Cobra-5G – an AI-driven Solution for Resilient Industrial Applications in Private 5G Environments

Sebastian Peters*, Fikret Sivrikaya*, Satyatma Winarga Rochadi†, Amina Ayadi-Miessen‡

*GT-ARC gGmbH, Berlin, Germany – {sebastian.peters, fikret.sivrikaya}@gt-arc.com
†DAI-Labor, Technische Universität Berlin, Berlin, Germany – satyatma.rochadi@dai-labor.de

‡HMF Smart Solutions GmbH, Bad Münder, Germany – amina.ayadi-miessen@hmf-germany.com

***Abstract*—The proliferation of private 5G networks requires novel means to support the owner in its operation. This mandates keeping the 5G network fulfilling the specific application scenario secure and ensuring an overall resilient deployment. Along these lines this paper presents a novel NWDAF-based user session anomaly detection approach, contributing with a blueprint for the tight coupling of a 5G-based private network deployment with an example industrial use case. We therefore exploit the prior knowledge of user sessions in the industrial context and utilize the metadata exposed by 5G core network functions amending the open source Open5GS core. We present two ML models trained on the obtained metadata of PDU session and user traffic patterns, which showed a high accuracy in detecting anomalies when fed with our implemented anomaly generator component.**

I. Introduction

The widespread commercial availability of 5G technology - manifested by a plethora of 5G end-user devices, radio access infrastructures and 5G core network implementations - has made it viable to deploy small scale, purpose-built private 5G networks. In addition, since 5G's technological foundations are built on web (Rest API) and cloud technologies (virtualized network functions), the private 5G network can be easily dimensioned and deployed to fit the operation to the intended environment, with the end-to-end communication all being under the control of the owner of the 5G infrastructure, as compared to connecting via public Mobile Network Operators (MNOs). Based on this outlook, market research firms are expecting high growth rates of private 5G deployments in the years to come[1].

However, in order for these growth expectations to materialize, we believe that several challenges need to be overcome. First, it will be challenging, e.g., for the manufacturing industry unacquainted with 5G technology, to operate and maintain the private 5G network, as highly specialized staff - similar to the engineers working in the Network Operations Center (NOC) of mobile network providers - need to be hired to ensure the seamless operation across all network segments from radio access to backend applications. Secondly, a private 5G deployment needs to support the improvement of a legacy industrial manufacturing process in order to justify the investment. Both challenges may be addressed by i) better supporting non-expert workers in operation of a private 5G network, making it a secure and resilient deployment, and ii) by facilitating a tighter integration of the 5G end-to-end application with the industrial process, fostering a tailored and highly specialized private 5G network.

Attacking these challenges is the overall goal of the Cobra-5G[2] research project, funded by the German Federal Office for Information Security (BSI) in the "Cyber Security and Digital Sovereignty in 5G/6G Communication Technologies" program.

A. Focus and Contribution

Embedded into the overall Cobra-5G approach, this paper contributes an AI-driven solution to detect abnormal behavior of user sessions, utilizing data collected via the Network Data Analytics Function (NWDAF) from 5G core network functions and the user plane, feeding it into machine learning models that are tailored to the industrial application that is executed on the factory floor. In doing so, this paper contributes with a blueprint for the tight coupling of a 5G-based private network deployment with an example industrial use case. To achieve this, we created an NWDAF implementation to demonstrate this use case, particularly detecting abnormal UE traffic patterns by involving two machine learning models to detect anomalies in UE Protocol Data Unit (PDU) session statuses and throughput. Based on our approach, we further contribute with a proposed addition to 3GPP's definition of abnormal UE behaviour analytics.

II. Cobra-5G and 5G Background (NWDAF)

The Cobra-5G project aims at gaining new insights into private 5G networks (also referred to as 5G campus networks in Germany), bringing partners from research, infrastructure supplier and system integrator perspectives together to attain resilient and secure private 5G deployments. Figure 1 shows the high-level areas covered within the project. In brief, the **Co**ntainer-**b**ased **r**esilient **a**rchitecture accepts the fact that private 5G networks are characterized by containerized

[1]https://www.grandviewresearch.com/industry-analysis/private-5g-network-market

[2]https://www.cobra-5g.de/

979-8-3503-5469-0/24/$31.00 ©2024 IEEE

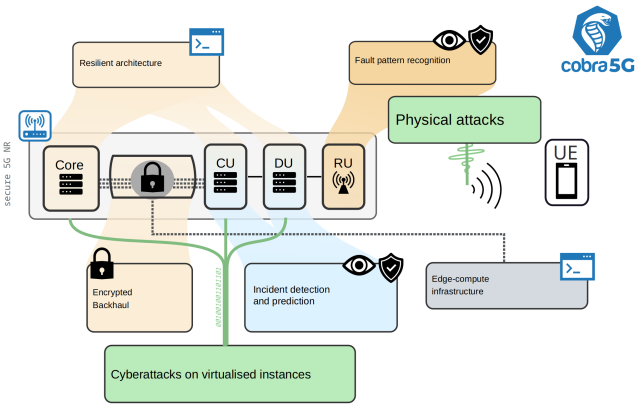

Fig. 1. Overview of the Cobra-5G Project

environments across all network segments from RAN to back-end, needing protection from physical and cyber attacks and requiring targeted measures to increase their overall resilience. To attain this goal, the project pursues a range of AI-based detection approaches. The focus of this work lies in exploiting the capabilities of the 5G core network and its network data analytics function to detect anomalies in the behavior of users connected to the private 5G network.

A. The Network Data Analytics Function (NWDAF)

The 3GPP has devised the NWDAF as the key network function to collect and store relevant information pieces of an end-to-end 5G system, serving as a central instance for arbitrary pieces of information provided to AI-based approaches. It provides analytics, which can be either statistical information of past events or predictive information, to other Network Functions (NFs), Application Functions (AF), and Operations, Administration and Maintenance (OAM). In [1], the 3GPP standardised the interface for deriving these analytics, detailing the required data, its source, and the corresponding service to contact. 3GPP Release 17 specifications define two main functional components that NWDAF instances can implement: Analytics Logical Function (AnLF) and Model Training Logical Function (MTLF). While the former provides support for inference and exposes analytics services, the latter enables the training of new machine learning models and thus providing new trained ML models for exposure to other NFs and AFs. Given the unlimited possibilities of data feeding into NWDAF, its fundamental perspective goes beyond the traditional network management systems that facilitate the operation and maintenance of the deployed network. By taking this perspective novel AI-driven approaches are enabled to exploit data that add to the knowledge about 5G networks and the use cases utilizing them.

In the present work we have extended the relevant network functions of the Open5GS Core Network with the capabilities to expose user and session metadata of interest to NWDAF, in order to facilitate an anomaly detection of potentially erroneous behavior regarding a specific application, visible from aberrant user communications.

B. State of the art in NWDAF Abnormal UE Behaviour Analytics

The NWDAF 3GPP specification [1] (Sec. 6.7.5.1) already standardizes abnormal traffic patterns as an example of abnormal UE behaviour and correspond to “Unexpected long-live/large rate flows” and the “too frequent service access” exceptions. However, the abnormal UE behaviour targeted in this paper, namely irregular PDU session patterns and low throughput rates do not fit neatly into any of the exception IDs standardized so far. We therefore propose to introduce a new exception category: “Unexpected UE Traffic Pattern”. The expected UE behaviour parameter to be provided would then be the expected UE traffic pattern. This exception ID may be particularly relevant for private 5G network deployments, where operators have detailed knowledge of predictable UE traffic patterns.

III. Related Work

In this section we present the related work on private 5G networks in industrial contexts, works involving the Network Data Analytics Function (NWDAF) as well as those combining the NWDAF with anomaly detection approaches.

In general private 5G networks address the specific needs of enterprises across various sectors, including healthcare and industrial Internet of Things (IIoT) [2]. Typically, these enterprises deploy their private 5G networks entirely independent of public networks to meet stringent QoS requirements and enable independent network operations [3]. However, one of the main challenges in private 5G deployments is the lack of skilled network professionals [2], where increased intelligence to optimize operations of the network is a potential solution. In the following we present relevant approaches implementing the NWDAF and applications of anomaly detection in this connection.

A. Implementations of the Network Data Analytics Function

In [4] Manias et al. introduced their NWDAF implementation with a focus on analyzing core network signaling traffic, emulating a 5G core architecture using Open5GS and UERANSIM. Within this architecture, each NF operates on a separate virtual machine, and all internal network traffic is duplicated and routed to a central host, serving as the primary hub for NWDAF analytics and operations. The authors categorized the collected packets based on their originating and destination NFs and extracted various statistics such as average packet length, maximum packet length, standard deviation of packet lengths, and total packet count for each NF-to-NF interaction. These statistics served as features for clustering NF-to-NF interactions using different k values. Notable clusters identified included NF-NF pairs with no packet exchange, NF-NF pairs with packet exchange, further subdivided based on maximum packet length, and interactions involving the NRF. By categorizing NF-to-NF interactions,

the authors foresee several insights and opportunities, including facilitating intelligent networking decisions, implementing proactive resource allocation strategies based on monitored NF resource requirements, and constructing network models for anomaly detection tasks to detect and mitigate abnormal network conditions. Ultimately, the study aimed to support Management and Orchestration (MANO) in their decision-making processes regarding network management and optimization.

Kim et al.'s [5] NWDAF implementation comprises both MTLF and AnLF components. Although the service API (Nnwdaf AnalyticsInfo) of the AnLF was generated, it remains partially implemented, allowing the service to accept analytics requests but providing hardcoded responses. Additionally, their implementation integrates with Free5GC, another open-source 5G core network implementation. The data for deriving analytics are collected from the NFs within Free5GC and then forwarded to the MTLF for further processing and training of machine learning models. However, the specific nature of the data received by the MTLF from the NFs is currently undisclosed.

Lee et al.'s [6] NWDAF implementation includes both the AnLF and MTLF components, with a specific focus on the Nnwdaf AnalyticsInfo service of the AnLF and the Nnwdaf MLModelProvision service of the MTLF. The authors demonstrated the interaction between these two services when a request for prediction-type analytics is directed to the Nnwdaf AnalyticsInfo service. In this scenario, the AnLF would require machine learning (ML) models for inference. If the required model is unavailable within the AnLF, the NWDAF containing the AnLF sends a request to the NWDAF containing the Nnwdaf MLModelProvision service. Subsequently, the NWDAF with the AnLF receives a URL to the ML model file, which it then downloads to execute inference and generate analytics results for the consumer. In this work, this workflow was not implemented as the Nnwdaf AnalyticsInfo service solely provides statistics and does not offer predictions on the number of UEs in a given area of interest. Nevertheless, the workflow proposed by the authors appears to be the appropriate approach for implementing prediction-type analytics. However, it is important to note that a 5G core network was not simulated in their experiments. Therefore, it is unclear where the data for deriving the analytics originates. Additionally, the type of analytics requested for demonstration is not defined in the standards.

B. NWDAF in combination with Anomaly Detection

Relevant works on combining the NWDAF with anomaly detection scenarios have been published by Mekrache et al. [7] and Sevgican et al. [8]. [7] utilized the NWDAF to detect UE traffic anomalies using an unsupervised machine learning approach, specifically utilizing the Long Short-Term Memory (LSTM) Autoencoder algorithm. They trained the LSTM Auto-encoder with real data from the Milano dataset and incorporated various 3GPP-standardized analytics provided by the NWDAF. These analytics included network performance analytics, specifically session success ratios and the number of UEs in an area of interest, UE communication analytics, and NF load analytics. Additionally, they implemented the "Unexpected large rate flows" exception for abnormal UE behavior analytics. Notably, the anomaly defined in their work pertains to an unexpectedly large amount of network traffic. In contrast, we define it as an unexpectedly low amount, possibly indicating issues such as signal interference or network congestion. Furthermore, a key distinction of their approach and ours lies in their utilization of OAI 5G-CN5 as the 5G core network, which already integrates AMF and SMF Event Exposure services. While they implemented the number of UEs in area of interest analytics, they derived this information through AMF registration events. Arguably, the specific events required to derive NUM_OF_UE analytics are not standardized; the standard merely defines the type of data and its source for deriving the analytics. However, the AMF event exposure service offers the UES_IN_AREA_REPORT event, which, while not explicitly designated for NUM_OF_UE analytics, is logically suitable for this purpose. Therefore, in this work, the implementation of the number of UEs in an area of interest analytics relied on UES_IN_AREA_REPORT events from the AMF event exposure service. Similar to [7], [8] also focused on implementing network traffic anomaly detection using the NWDAF and defined the anomaly as unexpectedly large amounts of network traffic. One notable difference, however, is their use of synthetic data for the experiments. The data generation scenario involved five Remote Radio Unit (RRU) cells, each serving three subscriber categories: platinum, gold, and silver. Within each subscriber category, five different types of personal equipment were considered: IoT devices, vehicles, cell phones, smartwatches and tablet computers. Mean handover ratios per hour for each device type, with varying values based on the time of day to simulate real-world scenarios have been assigned. Additionally, a set of predefined initial loads for each cell was defined, categorized by subscriber category and device type. Subsequently, six months' worth of network traffic data have been generated, comprising snapshots of the network taken at 15-minute intervals. Within each interval, UEs may perform handovers between adjacent cells. Following this, the authors trained two anomaly detection models and compared their performances. They employed logistic regression and XGBoost techniques, with XGBoost demonstrating superior performance in predicting anomalies compared to logistic regression. It is noteworthy that in contrast to the present work none of the standardized APIs for the NWDAF were implemented, and a 5G core network was not deployed for the experiment.

IV. Session-based Anomaly Detection Concept

In a private 5G network environment a limited number of users connect via the base stations to the local network and resources, including machine to machine communication of devices working together on the factory floor. In contrast to public mobile networks with a highly dynamic, diverse and mobile user base, usage patterns of the private 5G deployment

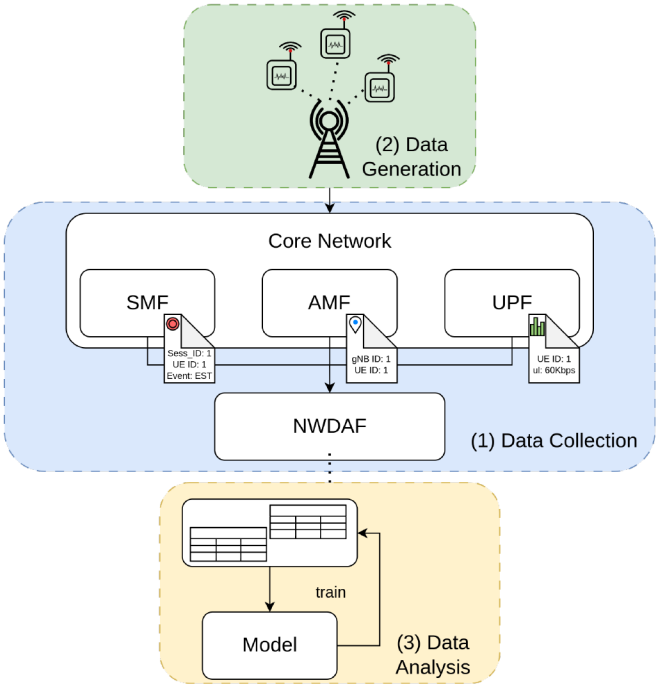

Fig. 2. Overall architecture and main stages of the 5G user session anomaly detection pipeline

occur in close connection with the actual purpose of the network and the applications (repeatedly) executed in the environment. The session-based anomaly detection concept presented in this work exploits this characteristic and designs a supervised learning approach that is closely linking the private 5G network with the executed use case. Consequently, when a communication pattern changes on the factory floor e.g. due to a different manufacturing process being started, the fitting model trained for this communication pattern needs to be loaded from the NWDAF in order to detect the corresponding anomalies.

The general stages of the proposed anomaly detection concept are as follows:

1) Devices participating in the communication on the factory floor trigger sequences of events in the 5G core network functions (NFs), e.g. in SMF upon starting or concluding a data transmission.
2) These metadata information pieces are collected from the NFs via purposefully designed interfaces referred to as event exposure services, providing the particular metadata to NWDAF.
3) The metadata collected in NWDAF's database is then utilized to train AI models that should detect a specific situation deemed undesirable on the factory floor, e.g. a failed communication of a node that leads to disturbances in the process.

To give an example, a robot on a production line may initiate a communication on the task that it is about to execute, e.g. to load the precise locations of drilling holes in a workpiece located in its 3D CAD model. The robot executes its steps and the metadata of this communication is screened by the session-based anomaly detection, which triggers an alarm when the session ends prematurely or when the transmission of data has been smaller than expected.

Figure 2 shows these stages of the session-based anomaly detection concept.

V. Implementation

In this Section we present the implemented session-based anomaly detection concept. Figure 3 provides a high-level view on the implemented components and interfaces within and outside of the NWDAF, building upon the open source Open5GS core network implementation. In general the implemented components can be subsumed by their intended functionality, with data collection functionality in blue and data analysis parts in yellow. In addition to the Open5GS and NWDAF related implementation work, we have created a data generation approach based on UERANSIM.

The implementation of the session-based anomaly detection concept is demonstrated through an industrial factory scenario, where multiple UEs act as sensors, continuously uploading data to a central server. The scenario is described as follows:

- A number of sensors upload 60KB of data every 5 seconds with a default bandwidth of 60 Kbps. The sequence of operations for each sensor includes turning on, establishing a PDU session, uploading the data, releasing the PDU session, and then turning off.
- Additionally, the NWDAF would be tasked to detect the following anomalies:
  - UE throughput anomaly: Sensors send data with an upload bandwidth ranging from 30% to 70% of their original bandwidth.
  - PDU session anomaly: Sensors go through a cycle of establishing, releasing, and reestablishing a PDU session.

In the following subsections we provide a description of the data generation, data collection and data analysis parts that simulates the aforementioned scenario.

A. Data Collection for Session-based Anomaly Detection

The data collection part aims at bringing the relevant information for the anomaly detection into the NWDAF. In order to achieve this we implemented the Namf EventExposure and Nsmf EventExposure services, which were not implemented in the 2.6.4. version of the Open5GS. The implementation of the Nsmf EventExposure service focused on notifying when a PDU session was established or released, represented by the PDU SES EST and PDU SES REL events, respectively. Conversely, the Namf EventExposure service facilitated notifications regarding UEs in specific areas of interest, represented by the UES IN AREA REPORT event. Upon event occurrence, these notifications are sent to the Data Collection Module, comprising an NGINX server and a callback server. Given that all Service-Based Interfaces (SBI) in Open5GS communicate via HTTP/2, the options were either deploying

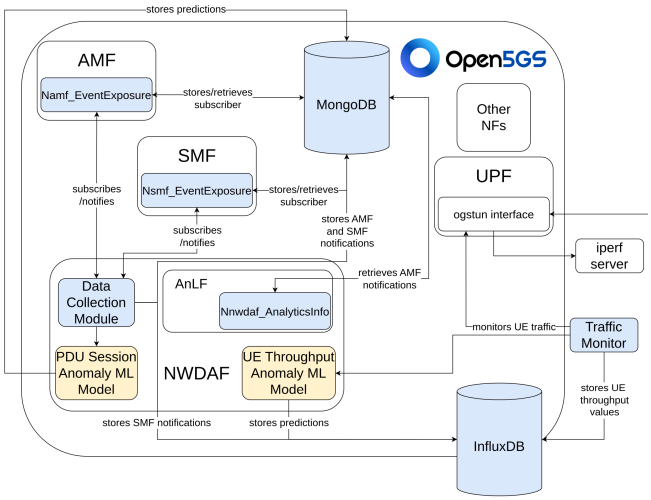

Fig. 3. Overview on the implemented concept for NWDAF-based anomaly detection

a reverse proxy to convert HTTP/2 requests into HTTP/1.1 or implementing a callback server that handles HTTP/2 requests directly. The former option was chosen to allow for different callback server implementations in the future, as not many server frameworks currently provide robust HTTP/2 support. Upon receiving notifications from the Nsmf EventExposure and Namf EventExposure services, the notifications are stored in MongoDB and InfluxDB. The Nnwdaf AnalyticsInfo service queries the MongoDB database for the UES IN AREA REPORT notifications to derive analytics on the number of UEs in an area of interest. On the other hand, the PDU session notifications were stored to train the PDU session anomaly ML model. Finally, the Traffic Monitor monitors the traffic of individual UEs, which are configured to transmit packets to an iperf2 server running on the Open5GS VM. It calculates the throughput and subsequently stores the values in the InfluxDB database to be utilized as training data for the UE throughput anomaly ML model.

B. Session-based Anomaly Detection Engine

In this subsection we describe the two anomaly detection models. Both anomaly detection models were implemented using the scikit-learn library in Python. Specifically, the models utilized Decision Tree to classify anomalies. The choice of the type of machine learning method was primarily driven by the aim to implement a straightforward machine learning approach. Preprocessing steps were conducted using the pandas library to group, filter, and manipulate the data. The implementation process of the anomaly detection models first involves constructing datasets from the collected data described in the previous section. There are three time series datasets in total:

1) PDU Session Dataset: contains information regarding the establishment or release of PDU sessions. It includes details such as the UE's SUPI, IP address, and the designated data network for the PDU session.
2) UE Throughput Dataset: records the uplink throughput values of the UEs during data transfer. The UEs are identified by their IP addresses.
3) Event Report Dataset: contains events from a simulation conducted during the data generation phase, which are used to label anomalies in both the PDU session and UE throughput datasets. The events captured in this dataset include:
   - Start and end of the simulation
   - Start and end of UE data transmission
   - Start and end of the UE throughput anomaly
   - Start and end of the PDU session anomaly

With these datasets, the following models were implemented:

1) PDU Session Anomaly ML Model: This model utilizes the PDU session and the event report datasets to detect instances where a UE is unable to maintain a PDU session, characterized by repeated cycles of PDU session release and re-establishments. The PDU session dataset was first divided into separate pandas dataframes, each concerning a single UE. Two additional features were then added to each dataframe: time_difference, which represents the time difference between the current row and the previous row, and is_pdu_sess_established, which indicates whether the PDU session was established after the event in the current row. Afterwards, labelling the anomalies in the dataset involves referencing the timestamps of the start and end of the anomaly events from the event report dataset. The start and end of PDU session anomaly events in the event report are represented by SINGLE_PDU_SESSION_ANOMALY_START and SINGLE_PDU_SESSION_ANOMALY_END events, respectively. Each PDU session establishment and release event in the PDU session dataset would be labelled as an anomaly if it occurred within the time range of the SINGLE_PDU_SESSION_ANOMALY_START and SINGLE_PDU_SESSION_ANOMALY_END events, as shown in Figure 4. Finally, the individual dataframes were combined into a single dataframe for training and testing. The combined dataframe was split into training and testing sets using an 80/20 ratio. The features used for training the model were time_difference and is_pdu_sess_established. After training, the NWDAF callback server, as described in the data collection subsection, invokes this model upon receiving PDU session notifications to predict whether a notification is part of a PDU session release-reestablishment anomaly cycle.

2) UE Throughput Anomaly ML Model: This model utilizes all three datasets to detect UEs exhibiting impaired throughput during data transfer. The PDU session dataset was used to map the IP addresses in the UE throughput dataset to the SUPIs, since the Traffic Monitor identifies packets by their IP addresses rather than by their SUPIs. Similar to implementing the PDU session anomaly model, the UE throughput dataset was divided into separate dataframes for each UE before labelling the anomalies in the throughput values. Additionally, the timestamps of

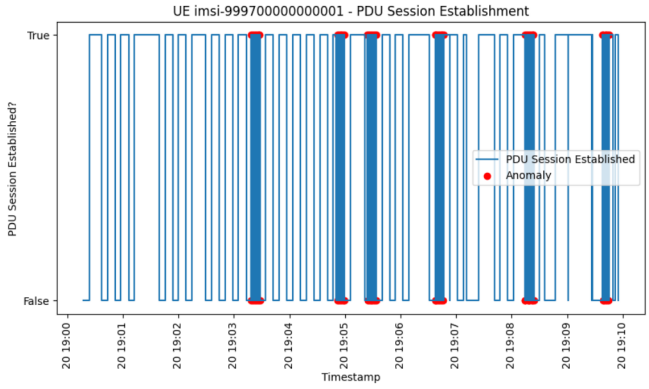

Fig. 4. Example PDU Session Notifications With Labeled Anomalies

the start and end of UE throughput anomaly and data transfer events in the event report dataset were referenced to label anomalies in the UE throughput dataset. The start and end of UE throughput anomaly events are represented by the LOWER_BANDWIDTH_ANOMALY_START and RESET_SENSOR events, respectively. The LOWER_BANDWIDTH_ANOMALY_START event sets the bandwidth of the next data transfer to be 30% to 70% of the default 60Kbps, while RESET_SENSOR resets the bandwidth back to the default for the next data transfer. In contrast, the start and end of the data transfer are represented by the UPLOAD_READING_START and UPLOAD_READING_STOP events, respectively.

```
# tpd holds throughput values of a UE
# lba holds data transfer events and UE throughput anomaly events

is_lower_bandwidth_anomaly = False

for i, row in lba:
    if row["name"] == "LOWER_BANDWIDTH_ANOMALY_START":
        is_lower_bandwidth_anomaly = True
    elif row["name"] == "RESET_SENSOR":
        is_lower_bandwidth_anomaly = False
    elif row["name"] == "UPLOAD_READING_START":
        if is_lower_bandwidth_anomaly:
            # find the next UPLOAD_READING_STOP event
            start = row["time"]
            end = None
            is_reset_sensor_event = False

            for j, subsequent row in lba:
                # if there are any RESET_SENSOR events
                # then the anomaly has ended
                # but throughput values are still anomalous
                # until the next UPLOAD_READING_STOP event
                if row2["name"] == "RESET_SENSOR":
                    is_reset_sensor_event = True
                if row2["name"] == "UPLOAD_READING_STOP":
                    end = row2["time"]
                    break

            if is_reset_sensor_event:
                is_lower_bandwidth_anomaly = False

            if end is None:
                # time of the last event in the simulation
                end = sim_ue_df["time"][sim_ue_df.index[-1]]

            label as anomaly in tpd \
            if throughput["time"] >= start and throughput["time"] <= end
        else:
            continue
```

Listing 1. Pseudocode of labeling throughput anomalies

Listing 1 details the algorithm for labelling the UE throughput anomalies in each UE throughput dataframe.

Unlike the anomaly labelling process of the PDU session dataset, not every value in the UE throughput dataset is anomalous within the time range of the start and end of the anomaly events. This is because the simulation was designed so that UE throughput anomaly events do not interrupt ongoing data transfer. Changes to the bandwidth value during an active data transfer will only affect subsequent data transfers. For example, the following sequence of events will result in the UE uploading data at normal bandwidth:

- `UPLOAD_READING_START, LOWER_BANDWIDTH_ANOMALY_START, UPLOAD_READING_STOP, LOWER_BANDWIDTH_ANOMALY_END`
- `UPLOAD_READING_START, LOWER_BANDWIDTH_ANOMALY_START, LOWER_BANDWIDTH_ANOMALY_END, UPLOAD_READING_STOP`

Conversely, the following order of events will result in the UE uploading data at impaired bandwidth:

- `LOWER_BANDWIDTH_ANOMALY_START, UPLOAD_READING_START, LOWER_BANDWIDTH_ANOMALY_END, UPLOAD_READING_STOP`
- `LOWER_BANDWIDTH_ANOMALY_START, UPLOAD_READING_START, UPLOAD_READING_STOP, LOWER_BANDWIDTH_ANOMALY_END`

Therefore, prior to the start of a data transfer, the LOWER_BANDWIDTH_ANOMALY_START event should have occurred for a throughput value to be anomalous. In Listing 1, this is depicted by the occurrence of a UPLOAD_READING_START event, while the is_lower_bandwidth_anomaly variable is set to true. If this condition occurs, the algorithm iterates through the subsequent rows of the event report dataset until an UPLOAD_READING_STOP event is found. During this process, all throughput values within the timestamp range of the UPLOAD_READING_START and UPLOAD_READING_STOP events are considered anomalous. Similar to the PDU session anomaly model, the individual dataframes were combined into a single dataframe for training and testing, with the combined dataframe split into training and testing sets using an 80/20 ratio. Only the throughput of data transmission from the UE to the iperf server was used as the model feature because the simulation involves only one type of UE, i.e. sensors. If multiple types of UEs with distinct traffic patterns were introduced, incorporating the type of UE as a feature would be necessary to accurately capture the differences in traffic patterns. Finally, the UE throughput anomaly model is then tasked to predict whether a calculated UE throughput value from Traffic Monitor is anomalous.

C. Data Generation

In order to generate characteristic load and obtain the required events from the amended Open5GS core, we have implemented an environment based on the UERANSIM simulator.

The overall data generation concept is visualized in Figure 5, with 2 VMs running on the same host PC. Originally

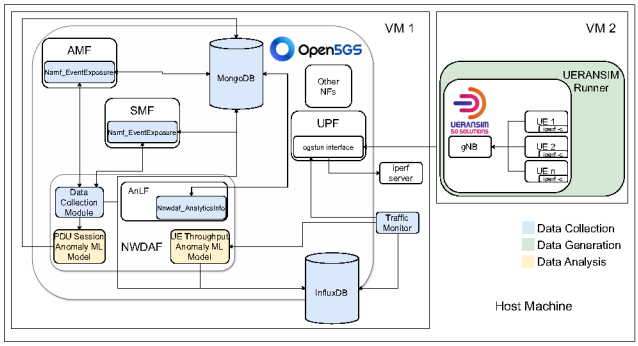

Fig. 5. Data generation with UERANSIM Runner

devised for testing 5G Core Network and studying the 5G system, we have implemented our own UERANSIM Runner component, which serves us as the simulated device layer and gNB to mimic the industrial private 5G deployment. Accordingly, UERANSIM Runner is implemented as a Python script that simulates a 5G industrial factory scenario with UEs acting as sensors and a single gNB, allowing for the simulation of an arbitrary number of UERANSIM UEs.

A large portion of the script relies on the third-party schedule[3] library to execute shell commands at predefined intervals. Each UE has two schedulers to orchestrate the execution of these commands:

1) **UE Scheduler:** responsible for handling UE operations such as turning on and off, establishing/releasing PDU sessions, and simulating data upload.
2) **Anomaly Scheduler:** coordinates the execution of UE throughput and PDU session anomalies.

Figure 6 visualizes the approach for generating the PDU session anomaly with the help of the anomaly scheduler.

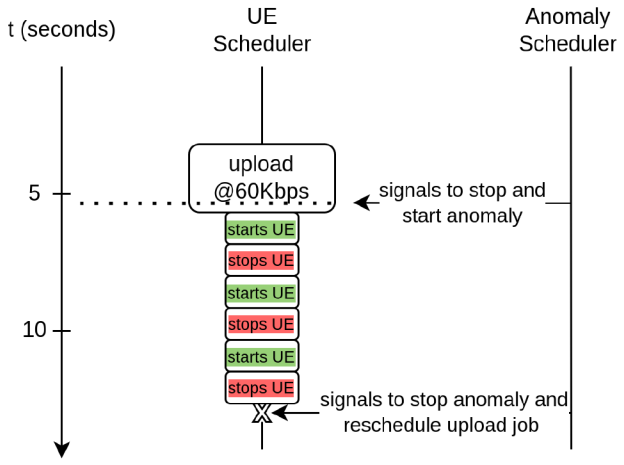

Fig. 6. PDU Session Anomaly

[3]https://github.com/dbader/schedule

VI. Experimental Validation Scenario and Evaluation

This section evaluates the performance of the PDU session and UE throughput anomaly ML models. The model predictions were generated after training the model with generated data from UERANSIM Runner. The scenario is described as follows:

- Two UEs were deployed. Each UE was scheduled to send 60KB of data at a default bandwidth of 60Kbps every 5 seconds.
- UE throughput anomalies were scheduled 30 times for each UE, with each anomaly lasting for the default duration of 20 seconds.
- PDU session anomalies were scheduled 12 times for each UE, and each anomaly lasted 10 seconds by default.
- The anomalies were scheduled to occur at random timestamps.
- The simulation ran for a total duration of 30 minutes.

The PDU session and UE throughput anomaly ML models have been handled for the preprocessing steps, labeling of the anomalies, and the training phase as described in Section V-B. The same scenario was then used to evaluate the models' predictions, which we describe in the following. The results follow a similar trend for both UEs, therefore only one is provided for the sake of brevity. Figure 7 shows the cycles of PDU session establish and release for UE1 with the output of the PDU session anomaly model. The thick lines indicate the change of the session status in quick succession, the red markers indicate that the PDU session anomaly model predicted an anomaly (e.g. a preempted session). Figure 8 on the other hand shows the uplink throughput of UE1 including the output of the UE throughput anomaly detection model. As can be seen the model successfully detects when the user traffic pattern for uplink traffic shows anomalous behavior, indicated again by the red dots.

During the 30 minute simulation, both models achieved high scores across all metrics. Notably, both models achieved precision scores of 0.96 for "False" and 0.97 for "True", indicating minimal false positives. The throughput anomaly model nearly achieved perfect recall for the "False" class while the PDU session anomaly model lagged slightly behind by 1%. However, both models exhibited identical recall scores of 0.95 for predicting the "True" class. Overall, the high accuracy scores of 0.96 for both models affirm in accurately classifying anomalies within the dataset. While the models were able to correctly detect PDU session or throughput anomalies, it should be noted that they have been designed and trained for a specific traffic and usage pattern, therefore they will not perform well for other scenarios were different usage patterns occur. Hence, the specific industrial application traffic and usage pattern is required for training the model to detect anomalies in a given scenario, tightly coupling then ML model and anomaly detection with the use case at hand.

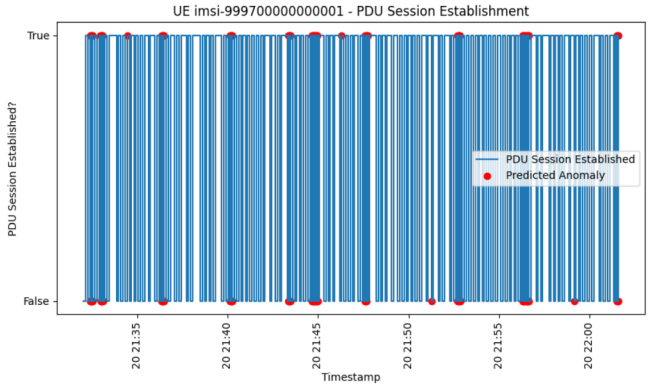

Fig. 7. PDU Session Status with Predicted Anomalies of UE1

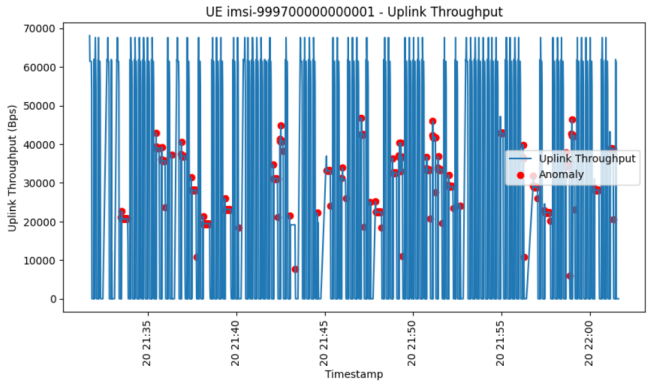

Fig. 8. Uplink Throughput with Predicted Anomalies of UE1

VII. Conclusion and Future Work

In this paper we have presented the concept of session- and throughput-based detection of UE anomalies, based on data gathered from a real 5G core network, fed with simulated user interactions utilizing UERANSIM. The approach may serve as a blueprint for the generation of tailored anomaly detection functions that exploit known traffic patterns in industrial factory scenarios. Regarding the existing body of work in the NWDAF specification, we contribute with our approach with an additional category of abnormal user behavior that amends the existing standardization of the NWDAF. We have implemented the required functionalities in the relevant core network functions of Open5GS in order to expose the metadata of PDU sessions from SMF, UE throughput from the UPF, as well as the UE location from the AMF. We presented two ML models trained on the obtained metadata, which showed a high accuracy in detecting anomalies when fed with our implemented anomaly generator component.

Having validated the concept by means of simulated users, we are currently working on applying the solution on our private 5G testbed deployed at the TU Berlin campus. Furthermore, we are working on integrating the session-based anomaly detection in other work packages of the Cobra-5G project, exploiting the core perspective on user sessions in other contexts.

Acknowledgment

This work was conducted within the Cobra-5G project, funded by the German Federal Office for Information Security (BSI) in the “Cyber Security and Digital Sovereignty in 5G/6G Communication Technologies” program.

References

[1] 3GPP, “Architecture enhancements for 5g system (5gs) to support network data analytics services,” Technical Specification (TS 23.288), 3rd Generation Partnership Project (3GPP), June 2023, version 17.9.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579

[2] S. Eswaran and P. Honnavalli, “Private 5g networks: a survey on enabling technologies, deployment models, use cases and research directions,” *Telecommunication Systems*, vol. 82, no. 1, pp. 3–26, Jan 2023.

[3] J. Ordonez-Lucena, J. F. Chavarria, L. M. Contreras, and A. Pastor, “The use of 5g non-public networks to support industry 4.0 scenarios,” in *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2019, pp. 1–7.

[4] D. M. Manias, A. Chouman, and A. Shami, “An nwdaf approach to 5g core network signaling traffic: Analysis and characterization,” in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*. IEEE, Dec. 2022.

[5] T. Kim, J. Kim, H. Ko, S. Seo, Y. Jeon, H. Jeong, S. Lee, and S. Pack, “An implementation study of network data analytic function in 5g,” in *2022 IEEE International Conference on Consumer Electronics (ICCE)*, 2022, pp. 1–3.

[6] S. Lee, J. Lee, T. Kim, D. Jung, I. Cha, D. Cha, H. Ko, and S. Pack, “Design and implementation of network data analytics function in 5g,” in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, 2022, pp. 757–759.

[7] A. Mekrache, K. Boutiba, and A. Ksentini, “Combining network data analytics function and machine learning for abnormal traffic detection in beyond 5g,” in *GLOBECOM 2023, IEEE Global Communications Conference, 4-8 December 2023, Kuala Lumpur, Malaysia*, IEEE, Ed., Kuala Lumpur, 2023.

[8] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz, and T. Tugcu, “Intelligent network data analytics function in 5g cellular networks using machine learning,” *Journal of Communications and Networks*, vol. 22, no. 3, pp. 269–280, 2020.