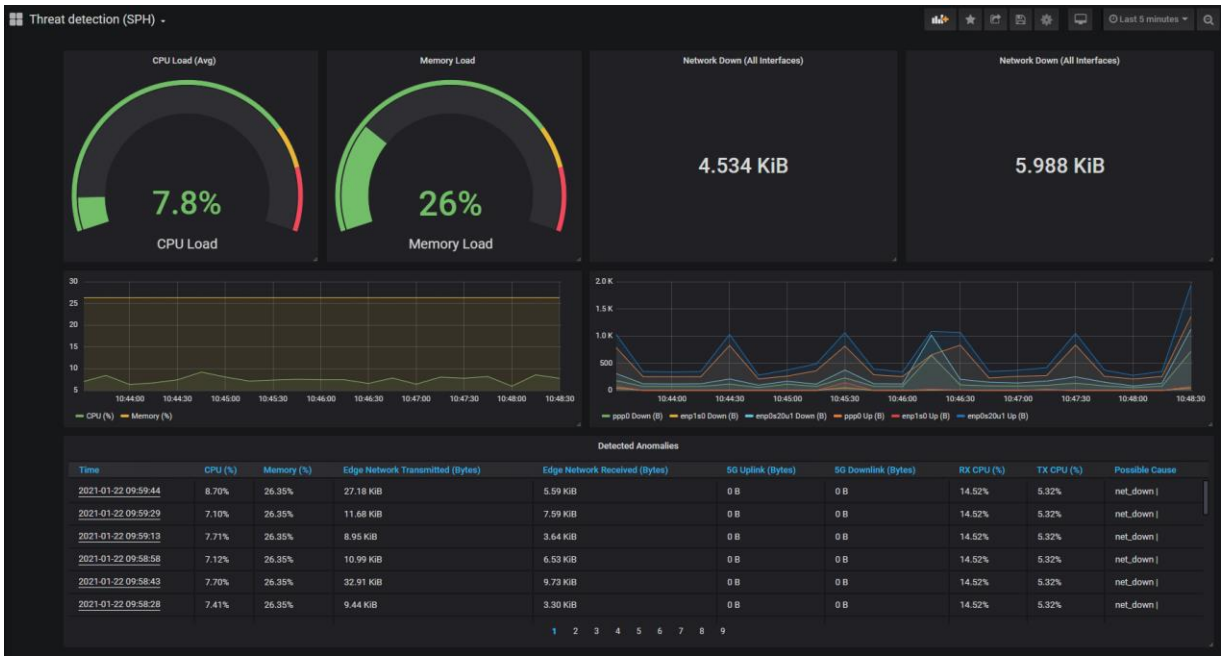

Figure 8 Metrics Processing Workflow UI

As is shown in Figure 8, in our configuration there are 6 panels for monitoring and a table with detected anomalies. The first two panels in top row are monitoring the percentage usage for all CPU cores in average and RAM. The next two panels in the top row are monitoring the sum of all received and transmitted bytes for all network interfaces. In the second row there are two panels. The left one shows the percentage of used CPU and RAM and the second one the network bytes that received and transmitted from three specific interfaces. Finally, in the bottom row there is the table with all detected anomalies. The table contains the time the anomaly detected and values for some features the time of anomaly, like CPU and RAM usage, network metrics and 5G metrics. Also, there is a message with possible causes if the cause of anomaly can be detected.

Of course, this is only an indicative Grafana configuration corresponding to the setup we used; this can be configured as needed, corresponding to the nodes monitored and the metrics collected.

© 5GENESIS Consortium

5. TESTING AND EVALUATION

5.1. Evaluation scenarios

5.1.1. Flow processing pipeline

In the flow processing pipeline, the Autoencoder algorithm developed within 5GENESIS has been evaluated compared with Spot's built-in LDA algorithm. The IDS 2018 Dataset[7] has been used for both training and evaluation for the Autoencoder. This is a dataset, which includes several different attacks. One day of the dataset has been used for training and 4 other days have been used for evaluation of the two algorithms. As evaluation metrics accuracy, precision, recall and F1 score have been selected.

5.1.2. Metrics processing pipeline

In the metrics processing pipeline, for evaluating the Deep Learning model, which is used for detecting anomalies three scenarios have been selected. In first scenario normal traffic is simulated in real time, using the records from training set. The second and third scenario simulate two different types of attacks in the edge machine. More specifically:

- The second scenario simulates a CPU overload caused by an attacker hijacking the edge node (Infrastructure compromise) and draining its resources to cause a DoS incident.
- The third scenario simulates an eavesdropping incident, caused by an attacker hijacking an edge VNF (Service compromise) and modifying it to replicate the user traffic and send it to another destination. This behaviour is emulated by initiating a second data stream (generated by iperf3[8]) between the edge node and the network core.

As evaluation metrics the prediction RMSE has been used. In the infrastructure compromise attack (CPU overload) the CPU usage has been affected. In the service compromise (eavesdropping) attack, the network and 5G data rates have been affected. All the above metrics are visualized, in order to check the prediction error of the algorithm in normal traffic.

5.2. Results and discussion

5.2.1. Flow processing workflow

In Figure 9 the accuracy and loss during Autoencoder training is shown.

In Figure 10, Figure 11, Figure 12 and Figure 13 the accuracy, precision, recall and F1 score of LDA and Autoencoder are shown. It is shown that Autoencoder can perform better compared to Spot's built-in LDA algorithm. Its accuracy is around 95%, while LDA's accuracy is around 50%-52%. In terms of precision and recall proposed Autoencoder can achieve recall from 75%

[7] https://www.unb.ca/cic/datasets/ids-2018.html
[8] https://iperf.fr/iperf-download.php

© 5GENESIS Consortium

up to 100% and precision from 10% up to 35%. On the other hand, LDA can achieve precision up to 3% and recall from 13% up to 50%. Autoencoder has been preferred for Spot's anomaly detection algorithm compared with LDA due to the big improvement it can achieve in performance. Also, using Autoencoder the time needed for anomalies to be predicted is significantly reduced, because of the type of the two algorithms. Autoencoder is a trained model, which needs a little time to predict incoming records. On the other hand, LDA is a probabilistic algorithm, which needs to run every time and do a lot of calculations in the batch of the incoming data in order to decide which net flows are normal and which are anomalies.

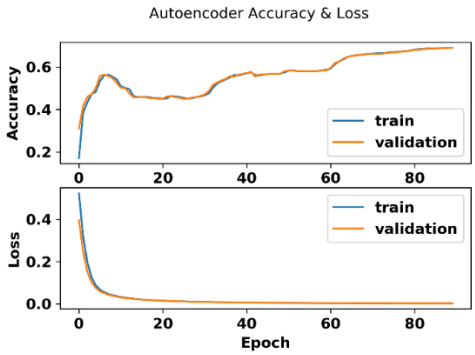

Figure 9 Autoencoder Training Loss & Accuracy

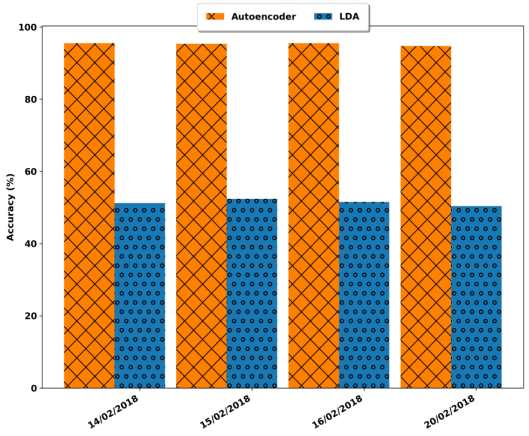

Figure 10 Autoencoder & LDA Accuracy

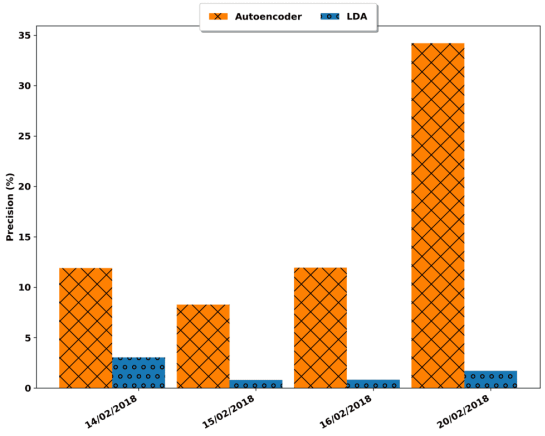

Figure 11 Autoencoder & LDA Precision

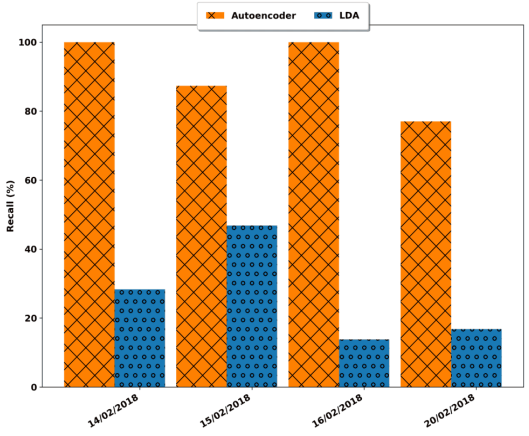

Figure 12 Autoencoder & LDA Recall

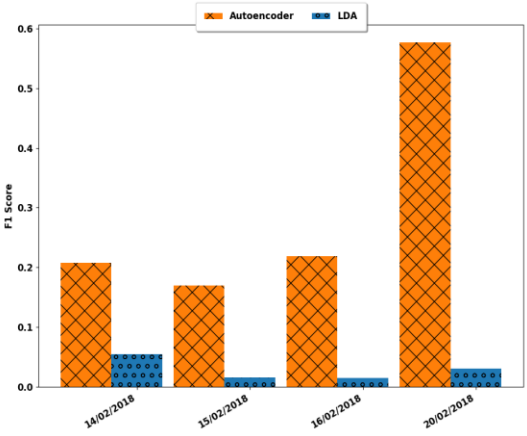

Figure 13 Autoencoder & LDA F1 Score

© 5GENESIS Consortium

5.2.2. Metrics processing workflow

In Figure 14 the loss of the Deep Learning Model during training is shown.

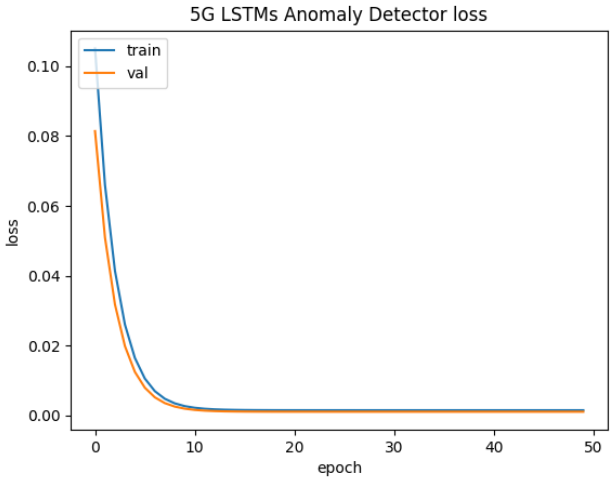

Figure 14 Training Loss

Figure 15 and Figure 16 show the results from first scenario (normal traffic). In this scenario the model gets the training data simulating the real time traffic. The size of window size is equal with the size of time window in training, which is equal to 30. In the left figure, the prediction error (RMSE) for each of edge metrics is visualized. It is shown that in almost all cases the RMSE is very low. In the right plot, which shows the predictions' error for 5G metrics, RMSE is also very low except for 2 sequences of records, where the error is big.

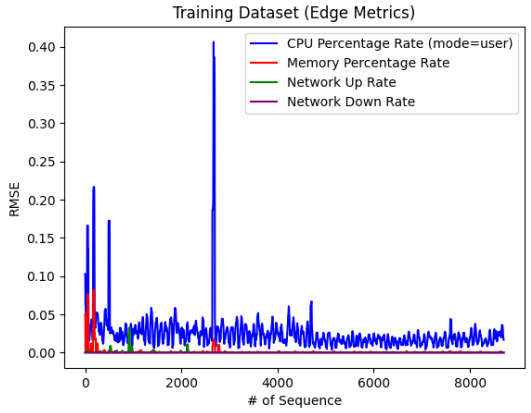

Figure 15 Prediction Error in Training Data (Edge metrics)

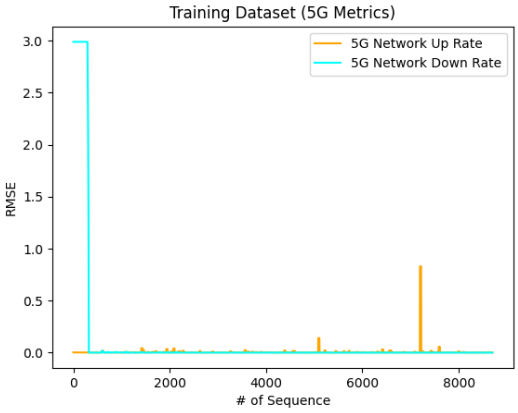

Figure 16 Prediction Error in Training Data (5G metrics)

In Figure 17, the prediction error in second scenario, which is a CPU overload attack, is shown. Only CPU prediction error is shown, because the rest of the metrics are not affected from this type of attack. As the figure shown the trained model has a small error except for two cases. These two spikes in the plot show the start and the end of the CPU overload attack. So, the trained Autoencoder has detected the attack and an anomaly entry is correctly inserted in Influx DB and shown in Grafana UI.

© 5GENESIS Consortium

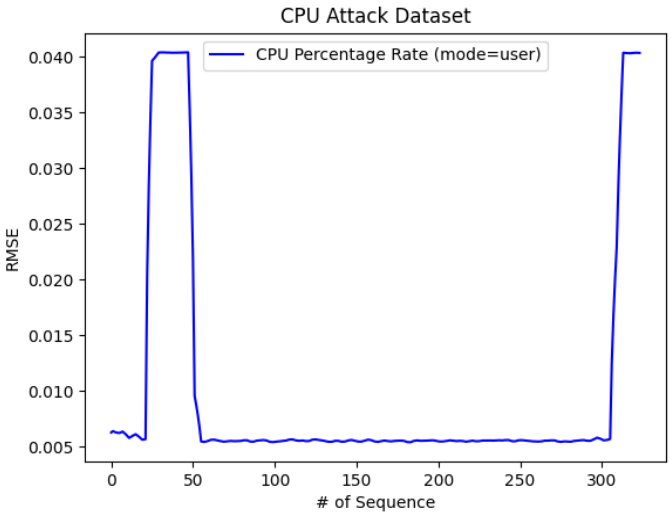

Figure 17 Prediction Error in CPU Overload Dataset

In Figure 18 and Figure 19 the results of third scenario are shown (eavesdropping simulated by iperf). This affects mainly the network connections. In the left plot, the prediction error for edge network metrics is shown and in the right plot the prediction error for 5g network metrics is shown. It is shown in both images that a network attack has been detected around the middle of the captured data, corresponding to one or two big spikes in both figures.

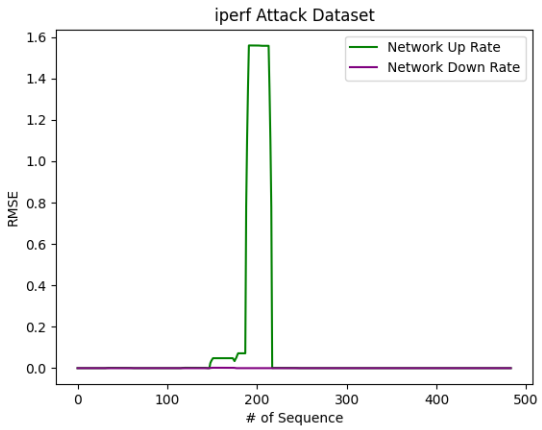

Figure 18 Prediction Error in iperf Dataset (Edge metrics)

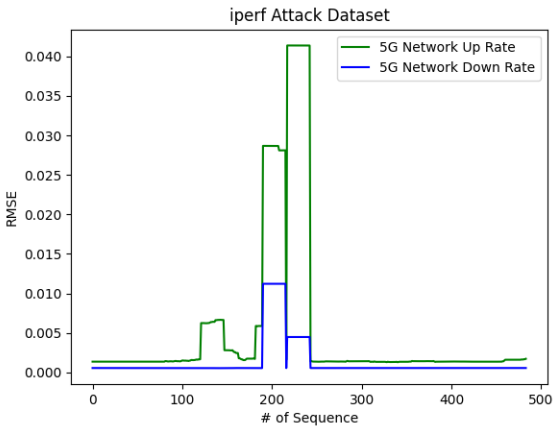

Figure 19 Prediction Error in iperf Dataset (5G metrics)

© 5GENESIS Consortium

6. CONCLUSIONS

Reinforcing security is crucial for the viability of next-generation 5G networks. The security analytics solution integrated in the 5GENESIS facility intends to contribute towards this direction. The 5GENESIS Security Analytics platform is established on proven technologies and has the potential to constitute a scalable and efficient solution for promptly detecting and classifying security incidents. The two pipelines developed address network level attacks as well as breaches in the compute infrastructure and services. The collection, storage and processing tasks rely on proven technologies, which guarantee the scalability and extensibility of the solution. In addition, the Machine Learning models developed and trained exhibit a very promising performance against a wide variety of threats. In the last phase of the project, the Security Framework will operate as an integral component of the Limassol and Athens 5G platforms, and will be further trained and fine-tuned according to more diverse and realistic scenarios.

© 5GENESIS Consortium

REFERENCES

[1] 5GENESIS Consortium, "D2.1 Requirements of the Facility," 2018. [Online]. Available: https://5genesis.eu/wp-content/uploads/2018/11/5GENESIS_D2.1_v1.0.pdf

[2] 5GENESIS Consortium, "D2.4 Final report on facility design and experimentation planning," [Online]. Available: https://5genesis.eu/wp-content/uploads/2020/07/5GENESIS_D2.4_v1.0.pdf

[3] 5GENESIS Consortium, "D2.3 Initial planning of tests and experimentation," [Online]. Available: https://5genesis.eu/wp-content/uploads/2018/12/5GENESIS_D2.2_v1.0.pdf.

[4] 5GENESIS Consortium, "D3.13 5G Security Framework – Release A" [Online], Available: http://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.13_v1.0.pdf

[5] 5G Security Landscape, June 2017, 5G-PPP, https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP_White-Paper_Phase-1-Security-Landscape_June-2017.pdf

[6] ENISA Threat Landscape for 5G Networks [Online], Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-for-5g-networks

[7] SHIELD project, https://www.shield-h2020.eu/

[8] PALANTIR project, https://www.palantir-project.eu/

[9] S. Omar, A. Ngadi, and H.H. Jebur, "Machine learning techniques for anomaly detection: An overview", International Journal of Computer applications, vol. 79, No 2, 2013

[10] M Umer, M. Sher, and Y. Bi, "Applying One-Class Classification Techniques to IP Flow Records for Intrusion Detection", Baltic Journal of Modern Computing, num 1, pp 70-86, http://dx.doi.org/10.22364/bjmc.2017.5.1.05

[11] H.Kim, K.Claffy, M.Fomenkov, D.Barman, M.Faloutsos, K.Lee: "Internet traffic classification demystified: myths, caveats, and the best practices." In Proc. of ACM CoNEXT, 2008 Madrid, Spain

[12] Y.Lim, H.Kim, J.Jeong, C.Kim, T.Kwon, Y.Choi: "Internet traffic classification demystified: on the sources of the discriminative power." 2010, In CoNEXT, pg. 9

[13] Apache Hadoop, https://hadoop.apache.org/

[14] Apache Kafka, https://kafka.apache.org

[15] M. Frampton, Mastering Apache Spark, Packt Publishing Ltd., 2015

[16] Apache Spot (incubating), https://spot.apache.org/

[17] D. Blei, A. Ng, M. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022 , 2003.

[18] Prometheus, from metrics to insight, https://prometheus.io/

[19] InfluxDB: Purpose-built, open-source time-series database, https://www.influxdata.com/

[20] Grafana, the open observability platform, https://grafana.com/

[21] 5GENESIS Consortium, "D3.5 Monitoring and Analytics – Release A" [Online]. Available: https://5genesis.eu/wp-content/uploads/2019/10/5GENESIS_D3.5_v1.0.pdf

[22] H. Koumaras et al., "5GENESIS: The Genesis of a flexible 5G Facility," 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Barcelona, Spain, 2018, pp. 1-6, doi: 10.1109/CAMAD.2018.8514956.

© 5GENESIS Consortium