Edge Machine Learning in 3GPP NB-IoT: Architecture, Applications and Demonstration

Dejan Vukobratovic, Milan Lukic, Ivan Mezei, Dragana Bajovic
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia
{dejanv, milan_lukic, imezei, dbajovic}@uns.ac.rs

Dragan Danilovic
A1 Serbia
Belgrade, Serbia
d.danilovic@a1.rs

Milos Savic, Zarko Bodroski, Srdjan Skrbic, Dusan Jakovetic
Faculty of Sciences
University of Novi Sad
Novi Sad, Serbia
{svc, zarko.bodroski, srdjan.skrbic, dusan.jakovetic}@dmi.uns.ac.rs

***Abstract*—The emergence of cellular Internet of Things (IoT) standards such as NB-IoT brings novel opportunities for low-cost wide-area IoT applications. Augmenting massive IoT deployments with Machine Learning (ML) algorithms deployed at the edge enables design and implementation of a novel intelligent IoT services. In this paper, we present an architectural outlook and an overview of our recent activities that target integration of ML modules into the cellular IoT architecture. The three-layer architecture considered in this paper embeds ML modules at the edge devices (ML-EDGE), within the core network (ML-FOG) and at the cloud servers (ML-CLOUD), thus balancing between the system response time and accuracy. We discuss alignment of the proposed architecture with ongoing trends in 3GPP architecture evolution. We design, integrate and demonstrate edge ML use cases relying on our real-world deployment of about 150 static and mobile nodes integrated into the NB-IoT network.**

***Index Terms*—NB-IoT, Machine Learning, Edge Computing**

I. Introduction

Billions of cellular Internet of Things (IoT) devices are being connected world-wide, providing wide area wireless sensing infrastructure. 3GPP NarrowBand-IoT (NB-IoT) represents the most popular solution currently deployed by mobile operators around the world [1]. NB-IoT provides a radio interface able to connect tens of thousands of low-cost NB-IoT devices per macro-cell. As the trends of integration of Machine Learning (ML) with IoT are gaining momentum [2], [3], the system-level solution for ML integration in 3GPP-based cellular IoT networks becomes highly relevant.

Deploying ML services within 3GPP network is an emerging topic in 3GPP standardization [4]. In 3GPP Release 16/17 of standards, a novel 5G core network (CN) Network Data Analytics Function (NWDAF) is introduced [5] to support intelligent network automation. In Rel. 16, only slice automation and slice load analytics is considered, while Rel. 17 extends the work towards user equipment (UE)-related analytics. Currently, 3GPP work is restricted to services offered to mobile operators aiming at provision of data analytics and automated network operation and management.

The work is supported in part by European Commission's Horizon 2020 Research and Innovation Programme, Grants No. 856967, 833828 and 871518.

In this work, we present our work that integrates ML services into a 3GPP NB-IoT network architecture, following an established three-level edge-fog-cloud approach [6]. The critical aspects are: i) The design of edge ML modules for resource-constrained NB-IoT devices [7], and ii) The design of an orchestration mechanism that trades-off resource limitations and system response [8]. The proposed ML-augmented NB-IoT service supports applications such as anomaly detection, identification of malfunctioning devices, security threat detection, and others. The complexity of ML modules is matched to different deployment levels, thus balancing between the NB-IoT service response and accuracy.

This work extends and generalises our recent work on NB-IoT-based anomaly detection [9]. We first present two custom-designed NB-IoT device platforms for indoor and outdoor use cases suitable for collection of both application data (e.g., raw data from on-board sensors) and UE-specific data (e.g., on device energy consumption and radio channel conditions). We further investigate different edge ML architectures suitable for ML-based NB-IoT applications. Thus our work builds upon recent 3GPP efforts, while focuseing on NB-IoT in-network data analytics, supported by real-world deployment and demonstration study.

The paper is organized as follows. In Sec. II, we present technical background and review the related work on edge ML for IoT. The ML-augmented 3GPP NB-IoT system architecture is presented in Sec. III. Sec. IV presents in detail ML-EDGE module and presents selected applications of edge ML in 3GPP NB-IoT. In Sec. V, we describe system integration and deployment, and provide example results from real-world experiments. The paper is concluded in Sec. VI.

II. Background

A. 3GPP NB-IoT System Architecture

NB-IoT is a recently introduced Cellular IoT technology that can be seamlessly integrated into an existing 3GPP architecture [1]. NB-IoT may coexists in the radio access network with both 3GPP 4G LTE and 5G NR, while using the same evolved packet core (EPC). NB-IoT user equipment (UE) connects to the network via eNodeB (eNB) within Evolved

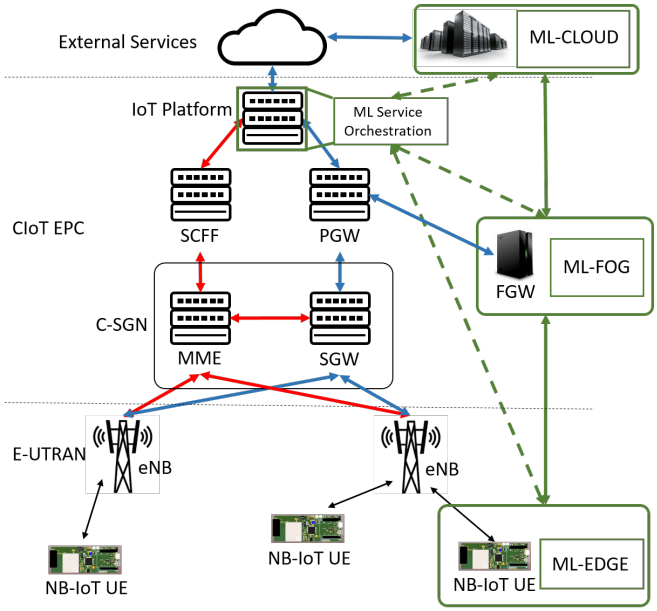


Fig. 1: ML-augmented 3GPP NB-IoT architecture.

Universal Terrestrial Radio Access Network (E-UTRAN). eNB provides user-plane and control-plane information transfer to the main EPC element, CIoT Serving Gateway Node (C-SGN), which embodies both Mobility Management Entity (MME) and Serving Gateway (SGW) functions. User-plane data are forwarded via Packet Gateway (PGW) to the mobile/operator IoT platform, or to the external network servers (Fig. 1).

Extension of 3GPP architecture to support data analytics services is under way as part of Rel. 16 and 17 through the introduction of NWDAF and management data analytics function (MDAF) in 5GCN service-based architecture (SBA). NWDAF/MDAF are able to support data analytics services or enhance other 5GCN network functions with various statistics and predictions [10]. The primary goal of NWDAF/MDAF introduction is to enable future 5G network automation and enhance 5G network operation, administration and management (OAM). Further steps in 5G data analytics architecture evolution is to extend its reach from 5GCN towards 5G radio access network (RAN) [10]. These steps are already taken as part of the Open RAN (O-RAN) initiative through the introduction of near-real-time (Near-RT) and non-real-time (Non-RT) Radio Interface Controllers [11].

B. Edge Machine Learning for Cellular IoT

ML naturally complements IoT networks as the latter represent a rich source of data [2], [3]. Recent trends see shifting ML deployment from the cloud towards the edge. The benefits of ML at the edge are: 1) reduced latency and improved responsiveness, 2) reduced traffic load towards the network core, and 3) enhanced privacy as data remains at the edge.

The main challenges of implementing ML models at the edge are scalability and resource scarcity [7]. Depending on the ML algorithm being run on the edge node, the size of the ML model can go as low as a few kilobytes. Since the capability of ML models deployed at, e.g., NB-IoT devices, is severely restricted, appropriate task offloading mechanisms are a key to efficient system design and implementation [8].

Although there are numerous applications for ML at the edge, current research focuses on ML for IoT security [12]. The anomaly detection algorithms at the edge represent an underlying approach for many threats detection solutions [13]. However, integrating edge ML into 3GPP Cellular IoT network is addressed only recently [14].

III. Edge ML in 3GPP NB-IoT: System Architecture

A. System Architecture

The three-level ML-augmented architecture considered in this work is illustrated in Fig. 1. It consists of: 1) the edge ML module embedded at the NB-IoT UE (ML-EDGE), 2) the fog ML module placed within dedicated EPC nodes (ML-FOG), and 3) the cloud ML module residing in mobile operator or external cloud servers (ML-CLOUD).

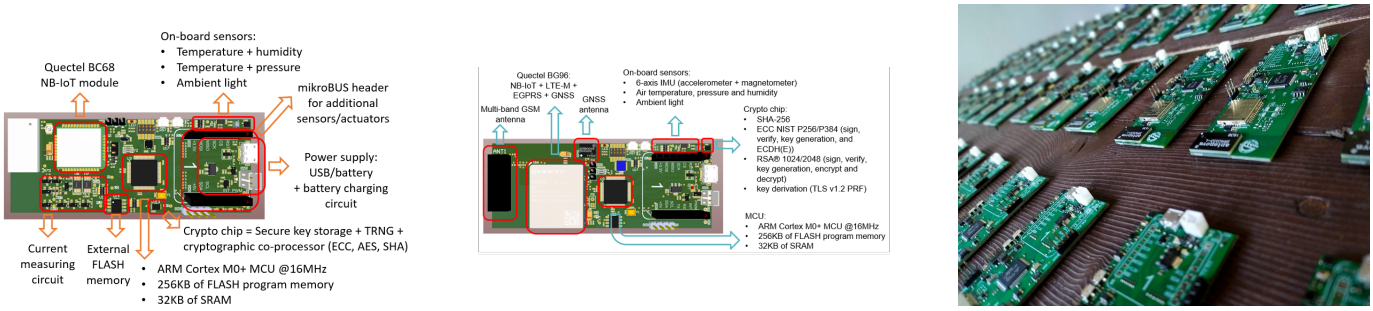
The ML-EDGE module hosts low-complexity ML algorithm implemented on top of the resource-constrained microcontroller. It operates on data locally acquired by an NB-IoT UE, trading-off minimum-delay system response with the ML algorithm complexity and accuracy. ML-FOG module is deployed at the fog gateways (FGW) within mobile operator EPC. FGW may support medium-to-high complexity ML algorithms fed by the data provided by a large pool of NB-IoT edge nodes. ML-FOG response time is affected by radio access and core network latency, which in the case of NB-IoT may extend in the range of seconds [15]. ML-CLOUD hosts high-complexity ML algorithms at cloud servers, and provides the highest flexibility for training and implementing ML algorithms, albeit with highest delays. Although not in the focus of this paper, the critical part of the three-layer architecture is an orchestration mechanism, whose task is to govern the level at which the decision is made. Since the ML-EDGE is the most challenging architecture element, we next provide details on NB-IoT edge node design providing support for ML-EDGE.

B. NB-IoT Edge Device Design

For the purpose of real-world demonstration and deployment, we designed and fabricated two different NB-IoT edge node platforms, one for a static indoor use case (e.g., Smart Buildings) illustrated in Fig. 2a, and another for a mobile outdoor use case (e.g., Smart Logistics) illustrated in Fig 2b. 100 static and 50 mobile NB-IoT edge nodes are produced to support emulation of a massive real-world NB-IoT setup (Fig. 2c). Both nodes feature standard on board components:

3GPP NB-IoT Module: We utilize BC68 (indoor node) and BG96 (outdoor node) modules from Quectel. The former supports 3GPP NB-IoT, while the latter supports both 3GPP NB-IoT and LTE-M connectivity. The latter also integrates GNSS module to provide the geolocation information which is essential for outdoor use cases.

On-board sensors: Both nodes are equipped with the set of sensors used to measure the atmospheric conditions



(a) (b) (c)

Fig. 2: NB-IoT edge node design. (a) Indoor use case; (b) Outdoor use case; (c) NB-IoT nodes for massive testbed.

such as air temperature, pressure and humidity. The indoor sensor includes additional illumination measurement sensor. The outdoor sensor, apart from the localization data provided by the GNSS module, uses the 6-axis Inertial Measurement Unit (IMU) to collect data about the vibrations.

Micro-controller unit (MCU): The MCU used is a low-power 32-bit ARM Cortex M0+ with 256KB of FLASH and 32KB of SRAM, operating at 16MHz. The absence of operating system as well as the hardware constraints limit the usage of ML tools only to lightweight models that are fully customized and optimized for a given application. Finally, an external FLASH memory module enables data logging over the intervals when there is no connectivity.

C. Data Inputs to ML-EDGE Module

NB-IoT edge nodes are capable of collecting two types of data: i) application data (sensor data), and ii) UE-specific data. Both data streams may feed edge ML, as we describe in Sec. IV-B. Application data includes data acquired from on-board sensors and is the source of data for end-user edge ML applications. The UE-specific data is suitable input for NWDAF/MDAF and are collect from three sources:

Radio channel conditions: Radio measurements are available from NB-IoT module through standard AT commands. They provide a snapshot of statistics for numerous parameters such as Signal-to-Noise-Ration (SNR), Received Signal Strength Indicator (RSSI), total Tx/Rx time, Block Error Rates (BLER), etc., at a given instant the command is executed.

NB-IoT module current consumption: This feature is available only in indoor NB-IoT platform. The dedicated current-measurement circuit measures the current consumed by BC-68 NB-IoT module, thus eliminating the influence of other on-board components. Current sampling period is 1 ms, i.e., it is aligned with a single subframe duration in NB-IoT.

NB-IoT and eNB message exchange logs: NB-IoT UE modules allow for extraction of message exchange logs between NB-IoT UE and eNB using appropriate applications. Such data logs provide opportunity for extracting a number of useful information that provide details on NB-IoT UE behavior, including reconstruction of detailed scheduling information and applied PHY layer configurations both in the uplink and the downlink.

IV. Edge ML Module and Applications

A. ML-EDGE module in NB-IoT

The ML-EDGE module in NB-IoT is designed taking into account the device constraints. Due to a low computation power of MCU and memory capacity, it is practically infeasible to train ML models directly on the NB-IoT device: 1) a large number of data points have to be stored on the device to train a predictive model exhibiting an acceptable level of accuracy, 2) the model training is a computationally intensive optimization process usually performed in a large number of iterative steps, 3) a low computation power prevents any serious model validation and tuning of model hyper-parameters. Consequently, we adopt a scheme in which a lightweight inference engine is directly integrated into the firmware of the NB-IoT device. The inference engine makes predictions according to a neural network that was previously trained, tuned and validated on an appropriately built data set.

The ML-EDGE inference engine performs the feed-forward operation on a neural network stored on the device for data points prior to their transmission to FGW. It is implemented in C programming language as a standalone, self-contained module without any external dependencies to third party libraries. The training, validation and tuning of ML-EDGE neural network models is performed offline in TensorFlow. Model parameters are determined using the Adam optimizer. Different loss functions are used for different types of ML-EDGE neural network models: mean squared error for regression neural networks and autoencoders, categorical cross entropy for n-ary classification models and binary cross entropy for binary classification models. Before model training, data points in the training dataset are normalized such that each feature has zero mean and unit variance. The structure of the trained models, its weights and data normalization parameters are then exported as C declarations to a header file that is included by the C module containing the implementation of the generic neural network ML-EDGE inference engine.

B. Application Examples for Edge ML in NB-IoT

In this section, we present a sample of our recent applications of edge ML concepts deployed and demonstrated in a real-world 3GPP NB-IoT network, carried out as part of a sequence of EU funded research projects.

NB-IoT module energy consumption measurements, modelling and prediction [16]: Using the precise circuitry for instantaneous current measurements, our indoor NB-IoT node provides energy consumption measurements for every packet transmitted/received by the NB-IoT UE. Using message exchange logs, this consumption is further disaggregated into phases of NB-IoT packet transmission (e.g., cell search and synchronization, system information acquisition, random access, UL/DL transmission) [17]. In our setup of 100 indoor NB-IoT devices, 15 devices are equipped with current measurement circuitry and are used as data loggers, associating appropriately designed feature vectors with labels defined in the form of packet-level energy consumption. Based on the data sets and labels accumulated at data loggers at different indoor reception conditions, deep learning based energy consumption models can be designed, pre-trained and deployed at the remaining NB-IoT edge devices (85 devices without current measurement support) to provide for prediction of energy consumed per packet at the NB-IoT UE. Such models can be also deployed at ML-FOG (or ML-CLOUD) nodes, providing a mobile operator with an efficient network-wide support for predicting, monitoring and maintenance of battery states and battery replacement requirements of massive sets of connected NB-IoT devices [18].

Deep autoencoder anomaly detection in Smart Logistics [19]: The most popular edge ML applications are in the domain of anomaly detection [13]. In a recent work [9], we have designed, implemented and deployed deep autoencoder (AE) based anomaly detection for applications in Smart Logistics. The outdoor NB-IoT node is deployed attached to shipping containers in a factory supply chain, in order to collect data, deploy and test the ML-EDGE module. Based on the collected data sets, deep AE is trained to distinguish between the normal state (usual container vibrations during transport) and the anomalous state (extensive vibrations or container overturns). Two-level architecture is implemented that orchestrates between decision being taken at the ML-EDGE or ML-FOG module deployed within FGW as part of the mobile operator EPC network.

Device identification for secure Industrial IoT using wireless fingerprinting [20]: Security threats in industrial IoT networks call for innovative applications of ML for IoT security. Securing the expanded attack surface of a large number of IoT devices will require using all available data at edge nodes. The indoor NB-IoT node provides abundance of metadata for these purposes. In particular, collection and usage of so called "wireless fingerprints," in the form of channel state information, could be used for additional device identification. ML-EDGE module may host anomaly detection algorithm that would distinguish between normal and anomalous wireless fingerprints received from NB-IoT UE, thus indicating possible physical tampering with the device. In addition, for a collection of NB-IoT nodes deployed at a certain location, ML-FOG module may host a ML classification method to distinguish between deployed devices not only by their MAC/IP address, but also by their wireless fingerprints.

V. Demonstration of Edge ML for 3GPP NB-IoT

A. System Integration and Demonstration

To integrate the system, collect real-world data and perform testing and evaluation, NB-IoT UEs are connected to the FGW set inside an EPC of a mobile operator network. NB-IoT UEs running ML-EDGE modules periodically send data points via macro-cellular NB-IoT eNB to the FGW using UDP. Within the virtualized environment of FGW server, FOG-ML software module receives and stores UDP packets sent by NB-IoT UEs. The FGW server provides sufficient resources to run higher complexity ML-FOG module, in contrast to the lightweight ML-EDGE module on the NB-IoT UE.

Based on the feature vector extracted from sensory or UE-specific data, ML-EDGE is periodically fed with an input feature vector. Depending on the design choices, the same or different feature vector is also sent to FGW for inference at the more powerful ML-FOG engine. For each input vector, ML-EDGE produces a decision and a confidence score, which is forwarded to the ML orchestration engine for further offloading decisions. Note that the communication delay incurred by the NB-IoT network connection may vary between the order of tens-of-milliseconds to tens-of-seconds, depending on the NB-IoT device radio conditions and the network load [15].

The ML-augmented NB-IoT architecture features several important properties: 1) ML-EDGE at the NB-IoT node provides immediate decisions after each data point providing extremely fast response time; 2) ML-FOG may apply more powerful design using a longer feature vector and more complex architecture, however, the NB-IoT uplink can be a bottleneck and cause unpredictable delays; 3) ML-EDGE module has access to raw data, while ML-FOG gets access to data aggregated at the NB-IoT UE in order to reduce communication load; (4) The final decision at the system level is achieved in coordination of ML orchestration engine, whose offloading decisions are out of the scope of this paper.

B. Performance Results

In this section, we present an experimental results demonstrating trade-off between system accuracy and response time. The setup consists of two anomaly detection models: ML-EDGE AE integrated into the firmware of the NB-IoT device, and ML-FOG AE deployed on the FGW. The difference between the two models is that ML-EDGE AE processes individual data points through a single hidden layer, while ML-FOG AE detects anomalies in timeseries data (k consecutive data points) through three hidden layers.

The results presented here extend our evaluation in [9]. Namely, ML-EDGE and ML-FOG AEs are expanded by incorporating two widely used regularization mechanisms to reduce overfitting and decrease errors of ML models based on neural networks: dropout layers after each hidden layer and L_2 regularization for output of hidden nodes. The considered regularization mechanisms are incorporated both individually (denoted by "Dropout" and "L2") and jointly ("Dropout-L2"). Expanded ML-EDGE and ML-FOG AEs are trained and

tested using the same datasets and evaluation methodology as in [9]: the training is performed on the dataset reflecting normal device behaviour, while the test dataset contains labeled anomalous events, so the AEs are examined and compared by measuring precision, recall and F_1.

The evaluation results of different variants of ML-EDGE AEs are summarized in Table I. ML-EDGE AEs with regularization mechanisms exhibit the same level of recall as the base ML-EDGE AE without regularization (ML-EDGE-Base). However, ML-EDGE AEs with regularization have considerably higher precision (F_1 scores) than ML-EDGE-Base. We conclude that regularization mechanisms can improve the accuracy of anomaly detection at NB-IoT devices nodes by lowering the number of false positive anomaly alarms.

TABLE I: Evaluation of ML-EDGE autoencoders.

	Precision	Recall	F_1
ML-EDGE-Base	0.7047	0.6897	0.6971
ML-EDGE-Dropout	0.7349	0.6769	0.7047
ML-EDGE-L2	0.7926	0.6821	0.7332
ML-EDGE-Dropout-L2	0.7661	0.6821	0.7216

The F_1 scores of different variants of ML-FOG AEs with regularization are shown in Figure 3 for timeseries lengths from 1 to 10. Expanded ML-FOG AEs processing timeseries longer than 3 points always achieve higher F_1 score than the best ML-EDGE AE (ML-EDGE-L2). Thus ML-FOG anomaly detection models are able to improve overall system accuracy by appropriate offloading decisions. The obtained results quantify trade-offs between performance of anomaly detection and response time for edge-based or fog-based anomaly detection. The response time of ML-EDGE AEs corresponds to one sampling period, while the response time of ML-FOG AEs depends on the length of the time series processed. The ML-FOG AE with both regularization mechanisms (ML-FOG-Dropout-L2) processing timeseries of length 6 achieves the highest F_1 score (F_1 = 0.8042). This AE improves the F_1 score of ML-EDGE-L2 by 9.68% at the cost of increased response time for $T = 6$ sampling periods.

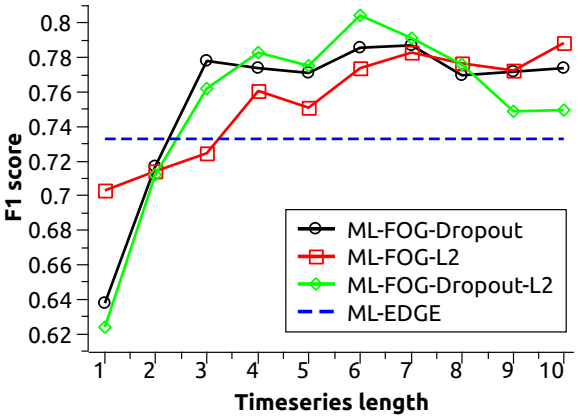


Fig. 3: F_1 scores of ML-FOG AEs. The dashed line indicates the F_1 score of the best ML-EDGE AE.

VI. Conclusion

In this paper, we presented the design, implementation, real-world deployment and evaluation of an ML-augmented architecture for 3GPP NB-IoT networks. In the future work, we will focus on bringing our implementation closer to demonstrating 3GPP-based NWDAF services in NB-IoT network. In addition, we plan to extend our work to designing optimal ML orchestration strategies that would maximize the overall efficiency of edge ML solutions in various NB-IoT use cases.

References

[1] O. Liberg, M. Sundberg, E. Wang, J. Bergman, J. Sachs, "Cellular Internet of things: technologies, standards, and performance," Academic Press, 2017.

[2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing,", Proc. of the IEEE, Vol. 107, No. 8, pp. 1738 – 1762, 2019.

[3] X. Ma, T. Yao, M. Hu, Y. Dong, W. Liu, F. Wang, J. Liu, "A survey on deep learning empowered IoT applications," *IEEE Access*, Vol. 7, pp.181721-181732, 2019.

[4] 3GPP Technical Report TR23.791: "Study of enablers for Network Automation for 5G."

[5] 3GPP Technical Specification TS23.503: "Policy and charging control framework for the 5G System (5GS); Stage 2."

[6] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, "Fog computing and its role in the internet of things," *Proc. Workshop on Mobile cloud computing,* (pp. 13-16), 2012.

[7] A. Kumar, S. Goyal and M. Varma, "Resource-efficient machine learning in 2 kb ram for the internet of things," In *International Conference on Machine Learning* (pp. 1935-1944), July 2017.

[8] W. Sun, J. Liu, Y. Yue, "AI-enhanced offloading in edge computing: When machine learning meets industrial IoT," *IEEE Network*, Vol. 33, No. 5, pp.68-74, 2019.

[9] M. Savic, M. Lukic, D. Danilovic, Z. Bodroski, D. Bajovic, I. Mezei, D. Vukobratovic, S. Skrbic, D. Jakovetic, "Deep Learning Anomaly Detection for Cellular IoT with Applications in Smart Logistics," *IEEE Access*, vol. 9, pp. 59406-59419, 2021.

[10] E. Pateromichelakis, et al. "End-to-end data analytics framework for 5G architecture," IEEE Access, 7, pp. 40295–40312, 2019.

[11] M. Polese, L. Bonati, S. D'Oro, S. Basagni, T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," arXiv preprint arXiv:2202.01032, 2022.

[12] F. Hussain, R. Hussain, S. A. Hassan and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," in IEEE Comm. Surveys & Tutorials, Vol. 22, No. 3, pp. 1686–1721, 2020.

[13] R. Chalapathy , S. Chawla, "Deep Learning for Anomaly Detection: A Survey", 2019, https://arxiv.org/abs/1901.03407v2.

[14] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi, "Machine learning at the edge: A data-driven architecture with applications to 5G cellular networks," *IEEE Trans. Mobile Computing*, 2021.

[15] B. Martinez, F. Adelantado, A. Bartoli and X. Vilajosana, "Exploring the Performance Boundaries of NB-IoT," in IEEE Internet of Things Journal, 6(3), pp. 5702-5712, 2019.

[16] H2020 INCOMING, "Innovation and excellence in massive scale communications and information processing," https://cordis.europa.eu/project/id/856967

[17] M. Lukic, S. Sobot, I. Mezei, D. Danilovic, D. Vukobratovic: "In-Depth Real-World Evaluation of NB-IoT Module Energy Consumption," IEEE SmartIoT 2020, Beijing, China, August 2020.

[18] M. Lukic, S. Sobot, D. Danilovic, I. Mezei, D. Vukobratovic, "Machine learning based power consumption estimation of NB-IoT edge nodes", in preparation.

[19] H2020 C4IIoT, "Cyber-security for Industrial Internet of Things," https://cordis.europa.eu/project/id/833828

[20] H2020 COLLABS, "Comprehensive Cyber-Intelligence framework for resilient Collaborative Manufacturing systems," https://cordis.europa.eu/project/id/871518