Deep Learning Anomaly Detection for Cellular IoT with Applications in Smart Logistics

Milos Savic, Milan Lukic, Dragana Bajovic, Zarko Bodroski, Dragana Bajovic Member, Ivan Mezei Senior Member, Dejan Vukobratovic Senior Member, Srdjan Skrbic and Dusan Jakovetic Member

Abstract—The number of connected Internet of Things (IoT) devices grows at an increasing rate, revealing shortcomings of current IoT networks for cyber-physical infrastructure systems to cope with ensuing device management and security issues. Data-based methods rooted in deep learning (DL) are recently considered to cope with such problems, albeit challenged by deployment of deep learning models at resource-constrained IoT devices. Motivated by the upcoming surge of 5G IoT connectivity in industrial environments, in this paper, we propose to integrate a DL-based anomaly detection (AD) as a service into the 3GPP mobile cellular IoT architecture. The proposed architecture embeds deep autoencoder based anomaly detection modules both at the IoT devices (ADM-EDGE) and in the mobile core network (ADM-FOG), thereby balancing between the system responsiveness and accuracy. We design, integrate, demonstrate and evaluate a testbed that implements the above service in a real-world deployment integrated within the 3GPP Narrow-Band IoT (NB-IoT) mobile operator network.

Index Terms—Anomaly Detection, Cellular IoT, Industrial IoT, Machine Learning, Smart Logistics

I. INTRODUCTION

T
HE proliferation of Internet of Things (IoT) and deployment of massive amount of IoT devices in cyber-physical infrastructure systems such as Smart Factories [1], [2], Smart Grids [3], Smart Logistics [4] and others, brings forward increasing number of cyber-security [5] and property management challenges [6]. For example, Smart Factory or Smart Logistics operations include asset management, intelligent manufacturing, performance optimization and monitoring, planning, human-machine interaction, all of which are not designed with cyber-security protection or data management of Industrial IoT scale [7], [8]. Handling massive IoT device data integrity and device behaviour in real-time industrial IoT operation and management requires novel approaches which are, in recent research, mainly addressed using machine-learning (ML) and deep-learning (DL) techniques [9]–[11]. The ability of ML/DL algorithms to process massive data sets while extracting useful features allow them to quickly identify anomalies and prevent breakdowns, which has potentially broad application space in cyber-physical systems [12], [13].

With the introduction of 5th generation (5G) cellular networks, IoT cyber-physical infrastructure systems are becoming increasingly reliant on cellular networks [14]. 3GPP standardization initiated work on support for Cellular IoT (CIoT) during the 4G Long-Term Evolution (4G LTE) development [15], which resulted in first CIoT technologies such as Narrow-Band IoT (NB-IoT) being introduced in 3GPP Release 13 [16], [17]. This work has since then expanded to Ultra-Reliable Low-Latency Communications (URLLC) and massive Machine-Type Communications (mMTC) services in 5G [18]. As billions of new CIoT devices are expected to be connected worldwide in the following years, providing efficient and automated monitoring and threat detection both at the CIoT devices and within the CIoT network architecture will be critical to securely manage devices and cover this attack surface [19], [20].

In this paper, we propose to augment the 3GPP mobile cellular architecture with additional enhancements that provide support for network-wide anomaly detection (AD) service. Our target is a generic AD CIoT service which can be tailored to applications ranging from identifying malfunctioning devices to threat detection for secure CIoT. The proposed hierarchical AD architecture embeds anomaly detection modules (ADMs) both at the IoT devices (ADM-EDGE) and in the mobile core network (ADM-FOG). The ADM modules are based on deep autoencoders (AE) whose complexity is matched to both the edge and the fog deployment, balancing between the system responsiveness and accuracy. The distinguishing feature of our work is that the proposed AD enhancement of CIoT architecture, including both ADM-EDGE and ADM-FOG modules, is implemented and deployed in a real-world CIoT network based on 3GPP NB-IoT standard and demonstrated in the context of Smart Logistics. Moreover, we custom-designed a novel NB-IoT device platform for Smart Logistics use case, where NB-IoT devices are connected to shipping containers in a factory supply chain, in order to collect data, deploy and test the ADM-EDGE module.

The paper is organized as follows. In Sec. [II] we provide technical background, review the related work and present the contributions of this paper. The proposed solution for DL-based anomaly detection in CIoT is presented in detail in Sec. [III] In Sec. [IV] we describe system integration, data generation and provide numerical results from real-world experiments. The paper is concluded in Sec. [V].
II. BACKGROUND

In this work, we augment the CIoT architecture with anomaly detection capabilities at the IoT devices (edge) and the mobile core network servers (fog). Before going to details, we first provide the technical background needed for understanding the proposed system architecture and functionality.

A. 3GPP Cellular IoT Architecture

We start by describing current state-of-the-art CIoT architecture focusing primarily on 3GPP NB-IoT technology [15], [16]. NB-IoT is a new CIoT technology that can be seamlessly integrated in existing 3GPP 4G/5G architecture, coexisting in the radio access network with the current 3GPP 4G LTE and the emerging 3GPP 5G NR technology, and using the same evolved packet core (EPC) network functionalities [22]. Focusing on the current 3GPP 4G LTE architecture, relevant 3GPP CIoT architecture elements are illustrated in Fig. 1. CIoT user equipment (CIoT UE), which is a formal name for NB-IoT device, connects to the network via a neighbouring base station or eNodeB (eNB), which is the main element of Evolved Universal Terrestrial Radio Access Network (E-UTRAN). NB-IoT downlink/uplink resources are allocated either within 4G LTE band (in-band deployment), at its edge (guard-band deployment), or as a separate channel (out-of-band deployment). After eNB, both user-plane (i.e., user data packets) and control-plane (i.e., signalling messages) information is processed at CIoT Serving Gateway Node (C-SGN), which covers functionalities of both control-plane Mobility Management Entity (MME) and user-plane Serving Gateway (SGW). User-plane data further flows through Packet Gateway (PGW) to the IoT platform, which forwards data via the Internet to the external network application servers [21].

Two options for data transfer between the CIoT UE and the IoT platform are envisioned. The first one (mandatory) uses signalling radio bearers to transmit user data, thus avoiding establishment of data radio bearers for energy efficiency. From eNB, data is routed either following a control-plane path via an EPC element called Service Capability Exposure Function (SCEF) for non-IP data, or a user-plane path via C-SGN and PGW for both IP/non-IP data. The second one (optional) establishes a data radio bearer to send IP/non-IP data via an eNB/C-SGN/PGW user-plane path to the IoT platform. Herein, we assume that a UDP encapsulated IP data from CIoT UE device traverses the path following the latter approach, which will impact the deployment choices for the proposed anomaly detection enhancements strategy described in Sec. III.

B. Machine Learning for Anomaly Detection at the Edge

Security challenges and threats in industrial IoT networks call for innovative applications of ML/DL techniques for IoT security. More specifically, these techniques can be employed for authentication and access control, anomaly and intrusion detection, malware analysis and distributed denial-of-service (DDoS) attacks detection and mitigation [23], [24]. The main challenges of implementing ML/DL models at the edge are scalability issues and IoT edge platforms resource limitations [13]. 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. Also, the requirements in regard to the memory capacity and computational power depend heavily on the choice whether the models are trained on the edge, or pre-trained models are being used.

Besides the sensor readouts, which are the primary source of data for ML/DL at the edge, the IoT module itself can provide a host of useful insights about the network and wireless link conditions, the feature we also exploit in our edge device design described in Sec. III-B. The amount of useful data that can be extracted from the IoT module generally exceeds the capacity of the wireless communication channel, however, this kind of metadata can be used to feed a locally run ML algorithm for anomaly detection, or be aggregated and sent to the core network fog gateway periodically, for further analysis. In this work, to perform AD, we apply deep autoencoders (AE). AE is a neural network that learns a latent lower-dimensional representation of training data by reproducing its inputs through latent variables in the hidden layers at the output layer with the smallest possible error. The error function

---

Fig. 1. 3GPP CIoT architecture augmented with Anomaly Detection enhancements.
captures differences between values at the input and output layer. This so-called reconstruction error is used as the outlier score in an anomaly detection process. The proposed AD architecture is hierarchical, as it comprises AD models running at different levels within an CIoT system (both IoT edge devices and core network fog gateway), where more powerful higher-level models are activated if decisions of lower-level models have low confidence scores (see Sec. III-C for details).

C. Related Work

Recent research efforts in the area of ML methods for anomaly detection at the edge IoT devices have been focused on efficient utilization of the limited computation resources at the edge. It is well-known that the training process for most of the deep learning-based AI models is highly resource-intensive, usually requiring hardware resources (e.g., GPU, FPGA) [25]. Resource-aware edge AI models designs have been considered in a different line of research. The AutoML idea [26] and the Neural Architecture Search techniques [27] have been used to devise resource-efficient edge AI models tailored to the hardware resource constraints of both the underlying edge devices and network servers. Important research advances were also made regarding the tailored design of DL architectures for resource-constrained devices: Zhang et al. [30] proposed an extremely efficient convolutional neural network (CNN) for mobile devices and Nikouei et al. [31] introduced a lightweight CNN that can run on edge devices [28].

A number of proposals using distributed ML/DL for security in Industrial IoT are recently considered [29]. In CIoT, a recurrent neural network (RNN) is trained for each device type present in the IoT network to learn a normal communication profile. A federated (distributed) learning scheme is employed to learn device-type specific RNNs [30]. Wang et al. [31] proposed a control algorithm that determines the best trade-off between local update and global parameter aggregation. In the proposed decentralized architecture, every IoT device monitors its own data as well as neighbor IoT devices to detect internal and external attacks [32]. Meidan et al. [32] proposed N-BaIoT – a method for detecting IoT botnet attacks based on deep autoencoders. For each device present in a IoT network, a deep autoencoder is trained on features extracted from normal traffic data [33]. Bezerra et al. [33] proposed IoTDS – a distributed method for detecting IoT botnet attacks based on lightweight one-class classification models [34]. Rathore and Park created a decentralized attack detection framework for IoT networks based on semi-supervised learning employing extreme learning machines and fuzzy C-means algorithms [35]. Doshi et al. [36] employed various machine learning algorithms (k-nearest neighbor, support vector machines, decision trees and neural networks) to detect DDoS attack traffic in consumer IoT devices [37]. Pajouh et al. [37] proposed a malware detection approach for IoT based on deep RNNs [37], while [38] presents an approach to anomaly detection that implements autoencoders at each edge device, while the edge devices are orchestrated via a federated learning model with the central server. In [39], authors show that Random Forest, Multilayer Perceptron, and Discriminant Analysis models can viably save time and energy on the edge device during data transmission, while K-Nearest Neighbors, although reliable in terms of prediction accuracy, is resource-inefficient in their studies.

D. Contributions

We now summarize the main contributions of the paper. We propose an approach to embed anomaly detection capabilities in the Cellular IoT architecture, providing for combined threat detection both at the IoT devices (edge) and in the mobile core network servers (fog). The corresponding architecture design is motivated by and well-suited for Smart Logistics. The proposed edge-based ADM-EDGE and fog-based ADM-FOG modules can balance between the responsiveness and accuracy by employing deep autoencoder (AE) based learning modules whose complexity is matched to both edge and fog deployment. We carry out implementation, integration, and evaluation of an end-to-end testbed according to the proposed architecture. This includes: 1) real IoT data generation and emulation of a real-world Smart Logistics scenario; 2) fabrication and configuration of the relevant edge and fog hardware and infrastructure; 3) development and implementation of a software library for edge and fog-based anomaly detection; and 4) evaluation of the developed anomaly detectors on the generated data and quantification of detection performance-tradeoffs. For the latter contribution, we explicitly quantify the tradeoffs that take into account limited computational and storage budget at the edge devices, and communication and processing costs due to processing larger amounts of data at the fog for improved AD performance.

III. DL-BASED ANOMALY DETECTION IN 3GPP NB-IoT

In this section, we describe in detail the design and system architecture of the proposed AD support for the 3GPP NB-IoT mobile cellular network.

A. System Model and Architecture

We augment 3GPP CIoT system architecture with support for CIoT device anomaly detection. Augmented architecture is illustrated in Fig. 1 and introduces two additional ADMs: one placed at the edge CIoT UE (ADM-EDGE) and another placed at the fog gateway (ADM-FOG). The architecture represents generic CIoT enhancement for anomaly detection, although in this work, we specialize it to the domain of Smart Logistics. This includes managing supply of items from various origin points delivered to warehouses in manufacturing plants (Fig. 1). Items being delivered are packed into containers, each of which has an NB-IoT device attached. For this purpose, we designed an entirely new NB-IoT UE device, and deployed suitable ADM-EDGE and ADM-FOG modules at both NB-IoT UEs and the FGW server within the mobile core network.

1Response time is the time passed from the occurrence of an anomaly to its detection
**ADM-EDGE:** As described below, NB-IoT devices collect various information such as acceleration and GPS coordinates. This sensory information can be used to detect anomalies such as physical tampering of items, container mishandling such as overturning, delays, routing problems, incidents with the delivery vehicles, etc. We assume each NB-IoT device possesses two types of sensors: i) sensor S1 with low sampling rate $f_1$ [Hz] and sampling period $\Delta_1 = \frac{1}{f_1}$ [s] (in our case, we consider GPS sensor that samples the outdoor device location), and 2) sensor S2 with high sampling rate $f_2$ [Hz] and sampling period $\Delta_2 = \frac{1}{f_2}$ [s] (in our case, we consider accelerometer/gyroscope that samples vibration monitoring parameters), as illustrated in Fig. 2.

Due to limited memory capacity and processing power, ADM-EDGE integrated into an NB-IoT device firmware requires restrictive design. ADM-EDGE consists of a pre-trained autoencoder with a single hidden layer. At the input, ADM-EDGE processes a single data point that consists of a single S1 and S2 value. As illustrated in Fig. 2, we assume ADM-EDGE is triggered synchronously with the low-rate sensor S1 outputs $X_{S1}[k] = X_{S1}(t = k\Delta_1), k = \{1, 2, \ldots\}$, where $\Delta_1$ is the sampling period of the sensor S1 output function $X_{S1}(t)$. Besides an S1 sample, ADM-EDGE is fed with the sensor S2 value $X_{S2}[k]$, which is a root mean square (RMS) aggregate value of high-rate sensor S2 output samples calculated over the interval of duration $\Delta_1$ between the last two S1 outputs. In other words, $X_{S2}[k] = \sqrt{\frac{1}{M}\sum_{i=1}^{M} X_{S2}^2(t = \ell\Delta_2)}$, where $\ell$ satisfies $(k-1)\Delta_1 < \ell\Delta_2 \leq k\Delta_1$, which amounts to the last $M = \frac{\Delta_2}{\Delta_1}$ S2 samples preceding $t = k\Delta_1$. To summarize, a pair of S1 and aggregated S2 values $(X_{S1}[k], X_{S2}[k])$ represents a data point fed into an ADM-EDGE autoencoder every $\Delta_1$[s]. For each decision, after ADM-EDGE processing time, the device outputs a confidence score (see Sec. II-C).

**ADM-FOG:** NB-IoT devices connect to a mobile network and transfer data via the nearest base station. Each ADM-EDGE data point is forwarded to the FGW, adjoining with the ADM-EDGE confidence score evaluated from the last available data point. The communication delay incurred by NB-IoT network connection may vary between the order of tens-of-milliseconds to several tens-of-seconds, depending on the NB-IoT device radio conditions and network load. FGW server runs an instance of ADM-FOG relying on higher memory capacity and processing power. Thus ADM-FOG uses more powerful autoencoder processing multi-variate time series through several hidden layers. Larger input is considered which is formed by concatenating the last $L$ ADM-EDGE data points (see Fig. 2). Thus at the time instant $t_k$ when the $k$-th data point is received at the FGW (note that $t_k = k\Delta_1 + \tau_k$, where $\tau_k$ is communication delay of the $k$-th data point), the ADM-FOG is triggered with the input containing the set of the last $L$ data points $(X_{S1}[i], X_{S2}[i])_{k-L<i<k}$ received prior to the time instant $t_k$. Decisions made by ADM-EDGE are revised in case of confidence scores below certain threshold.

To summarize, the above AD-augmented CIoT architecture features several important properties: 1) ADM-EDGE at the NB-IoT node immediately detects anomaly over a single data point which may result in extremely fast response time (order of milliseconds) [40]; 2) ADM-FOG collects time series of specific lengths matched to the more powerful AE design through a communication channel that can be a bottleneck and cause unpredictable delays (order of seconds); 3) Only ADM-EDGE has access to raw data, while ADM-FOG gets access to aggregated data; 4) ADM-FOG applies deep learning analyses over the longer time series of data points using more powerful AE design with more hidden layers, requiring higher processing power and memory capacity unavailable at the edge; 5) In the worst-case scenario, anomaly detection decision at the system level is obtained within the time frame of several seconds. It is worth noting that this response time meets the requirements and is well-aligned with the targeted Smart Logistics applications.

**B. NB-IoT Edge Device Design**

We designed the NB-IoT edge device illustrated in Fig. 3 having in mind the specific requirements of a Smart Logistics environment: tracking and monitoring the vibration of the shipping containers. Here, we reflect on the most important features supported by our device.

1) **Cellular connectivity:** To fulfill the requirement for ubiquitous connectivity, while keeping the power consumption of the battery-powered device low, we utilize a BG96 cellular module from Quectel, which supports NB-IoT and LTE-M, as state-of-the-art 3GPP CIoT communication standards, that will be further evolved in 5G standardization [42]. In addition, EGPRS is supported to ensure the connectivity in areas where LTE carrier might not be available. Finally, the integrated

---

![Fig. 2. 3GPP CIoT Anomaly Detection processing flow.](image-url)
GNSS module provides the geolocation information which is essential to the asset tracking task in the logistics use case. The intention is to use NB-IoT as the primary means of communication due to its desirable properties, namely energy efficiency combined with extended coverage [41]. However, in occasions when it is necessary to transfer larger amounts of data, (e.g. a new firmware image), LTE-M is more efficient solution. The architecture of our edge node provides flexibility which allows us to adapt the throughput of the communication module according to the needs of the application.

2) On-board sensors: Apart from the localization data provided by the GNSS module, the on-board environmental sensors are used to measure parameters relevant to the logistics use case. The 6-axis Inertial Measurement Unit (IMU) provides information about the vibrations and the magnetic field along X, Y and Z axes relative to the chip position. The additional set of sensors is used to measure the atmospheric conditions such as air temperature, pressure and humidity.

The designed platform provides additional metadata that could be used as inputs to ADM-EDGE. For example, the cellular modem is capable of providing the standard set of radio condition metrics (SNR, RSSI, RSRP, etc.). In addition, our design includes the on-board current measuring circuitry that allows the micro-controller unit (MCU) to acquire precise measurements of the power consumption by BG96 module.

3) The MCU features and capabilities: The main MCU inside edge node is a low-power 32-bit ARM Cortex M0+ with 256KB of FLASH and 32KB of SRAM, operating at 16MHz. The MCU resources are sufficient to efficiently control the rest of the circuitry, while maintaining the low power consumption, especially in the sleep mode. However, 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, and is used to store the firmware images during over-the-air updates.

4) Security: In an industrial setup, the security is of the critical importance. Thereby, we use hardware crypto element which enables offloading the computationally expensive asymmetric cryptographic algorithms (ECC and RSA) from the resource-constrained MCU [43]. Tampering-resistant memory within the crypto chip is used to store security credentials, making FW on the host MCU oblivious of the sensitive information such as the encryption keys and certificates.

C. Anomaly Detection using ADM-EDGE and ADM-FOG

ADM-EDGE and ADM-FOG autoencoders identify anomalies according to the previously described rule. For each anomaly detection decision, the confidence score $C(y)$ is computed according to the following formula:

$$
C(y) = \sigma(\text{Err}\{y, A(y)\} - \epsilon),
$$

where $\text{Err}\{\cdot\}$ is the error function used to train $A$ (e.g., the mean squared error) and $\sigma$ denotes the sigmoid function. The important property of the confidence score function is that non-anomalous data points have scores in the range (0, 0.5], whereas anomalous data points exhibit scores higher in the interval (0.5, 1). In other words, confidence scores close to 0 indicate non-anomalous data points, while values close to 1 signify anomalies. Thus, confidence scores for non-anomalous data points after making decision are further transformed into $1 - C$, where $C$ is a value obtained by Eq. (1).

ADM-EDGE autoencoders have a predefined structure with a single hidden layer containing $n/2$ nodes, where $n$ is the number of input features. They use the ReLU activation function for the hidden layer. Additionally, bias variables are not considered for internal nodes. Due to constraints of NB-IoT devices, the training of lightweight autoencoders is performed offline using a Python module utilizing the Tensorflow library. This ADM module determines lightweight autoencoder weights by optimizing the mean squared error using the Adam optimizer [44] for a given number of epochs and batch size. Before training, data points in the input training dataset are normalized such that each feature has zero mean and unit variance. The weights of the trained model and data normalization parameters are then exported to textual files. An inference function performing anomaly detection on a pre-trained lightweight autoencoder is implemented in C without relying on any external library. This inference function is directly integrated into the firmware of our NB-IoT devices.
Decisions made by ADM-EDGE lightweight autoencoders are re-evaluated by ADM-FOG autoencoders in case of low confidence scores. The default value of the threshold is set to $C_{th} = 0.75$, i.e., the decisions with $C < C_{th}$ are re-evaluated. We adopt here a standard, confidence-score based decision that is simple but effective; for more advanced mechanisms on how to offload decisions from the edge, see, e.g., [45]. The threshold $C_{th}$ is a tunable parameter that allows to trade-off confidence in the decision about anomaly and response time. Lower threshold corresponds to the system designer’s satisfaction with lower confidence scores, but the average response time within a time interval for the same input data set is decreased. In contrast to ADM-EDGE lightweight autoencoders, ADM-FOG autoencoders may have an arbitrary number of hidden layers. Additionally, they process multivariate time series constructed using the sliding window approach instead of single data points.

IV. System Integration, Data Generation and Numerical Results

A. System Integration

To integrate the system, collect real-world data and perform testing and evaluation, CIoT UE is connected to the FGW via a mobile operator macro-cellular NB-IoT eNB. CIoT UE is running ADM-EDGE software module and periodically sends data points to the FGW encapsulated into UDP packets. Within the mobile operator core network, the general purpose server is set and connected to the PGW gateway. ADM-FOG software module within the server accepts UDP packets sent by CIoT UE. The server provides sufficient resources to run ADM-FOG module, so in the sequel, we focus on the ADM-EDGE module deployment on the CIoT UE device.

To estimate the resource utilization of ML/DL ADM-EDGE model in terms of memory footprint the following results are given in Table I. One can note that ADM-EDGE consumes a small fraction of standard NB-IoT device firmware needed for basic device sensing, processing and communication functionality. Tensorflow and Tensorflow lite exported models sizes are also given for reference.

| MODEL | Size in bytes |
|-------|---------------|
| Firmware without ADM-EDGE | 55816 (21.3%) out of 262144 |
| Firmware with ADM-EDGE | 61896 (23.6%) out of 262144 |
| ADM-EDGE only | 6080 (~2%) |
| Tensorflow ADM | 21596 |
| Tensorflow lite ADM | 1452 |

B. Data Generation

To generate the dataset (elaborated in Section IV-C), we used NB-IoT edge nodes described in Section III-B. We created a setup where an edge node has been attached to a box-shaped container inside a transport vehicle moving through the city of Novi Sad. The device was initially connected to the NB-IoT network, and it had the uninterrupted connectivity throughout the path. We collected the positioning data from GNSS module (timestamp, latitude, longitude, altitude, speed and number of satellites in range), as well as the outputs of the IMU (acceleration and magnetic field along the 3 spatial axes). The time resolution of the GNSS samples was $\Delta_1 = 10$ s. The sampling rate of the IMU is $\Delta_2 = 15$ ms (see Fig. 4 for an example of IMU signals), thus we calculated the RMS for the acceleration and magnetic field samples collected within a sampling interval $\Delta_1$ (as described in Sec. III.A). The collected data was stored at the database at the FGW, and were used to train the AD model discussed in the following section.

![Example of acceleration data from IMU.](image)

C. Numerical Results

ADM-EDGE and ADM-FOG autoencoders were evaluated using two independent datasets. The first dataset reflects the behaviour of the edge node device under normal driving conditions without large disturbances. This dataset contains 1470 data points collected in a period of three days and it is used to train ADM-EDGE and ADM-FOG autoencoders.

The trained autoencoders were tested on the second dataset. The test dataset has 318 data points collected in a single day with 10 intentionally caused anomalous events induced by shaking and overturning the container with the attached device. Since the edge node records both location-based features (GPS longitude and latitude) and IMU-based features, we can distinguish two types of anomalous events: location-based anomalies (large deviations from learned trajectories) and behaviour-based anomalies (large deviations from learned IMU signals). Our test dataset does not contain location-based anomalies.

The accuracies of ADM-EDGE and ADM-FOG autoencoders were assessed by computing the following basic measures:

- $TP$ (true positives) – the number of correctly identified anomalous events,
- $FP$ (false positives) – the number of times an autoencoder indicated a non-existing anomalous event, and
- $FN$ (false negatives) – the number of times an autoencoder missed to indicate an existing anomalous event.

We define the anomalous data points as those that correspond to the intentionally caused incident events; these data points are known to the experiment designer and system evaluator but are not known beforehand to the AD modules. The goal of AD is then to uncover the defined anomalies from the data.
From $TP$, $FP$ and $FN$ we have derived the precision ($P$) and recall ($R$) scores of our anomaly detection models: $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$. Both precision and recall take values in the range $[0, 1]$. Precision indicates the degree of correctness of an anomaly detection model: small precision values imply that the model makes a lot of errors when stating anomalous events. Recall reflects the degree of model’s ability to detect existing anomalous events. Small recall values indicate that the model often remains "silent" in cases when it should alarm anomalous events.

When comparing different anomaly detection models it is useful to have a single overall score reflecting their performances. For this purpose we have used the $F_1$ measure which is the harmonic mean of precision and recall: $F_1 = \frac{2 \cdot P \cdot R}{(P + R)}$.

For the ADM-FOG model we have a greater flexibility than for the ADM-EDGE model. Thus, in our experimental evaluation, we have examined a single ADM-EDGE model (see Sec. III-C), 10 ADM-FOG models with three hidden layers (sequentially containing $n/2$, $n/4$ and $n/2$ nodes, where $n$ denotes the number of input features) accepting time-series of lengths between $L = 1$ to $L = 10$, and 10 ADM-FOG models with five hidden layers (sequentially containing $3n/4$, $n/2$, $n/4$ and $3n/4$ nodes) also working with time-series of lengths between $L = 1$ and $L = 10$. Due to the stochastic nature of the autoencoder learning algorithm, an ensemble of 20 autoencoders was trained for each examined model. All autoencoders were trained in maximally 200 epochs, with the batch size equal to 16, the learning rate of the Adam algorithm was set to 0.001 and early stopping was activated after 10 epochs without a decrease in the value of the loss function.

The evaluation metrics for a particular model were estimated by averaging results individually obtained from all autoencoders in the corresponding ensemble. Additionally, for each model we have examined two variants: a model trained without location-based features a model trained on all features.

The results of the evaluation of the ADM-EDGE autoencoder in both variants (with and without location-based features used) are summarized in Table II. It can be seen that the ADM-EDGE autoencoder working without location-based features has a slightly larger precision score and a slightly lower recall score compared to the ADM-EDGE autoencoder trained on all features. However, the observed differences are not significant which is evident by similar values of $F_1$ scores. This result is expected since the test dataset does not contain location-based anomalies. Therefore, small differences in the obtained results can be explained by the stochastic nature of the autoencoder learning algorithm. The obtained values of precision and recall indicate that the ADM-EDGE autoencoders have a quite good performance. Describing results in more practical terms, on average, the ADM-EDGE anomaly detection model was able to recognize 8 out of 10 existing anomalous events, it missed 2 real anomalous events and it has 1 or 2 false positive alarms (the average number of false positives in the NO-GPS case is 1.25, while the average number of false positives in the WITH-GPS case is 1.85).

In the second experiment we have examined the performance of ADM-FOG autoencoders with 3 and 5 hidden layers.

The obtained F1 scores are presented in Figures 5 and 6. It can be seen that ADM-FOG autoencoders exhibit significantly higher F1 scores compared to ADM-EDGE autoencoders for all timeseries lengths except for the time-series length equal to $L = 1$ (i.e., individual data points). The average improvement in the F1 score when offloading anomaly detection decisions to ADM-FOG is approximately 7%. Similarly as for ADM-EDGE autoencoders, the location-based features do not have a significant impact to the performance of ADM-FOG autoencoders. The performance of ADM-FOG autoencoders with 3 hidden layers is similar to those with 5 hidden layers: the largest difference in F1 scores is equal to 0.027 (excluding ADM-FOG models working with timeseries of length 1).

The results above allow us to explicitly quantify trade-
offs between performance of anomaly detection and response time, with respect to whether the decision on the presence of anomalies is carried out at the edge or at the fog. For this, note that the response time of ADM-EDGE corresponds approximately to one sampling period $\Delta t$. On the other hand, the response time of ADM-FOG depends on the length $L$ of the time series processed. In the case of ADM-FOG autoencoders trained without location-based features, the largest $F_1$ score is achieved by the autoencoder with 3 hidden layers working on time-series of length $L = 9$. The increase in precision and recall compared to the corresponding ADM-EDGE autoencoder is equal to 0.02 and 0.15, respectively. Therefore, that by increasing the confidence threshold for offloading anomaly detection decisions to the ADM-FOG autoencoder the whole system has less false negative decisions at the cost of decision delays by $L = 9$ time slots. The ADM-FOG autoencoder with 5 hidden layers working on time-series of length $L = 10$ has the highest $F_1$ scores among FOG models trained on all features. The increase in precision and recall in this case is 0.1 and 0.05, respectively. Therefore, by increasing the offloading threshold the performance of the whole system improves by having less false positive decisions at the cost of decision delays by $L = 10$ time slots.

V. CONCLUSION

In this paper, we present the design, implementation and real-world deployment and evaluation of a novel anomaly detection architecture for Cellular IoT networks. Our system, tailored for Smart Logistics use case, demonstrated the major system-design trade-offs involving proper balance between responsiveness vs accuracy of deploying anomaly detection at the edge or in the fog of the Cellular IoT network.

REFERENCES

[1] L. Da Xu, W. He, S. Li, “Internet of things in industries: A survey,” IEEE Trans. on Industrial Informatics, Vol. 10, No. 4, pp. 2233–2243, 2014.
[2] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee and B. Yin, “Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges,” in IEEE Access, Vol. 6, pp. 6505–6519, 2018.
[3] X. Fang, S. Misra, G. Xue and D. Yang, “Smart Grid — The New and Improved Power Grid: A Survey,” in IEEE Communications Surveys & Tutorials, Vol. 14, No. 4, pp. 944–980, 2012.
[4] X. Tang, “Research on Smart Logistics Model Based on Internet of Things Technology,” IEEE Access, Vol. 8, pp. 151150–151159, 2020.
[5] K. Sha, W. Wei, T. A. Yang, Z. Wang, W. Shi, “On security challenges and open issues in Internet of Things,” Future Generation Computer Systems, Vol. 83, pp. 326–337, 2018.
[6] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz and J. Lopez, “A Survey of IoT-Enabled Cyberattacks: Assessing Attack Paths to Critical Infrastructures and Services,” in IEEE Communications Surveys & Tutorials, Vol. 20, No. 4, pp. 3453–3459, 2018.
[7] N. Miloslavskaya, A. Toltstoy, “Internet of Things: information security challenges and solutions,” Cluster Comput., Vol. 22, pp. 103—119, 2019.
[8] H. Hindy et al., “A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems,” in IEEE Access, Vol. 8, pp. 104650–104675, 2020.
[9] W. Sun, J. Liu, Y. Yue, “AI-enhanced offloading in edge computing: When machine learning meets industrial IoT,” IEEE Internet of Things Journal, Vol. 33, No. 5, pp.68–74, 2019.
[10] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan and R. Jain, “Machine-Learning-Based Network Vulnerability Analysis of Industrial Internet of Things,” IEEE Internet of Things Journal, Vol. 6, No. 4, pp. 6822–6834, 2019.
[11] 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.
[12] R. Chalapathy, S. Chawla, “Deep Learning for Anomaly Detection: A Survey,” 2019, https://arxiv.org/abs/1901.03407v2.
[13] M.G. Sarwar Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, “Cybersecurity at the Network Edge: A Survey,” 2020, https://pdfs.semanticscholar.org/9f89/00080.pdf.
[14] M. Muller, D. Behnke, P.B., Bok, M., Peuster, S. Schneider, H. Karl, “5G as Key Technology for Networked Factories: Application of Vertical-specific Network Services for Enabling Flexible Smart Manufacturing,” IEEE ENDI 2019, pp. 1495–1500, 2019.
[15] Third Generation Partnership Project, Technical Report 45.820 v13.0.0, Cellular System Support for Ultra-Low Complexity and Low Throughput Internet of Things, 2016.
[16] A. Rico-Alvarino, M. Vajapeyam, H. Xu, X. Wang, Y. Blankenship, J. Bergman, T. Tironen, E. Yavuz, “An overview of 3GPP enhancements on machine to machine communications,” IEEE Communications Magazine, Vol. 54, No. 6, pp. 14–21, 2016.
[17] E. Rastogi, N. Saxena, A. Roy, D.R. Shin, “Narrowband Internet of Things: A Comprehensive Study,” Computer Networks, vol. 173, 2020.
[18] F. Ullah, H. Naeem, S. Jabbar, S. Khalid, M.A. Latif, F. Al-Turjman, L. Mostarda, “Cyber security threats detection in internet of things using deep learning approach,” IEEE Access, Vol. 7, pp. 124379–124389, 2019.
[19] 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.
[20] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li and S. Han, “Amp: Automl model compression and acceleration on mobile devices,” European Conference on Computer Vision, Springer, pp. 815–832, 2018.
[21] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” Proceedings of the International Conference on Learning Representations, Toulon, France, 2019
[22] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B. Choi and T. Faghnav, “Smart Surveillance as an Edge Network Service: From Harr-Cascade, SVM to a Lightweight CNN,” Int’l Conference on Collaboration and Internet Computing (CIC), pp. 256–265, 2018.
[23] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, “A distributed deep learning system for web attack detection on edge devices,” IEEE Trans. on Industrial Informatics, Vol. 16, No. 3, pp.1963–1971, 2019.
[24] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan and A. Sadeghi, “DIO&T: A Federated Self-learning Anomaly Detection System for IoT,” IEEE ICDCS 2019, pp. 756–767, 2019.
[25] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, K. Chan, “Adaptive Federated Learning in Resource Constrained Edge Computing Systems,” IEEE Journal on Selected Areas in Communications, Vol. 37, No. 6, pp. 1205 – 1221, 2019.
[26] A. Ferdowsi and W. Saad, “Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things,” IEEE GLOBECOM 2019, pp. 1–6, 2019.
[27] Y. Meidan et al., “N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders,” in IEEE Pervasive Computing, Vol. 8, No. 3, pp. 12–22, 2019.
[28] V. H. Bezerra, V. da Costa, S.B. Junior, R.S. Miani,B.B. Zarpelão, “IoTDS: A One-Class Classification Approach to Detect Botnets in Internet of Things Devices,” Sensors, Vol. 19, No. 14, 2019.
[35] S. Rathore, J. H. Park, "Semi-supervised learning based distributed attack detection framework for IoT," App. Soft Comp., Vol. 72, pp. 79–89, 2018.

[36] R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," IEEE Security and Privacy Workshops SPW 2018, pp. 29–35, 2018.

[37] H. HaddadPajouh, A. Dehghantanha, R. Khayami, K.R. Choo, "A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting," Future Generation Comp. Syst., Vol. 85, pp. 88–96, 2018.

[38] J. Schneible and A. Lu, "Anomaly detection on the edge," IEEE MILCOM 2017, pp. 678–682, 2017.

[39] K. Kamaraj, B. Dezfouli and Y. Liu, "Edge Mining on IoT Devices Using Anomaly Detection," Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 33–40, 2019.

[40] 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.

[41] B. Vejlgaard, M. Lauridsen, H. Nguyen, I.Z. Kovács, P. Mogensen, M. Sørensen, “Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot,” IEEE VTC Spring, pp. 1-5, 2017.

[42] R. Ratasuk, N. Mangalvedhe, D. Bhatoolaul, "Coexistence Analysis of LTE eMTC and 5G New Radio," IEEE PIMRC 2019.

[43] A. Shamsoshoara, A. Korenda, F. Afgah, S. Zeadally, “A survey on hardware-based security mechanisms for internet of things,” arXiv preprint arXiv:1907.12525, 2019.

[44] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", arXiv:1412.6980, 2014

[45] A. Jaddoa, G. Sakellari, E. Panaousis, G. Loukas, and P.G. Sarigiannidis, "Dynamic decision support for resource offloading in heterogeneous Internet of Things environments," Simulation Modelling Practice and Theory, 101, p.102019, 2020.