RESEARCH ARTICLE

Toward the Protection of IoT Networks: Introducing the LATAM-DDoS-IoT Dataset

JOSUE GENARO ALMARAZ-RIVERA1, JESUS ARTURO PEREZ-DIAZ1, JOSE ANTONIO CANTORAL-CEBALLOS1, (Member, IEEE), JUAN FELIPE BOTERO2, (Member, IEEE), AND LUIS A. TREJO1
1School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey, Nuevo Leon 64849, Mexico
2Electronics and Telecommunications Engineering Department, Universidad de Antioquia, Medellin, Antioquia 050010, Colombia

Corresponding author: Jesus Arturo Perez-Diaz (jesus.arturo.perez@tec.mx)

This work was supported in part by the Fondo Regional para la Innovación Digital en América Latina y el Caribe (FRIDA), and in part by the Project “Red temática Ciencia y Tecnología para el Desarrollo (CYTED)” by the Ibero-American Science and Technology Program for Development under the grant CYTED 519RT0580.

ABSTRACT Anomaly detection is a well-known topic in cybersecurity. Its application to the Internet of Things can lead to suitable protection techniques against problems such as denial of service attacks. However, Intrusion Detection Systems based on Artificial Intelligence, as a defense mechanism, need robust data sources to achieve strong generalization levels from the knowledge domain of interest. Therefore, in this research we present the LATAM-DDoS-IoT dataset, which results from a collaboration among Aligo, Universidad de Antioquia, and Tecnologico de Monterrey. The LATAM-DDoS-IoT dataset includes attack traffic to physical Internet of Things devices and normal traffic from real external users consuming actual services from Aligo’s production network. We also compare this new dataset with the Bot-IoT dataset, as the latter is a collection of data used in recent approaches to create detection systems in the Internet of Things domain. Furthermore, we build a smart anomaly-based Intrusion Detection System from our new dataset, training Decision Tree and Multi-layer Perceptron models, for later deployment and evaluation on a Software Defined Networking architecture with physical and virtual components. Before deployment, we obtained an average accuracy of 99.967% and 98.872% with our new dataset’s balanced denial of service and distributed denial of service versions. After deployment, we show that our Intrusion Detection System does not misclassify legitimate traffic and detects more than 90% of the attacks.

INDEX TERMS DDoS attacks, DoS attacks, Internet of Things, intrusion detection systems, LATAM-DDoS-IoT dataset, software defined networking, transfer learning.

I. INTRODUCTION

The Internet of Things (IoT) requires protection techniques against cyberattacks. The massive and spread use of these physical objects connected to the Internet [1] represents a large target that can turn into botnets after hijacking to conduct attacks such as phishing and Distributed Denial of Service (DDoS) [2]. The required level of protection can be obtained by identification and mitigation systems based on Machine Learning (ML) and Deep Learning (DL), which have demonstrated adequate time performance and high accuracy detection rates [3].

Nevertheless, Artificial Intelligence (AI) models need robust data sources to achieve high generalization levels. Due to its low computing power, constrained memory, and heterogeneous device types, IoT can be considered a separate phenomenon from standard computers and servers, requiring datasets that reflect this nature. Moreover, there is currently a notorious absence of publicly available IoT datasets for security research and practice [4]. In addition, current datasets do not allow the modeling of real user traffic and the representation of DDoS attacks against physical IoT devices simultaneously.

The associate editor coordinating the review of this manuscript and approving it for publication was Luca Bedogni1. 
Therefore, in this paper we present a new dataset based on physical IoT devices and real users consuming actual services from a production network to enhance the current state of the art for IoT security, particularly concerning denial of service attacks. We call this new data collection LATAM-DDoS-IoT and can be downloaded from [5]. We provide the ground truth pcap files and the generated network flows, their features, and the labeled categories and subcategories to facilitate the implementation of supervised learning methods. In addition, based on this dataset, we developed a novel Intrusion Detection System (IDS) [6], which we deployed and tested on a Software Defined Network (SDN) with physical and virtual components, obtaining an accuracy above 94% and a recall value of 100%. We chose this environment type due to its administration, transparency, and automation and virtual components, obtaining an accuracy above 94% and a recall value of 100%. We chose this environment type due to its administration, transparency, and automation characteristics [7].

Lastly, we performed two additional experiments to compare our new dataset with the Bot-IoT dataset [8]. The latter is a state-of-the-art solution to create Intrusion Detection Systems in the IoT domain. After performing transfer learning, we observed apparent differences due to the synthetic normal traffic from the Bot-IoT.

We can summarize the main contributions of this work as follows:

- A new publicly available dataset for security researchers and practitioners in the IoT field, based on physical IoT devices and real external users consuming actual services from a production network. This dataset is called LATAM-DDoS-IoT.
- A new IDS based on anomaly detection AI models trained using the LATAM-DDoS-IoT dataset to identify DoS and DDoS attacks and tested on an SDN environment.

The remaining sections of this paper are structured this way: we present the related work in section II. Section III describes our methodology for creating the LATAM-DDoS-IoT dataset, the proposed AI-based IDS, and the process for building and implementing the SDN environment for testing. The results and discussion are provided in section IV. Finally, in section V, we present the conclusions and future work.

II. RELATED WORK

This section describes the most commonly used datasets for network forensics and designing protection systems [4], [9]. Additionally, we present information about the limitations of these datasets and a benchmark between them and the LATAM-DDoS-IoT. In the end, we introduce some Intrusion Detection Systems for IoT networks to define what experimentation tasks are relevant to conduct with our new dataset.

- DARPA: it was first made available in February 1998 [10] and was published by the Lincoln Laboratory of the Massachusetts Institute of Technology. During seven weeks, different attacks were used while creating the dataset, such as Denial of Service (DoS) and Remote to Local (R2L). Although DARPA is still used for performance evaluation on new anomaly detection systems [11], [12], since 2008 [13] it has been considered outdated for modern networks.
- KDD99: it was built based on the DARPA dataset and was published in 1999 [14]. It contains attacks such as DoS, portsweep, and User to Root (U2R). Inherent problems exist in the KDD99 dataset, such as the issue of redundant records, where it has been found that about 78% and 75% of the records are duplicated in the training and testing sets, respectively [15]. This redundant information can cause bias problems in the AI methods used for learning.
- NSL-KDD: it is a newer version of KDD99 published in 2009 [16], from selected records of the complete KDD99 dataset to overcome its inherent problems. However, as stated by the NSL-KDD authors, this new version may not be a perfect representative of existing real networks [15].
- UNSW-NB15: it was created in 2015 [17] by the University of New South Wales, containing nine types of attacks, such as DoS, Backdoors, and Worms. The total number of normal traffic records is 2,218,761, and the number of attack records is only 321,283. Although this dataset is recent and reflects a wide variety of abnormal traffic, the number of records for each attack type is minimal. An IDS trained on this data could be biased towards normal traffic.
- CIC-IDS2017: it was created in 2017 [9] by the Canadian Institute for Cybersecurity. The data were captured over five days, containing normal and attack traffic, including SQL Injection, DoS using GoldenEye [18], and DDoS using the Low Orbit Ion Cannon (LOIC) tool [19] for sending UDP, TCP, and HTTP requests to the victim server. According to the dataset authors, the attack diversity of the CIC-IDS2017 is based on a report released by McAfee in 2016; thus, they argue that the dataset contains the most common and recent attacks up to that moment. Nevertheless, the CIC-IDS2017 dataset did not include IoT devices in their testbed.
- CSE-CIC-IDS2018: it was created in 2018 [20] by the Communications Security Establishment of Canada, and the Canadian Institute for Cybersecurity, as an update to the CIC-IDS2017 dataset. The CSE-CIC-IDS2018 includes prevalent attacks such as DoS using GoldenEye and DDoS using LOIC. However, one of the main differences from the original CIC-IDS2017 is that the testbed includes more Windows machines (hundreds of them in a LAN on Amazon Web Services) divided into five subnets to emulate the departments of a company. Nonetheless, this dataset does not reflect IoT behavior since it still ignores these devices in the network topology.
- Bot-IoT: it was published in 2019 [8] by the University of New South Wales and emulates a smart home with five simulated IoT devices: a fridge, a garage door, a weather monitoring system, motion-activated lights, and a thermostat. It includes attacks such as DoS and
DDoS based on HTTP using GoldenEye and based on UDP and TCP using Hping3 [21]. The normal traffic was generated using Ostinato [22] and collecting data produced by periodic normal connections between the virtual machines (e.g., when transferring files between each other). This dataset is focused on IoT; nevertheless, it does not include physical IoT devices in its testbed. The amount of normal flows is only about 9,000, compared to the attack traffic of millions of flows, presenting severe class balancing issues [23].

- **TON_IOT**: it was published in 2020 [24] by the University of New South Wales and contains simulated and physical IoT devices. The simulated devices using Node-RED [25] include a smart fridge and a thermostat, and the physical devices include two smartphones and a smart TV. The attacks launched against the IoT devices include DDoS and ransomware, and the normal traffic is generated from the publishing and subscribing methods between the mentioned devices to local and public Message Queue Telemetry Transport (MQTT) gateways [26]. Although this dataset includes a variety of simulated and physical IoT devices and different attacks, the normal traffic is still a product of the local testbed activities, so it is not exposed to real users for consuming actual services.

- **CIC IoT**: it was released in 2022 [27] by the Canadian Institute for Cybersecurity and profiles the behavior of physical IoT devices in different scenarios, including cameras, a lamp, and a coffee maker, emulating a smart home. The devices were under DoS attack using LOIC based on HTTP, UDP, and TCP protocols, and also brute force exploiting the cameras’ Real Time Streaming Protocol (RTSP). This dataset includes physical IoT devices; however, it does not contemplate DDoS attacks (only DoS), and it is mainly oriented (from design) to behavioral analysis of different IoT devices (e.g., during idle and powered-on states).

Based on this review, we argue that DoS and DDoS attacks are still prevalent problems faced by networks. These attacks are mainly based on UDP, TCP, and HTTP protocols, with tools such as GoldenEye, Hping3, and LOIC commonly used to launch them. However, this review shows a need for publicly available IoT datasets for network traffic analysis that include real users, such as the one we present in this paper.

Table 1 briefly compares our proposed dataset against the most commonly used IoT datasets in the literature to create defense systems. To the best of our knowledge, the LATAM-DDoS-IoT dataset is the only one that considers physical IoT devices while containing DDoS attacks, as well as normal traffic from real external users consuming actual services from a production network. Including DDoS attacks represents the actual distributed behavior against physical IoT devices; furthermore, the normal category allows the modeling of real user traffic.

Regarding the creation of Intrusion Detection Systems for IoT networks, we can mention some examples, such as [3], [28], [29], and [30], which we consider relevant since they show that a correct feature selection, class balancing, time-performance evaluation, and benchmark of different ML and DL models are essential aspects when deciding if an anomaly-based IDS is suitable for deployment in production environments. This exhaustive level of experimentation is achieved in this paper using the LATAM-DDoS-IoT dataset.

From now on, the names LATAM dataset (for short) and LATAM-DDoS-IoT dataset will be used indifferently.

### III. METHODOLOGY

This section explains the steps we followed to create the LATAM dataset, the data processing, and the hyperparameters for training the AI models of our anomaly-based IDS. Additionally, we present information related to our SDN architecture for the IDS deployment and evaluation.

#### A. LATAM-DDoS-IoT DATASET

The LATAM dataset was designed and created during a collaboration among Aligo, Universidad de Antioquia, and Tecnologico de Monterrey. Thanks to Aligo’s support, we built and implemented a testbed for DoS and DDoS attacks, using physical and virtual components (see Fig. 1).

Four physical IoT devices and one simulated IoT device were the victims of the DDoS attacks. The physical IoT devices were two Google Home Mini, one smart power strip, and one smart light bulb, connected via an access point.

![Image of the LATAM dataset setup](https://aligo.com.co)

1 Aligo. https://aligo.com.co
FIGURE 1. Testbed configuration for the DoS and DDoS attacks launch during the LATAM-DDoS-IoT dataset creation.

The simulated device using Node-RED was a thermostat running on a container in a virtual machine with Fedora Linux.

The attackers were two Kali virtual machines, running Hping3 for launching DoS and DDoS attacks based on UDP and TCP protocols and GoldenEye for the HTTP protocol. Moreover, these attackers used tcpdump [31] to capture the network traffic and Nmap [32] for port scanning to identify open services running on the victims.

The normal traffic was captured from the Gatherer node, connected to a span port, where production activities from Aligo’s customers were collected. The time window to capture such normal traffic was 50 minutes; in contrast, each victim attack’s time window was 10 minutes. Each time window was run after the previous one finished to avoid bias towards the timestamps. The collected pcap files were processed offline using Argus [33] to obtain data at flow-level, structured as the second feature set proposed in [3] for real-time IDS implementations. Although information such as the destination port numbers are a suitable indicator of the targeted services, this second feature set resulted from a Pearson correlation analysis that discarded those attributes.

Table 2 shows the total number of flows collected for both versions of the LATAM dataset (i.e., the LATAM-DoS-IoT and LATAM-DDoS-IoT datasets). The normal traffic is the same for both versions, with 799,187 flows collected during 50 minutes from the span port. The generated columns are shown in Table 3.

We labeled the data with a category column (where 0 means normal and 1 means attack traffic) and a subcategory column (with values from 0 to 3 for Normal, UDP, TCP, and HTTP classes, respectively). This labeling allows the implementation of supervised learning methods, with binary and multiclass classification tasks for detecting and identifying denial of service attacks.

The total number of samples for the LATAM-DoS-IoT dataset is 30,662,911 flows with 20 columns, and for the LATAM-DDoS-IoT dataset is 49,666,991 flows with the same number of columns. Due to the amount of normal traffic generated by actual customers consuming real services from a company, we consider that the LATAM dataset in its two versions (DoS and DDoS) can be used, for instance, to model one-class classifiers [34] for zero-day attacks [35] detection.

Our attacks based on TCP and UDP protocols had a flooding behavior [36]. We successfully made the two Google Home Mini fail, where the Google service remained processing the voice requests (with the status lights on) though it could not respond. Also, we successfully stressed the smart power strip since it did not resist the DDoS attack for more than eight minutes. The light remained on for the smart light bulb, but it could not change color. Hping3 seems to be a strong attack tool for denial of service since GoldenEye (used against the simulated thermostat) did not interrupt the victim. That can be explained because GoldenEye is a low-rate denial of service tool [37], so it requires more time to overwhelm the victim. Table 4 shows the commands used to launch the attacks. These commands were launched from one Kali virtual machine for the LATAM-DoS-IoT dataset and two Kali virtual machines (simultaneously) for the LATAM-DDoS-IoT dataset.

From Table 4, we can see three distinct commands patterns when launching the attacks:
TABLE 2. Total number of flows collected for both versions of the LATAM dataset.

| Victim (IP)                     | Attack | Tool               | Time       | Flows     |
|--------------------------------|--------|--------------------|------------|-----------|
| Google Home Mini (192.168.200.34) | DDoS   |                    |            | 14,804,835 |
|                                | DoS    |                    |            | 7,667,921  |
| Smart Light Bulb               | DDoS   | hping3 UDP         | 10 min.    | 13,053,593 |
|                                | DoS    |                    |            | 7,610,420  |
| Smart Power Strip              | DDoS   |                    |            | 12,632,022 |
|                                | DoS    |                    |            | 6,573,253  |
| Google Home Mini (192.168.200.33) | DDoS   | hping3 TCP         |            | 8,343,462  |
|                                | DoS    |                    |            | 8,002,589  |
| Thermostat                     | DDoS   | GoldenEye (HTTP)   |            | 33,892     |
|                                | DoS    |                    |            | 9,541      |

TABLE 3. Columns description for the LATAM dataset.

| Column         | Description                                      |
|----------------|--------------------------------------------------|
| StartTime      | Record start time.                               |
| TotPkts        | Total number of packets in transaction.          |
| TotBytes       | Total number of bytes in transaction.            |
| LastTime       | Record last time.                                |
| Seq            | Argus sequence number.                           |
| Dur            | Record total duration.                           |
| Mean           | Average duration at records aggregate level.     |
| StdDev         | Standard deviation of the duration at records aggregate level. |
| Sum            | Total duration at records aggregate level.       |
| Min            | Minimum duration at records aggregate level.     |
| Max            | Maximum duration at records aggregate level.     |
| SrcPkts        | Source to destination packets count.             |
| DstPkts        | Destination to source packets count.             |
| SrcBytes       | Source to destination bytes count.               |
| DstBytes       | Destination to source bytes count.               |
| Rate           | Total packets per second in transaction.         |
| SrcRate        | Source to destination packets per second.        |
| DstRate        | Destination to source packets per second.        |
| Category       | Type of binary traffic, either Normal or Attack. |
| Subcategory    | Type of multiclass traffic, either Normal or Attack (UDP, TCP, HTTP). |

TABLE 4. Attack commands for both versions of the LATAM dataset.

| Victim (IP)                     | Port | Command                                               |
|--------------------------------|------|-------------------------------------------------------|
| Smart Power Strip              | 6668 | hping3 -S -udp -flood -d 100 -p 6668 192.168.200.37  |
| Google Home Mini               | 8008 | hping3 -S -udp -flood -d 100 -p 8008 192.168.200.34   |
| Google Home Mini               | 8008 | hping3 -S -flood -d 100 -p 8008 192.168.200.33        |
| Smart Light Bulb               | 6668 | hping3 -S -udp -flood -d 100 -p 6668 192.168.200.32   |
| Thermostat                     | 80   | goldeneye.py http://192.168.200.26:80 -m post -s 75 -w 1 |

1) hping3 -S -udp -flood -d 100 -p 6668 192.168.200.37: this command launches a SYN UDP attack by sending packets as fast as possible, of body size 100 bytes, against destination port 6668 of the IP 192.168.200.37.

2) hping3 -S -flood -d 100 -p 8008 192.168.200.33: this command launches a SYN TCP attack by sending packets as fast as possible, of body size 100 bytes, against destination port 8008 of the IP 192.168.200.33.

3) goldeneye.py http://192.168.200.26:80 -m post -s 75 -w 1: this launches an HTTP attack against port 80 of the IP address 192.168.200.26, using the HTTP method POST, 75 concurrent sockets, and 1 concurrent worker.

From the two versions of the LATAM dataset, we decided to apply the same random selection of consecutive flows sections for each attack type in the same proportion to the normal samples to keep a balanced ratio, as proposed in [3]. Fig. 2 shows the data distribution of the DoS and DDoS versions of the LATAM dataset. In the end, the total dataset size for training the AI models was 2,407,102 samples for the DoS version and 2,431,453 for the DDoS. The difference between both distributions is the number of GoldenEye records, which is less than the other classes due to the low-rate behavior of...
B. DATA PROCESSING AND AI MODELS

HYPERPARAMETERS TUNING FOR TRAINING

Features of both versions of the LATAM dataset were normalized by applying the StandardScaler method from the scikit-learn ML library [38], [39] (see Eq. (1)). For training our models, we did not use either the timestamps or the Argus sequence number from Table 3 (i.e., we only used 15 of the total number of created features).

\[ x' = \frac{x - \text{mean}}{\text{stddev}} \]  

(1)

We decided to train Decision Tree and Multi-layer Perceptron (MLP) models according to the best ML and DL learning methods from [3]. The hyperparameter tuning for these models resulted from an iterative approach to evaluating different configuration settings. For instance, to select the best depth of our Decision Tree, we chose the one with the best accuracy from a finite range of values; for the MLP, we defined the best learning rate by minimizing the range between 0 and 1 using random decimal numbers. We show the values of the final hyperparameters across the different experiments of this work for the MLP in Table 5 and the Decision Tree in Table 6.

We split the dataset into 80% for training, 10% for validation, and 10% for testing. The Decision Tree model was built using scikit-learn, and the MLP model using PyTorch [40].

C. SDN TESTBED FOR THE IDS DEPLOYMENT

To deploy our smart IDS, we simulated an SDN infrastructure using Mininet [41] and an ONOS [42] as the network controller. Even though ONOS is more complex in its implementation, we decided to use it because it is one of the most robust controllers and is typically used in production environments. This testbed is an architecture with physical and virtual components, where we have ONOS installed and running as a Linux service in an Ubuntu 18.04.6 LTS machine, but we also have simulated devices using Mininet. The network diagram is depicted in Fig. 3.

The testbed includes one physical Ubuntu computer where the ONOS controller runs directly as a Linux service. Also, the architecture is structured with two switches; each one of them communicates to four nodes with an IP address in the subnet 10.0.0.x. We used iPerf [43] to generate normal client traffic and implement two servers listening on TCP port 5001. The attacker nodes generate high-rate denial of service attacks [44] based on TCP using Hping3. Next, we describe the commands used during traffic and attack generation:

1) `iperf -s`: this command runs iPerf in server mode. We used this command to implement the server nodes (i.e., the 10.0.0.1 and 10.0.0.8 IP addresses).

2) `iperf -c 10.0.0.1 -p 5001 -t 60 -p 3`: this command runs iPerf in client mode and creates three TCP simultaneous connections to the server 10.0.0.1 at port 5001 for 60 seconds.

| TABLE 5. Summary of the best hyperparameters values for the different experiments using the Multi-layer Perceptron model. |
|---------------------------------------------------------------|
| Hyperparameter | Value |
| Classes | 2 for binary classification and 4 for multiclass classification |
| Batch size | 128 |
| Input size | 15 |
| Hidden size | 512 |
| Layers | 4 |
| Epochs | 100 |
| Optimizer | Adam |
| Loss function | Cross Entropy |
| Learning rate | 0.0011 |
| Device | CPU |

| TABLE 6. Summary of the best hyperparameters values for the different experiments using the Decision Tree model. |
|---------------------------------------------------------------|
| Dataset | Binary classification | Multiclass classification |
|---------------------------------------------------------------|
| LATAM-DoS-IoT | Max depth: 11 | Max depth: 12 |
| Entropy criterion | Entropy criterion |
| LATAM-DDoS-IoT | Max depth: 14 | Max depth: 14 |
| Entropy criterion | Entropy criterion |
| LATAM-Bot-DoS-IoT | Max depth: 12 | Max depth: 12 |
| Entropy criterion | Entropy criterion |
| LATAM-Bot-DDoS-IoT | Max depth: 13 | Max depth: 14 |
| Entropy criterion | Entropy criterion |
3) hping3 -S -faster -d 100 -p 5001 10.0.0.1: this command launches a SYN TCP attack by sending 100 packets per second, of body size 100 bytes, against destination port 5001 of the IP 10.0.0.1.

For the sake of simplicity, we refer to host 10.0.0.1 as h1, 10.0.0.2 as h2, and so on, up to h8. Next, we explain the flow of communications in the SDN testbed using this notation. Legitimate traffic is generated as follows: h2 towards h8; h5 and h6 towards h1. Attack traffic is generated using the following paths: h3 and h4 towards h1, and h7 towards h8. Legitimate traffic is sent simultaneously from the source nodes for 60 seconds. During the last 10 seconds, the attack traffic is launched, overlapping with legitimate traffic. When performing a ping reachability test in the network, we confirmed that this traffic volume was high enough to see 98% of the communications drop.

We have a Java prototype installed and running in ONOS. Currently, this app is limited to working only for TCP traffic, but it can be further enhanced to capture UDP traffic. This is the reason why we only launched TCP traffic from the attackers. This Java app is based on [45] and was developed to capture network traffic, parse the packets into flows, and extract the features. It uses a Java implementation of Flowbag [46], initially written in Go. We extended the code to allow connectivity to our cloud-hosted REST API written in Flask [47], where our logic for flow classification resides. Hence, we were able to structure the JSON request with the variables we needed. Algorithm 1 describes the IDS classification process inside our API.

The results obtained from our SDN testbed and their discussion can be read in section IV.

### Algorithm 1: IDS Classification Process

**Input:** JSON request with the flow parameters.

**Output:** JSON response with the flow classification.

1. Normalize data by applying standardization.
2. Select the model specified in the JSON request.
3. Run the selected model in prediction mode to classify the flow.
4. Return a JSON response with the flow prediction, specifying the model’s name, the classification result, and the corresponding class id.

### IV. EXPERIMENTAL RESULTS AND DISCUSSION

The classification metrics used for evaluation are accuracy, precision, recall, and F1 score. See Eq. (2)–(5) for these metrics’ definitions. For the time performance evaluation, we measured the average number of flows per second each anomaly detection model could analyze, according to [48].

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (3)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (4)
\]

\[
F1 \text{ score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)
\]

### A. LATAM DATASET

The classification results for both versions of the LATAM dataset are shown in Table 7, whilst the time performance results are in Table 8. These results indicate that Decision
Tree is marginally better than MLP, and both anomaly detection models outperform the maximum peak of 1,681 flows per second discussed in [48]. The LATAM-DoS-IoT dataset classification results are slightly better than those of the LATAM-DDoS-IoT dataset. In both LATAM dataset versions, binary classification is slightly better than multiclass classification. The latter can be explained because the low amount of records in the HTTP category affects the predictions in multiclass classification, in contrast to binary classification, where this category is merged with the other attack classes (UDP and TCP).

To extend the proposed smart IDS from [3] and to add real traffic from real customers and physical IoT devices, we decided to conduct two experiments with the previously used balanced version of the Bot-IoT dataset. The first of the experiments applies transfer learning, and the second one implied concatenating both datasets. We provide more details in the following subsection.

B. COMBINING THE LATAM AND BOT-IoT DATASETS

1) TRANSFER LEARNING

Transfer learning is an AI technique focused on transferring knowledge from a source domain to a target domain when both of them share similarities [49]. We want to transfer knowledge from the LATAM dataset to the balanced version of the Bot-IoT dataset [3] because we believe more data (millions vs. thousands of records, respectively) can lead to better generalization of anomaly detection in network flows.

Instead of applying random initializations, we used transfer learning to initialize the weights and biases of our new MLP models. To accomplish this, we took our neural networks trained with the LATAM dataset and froze all their fully connected layers except for the last one in order to train our new and final linear layer with the balanced version of the Bot-IoT dataset. The first of the experiments applies transfer learning, and the second one concatenates both datasets. We provide more details in the following subsection.

2) DATASETS CONCATENATION

For the next set of experiments, we concatenated the balanced version of the Bot-IoT dataset with each balanced version of the LATAM dataset, which allowed the models to learn from both domains from scratch, albeit taking considerably longer to train. This approach led to the best classification results. After concatenation of the Bot-IoT and LATAM-DoS-IoT datasets, the resulting dataset size was 2,443,442 samples; and for the Bot-IoT and LATAM-DDoS-IoT datasets, 2,467,793 samples. From now on, we will refer to the first combination as LATAM-Bot-DoS-IoT and LATAM-Bot-DDoS-IoT for the second case.

The hyperparameters setting applied to the new datasets are shown in Tables 5 and 6. We employed the same tuning approach we have followed so far. Table 7 shows the classification results for both LATAM-Bot-DoS-IoT and LATAM-Bot-DDoS-IoT datasets, and Table 8 shows the time performance.

These time performance results indicate the feasibility of implementing our anomaly detection models in production networks. The classification experiments indicate that learning from scratch improves results, with our models successfully learning information from both domains: the Bot-IoT dataset and the LATAM dataset. The best classification accuracy obtained from transfer learning was 87.342%, compared to the worst MLP accuracy value of 98.654% obtained with dataset concatenation. Since Decision Tree outperforms the MLP in time performance and both classification tasks, we decided to choose it to conduct a final experiment deploying our smart IDS in the SDN topology described in section III.

C. IDS DEPLOYMENT RESULTS IN THE SDN TESTBED

We deployed our Java prototype for TCP traffic over the SDN topology, using the Decision Tree for binary classification from the LATAM-Bot-DDoS-IoT dataset. We obtained an accuracy of 94.608%, a precision of 100%, a recall of 91.406%, and an F1-score equal to 95.51%. Fig. 4 shows the normalized confusion matrix.

Our results show that 100% of the flows the smart IDS identified as attacks were correctly classified, and 91.406% of the attack flows were detected. The F1 score is 95.51%, indicating a good balance between precision and recall. Since the results do not show misclassifying of legitimate traffic and show detection of more than 90% of the attacks, we are confident that our smart IDS is a suitable defense against denial of service attacks in SDN environments.

Compared to the obtained results before the deployment, the accuracy degrades in the SDN scenario, mainly because of the use of Mininet for victim simulation, which is different from the physical IoT victims in the LATAM dataset (i.e., different from the vast majority of the traffic present in the used LATAM-Bot-DDoS-IoT dataset).

After launching the DDoS attacks, we show a ping reachability test from Mininet in Fig. 5, where we can see that 98% of the communication was dropped, and the attack successfully disrupted the network. This test was executed immediately after launching all the traffic on the SDN testbed, and the only ping that remained working was between the attacking

106916
TABLE 7. Classification results across the different experiments.

| Dataset            | Classification | Model    | Accuracy | Precision | Recall | F1 score |
|--------------------|----------------|----------|----------|-----------|--------|----------|
| LATAM-DoS-IoT      | Binary         | Decision Tree | 99.996%  | 99.996%   | 99.996%| 99.996%  |
|                    |                | MLP       | 99.940%  | 99.999%   | 99.912%| 99.955%  |
|                    | Multiclass     | Decision Tree | 99.995%  | 99.995%   | 99.995%| 99.995%  |
|                    |                | MLP       | 99.938%  | 99.938%   | 99.940%| 99.937%  |
| LATAM-DDoS-IoT     | Binary         | Decision Tree | 98.911%  | 98.942%   | 98.911%| 98.915%  |
|                    |                | MLP       | 98.834%  | 99.978%   | 98.287%| 99.125%  |
|                    | Multiclass     | Decision Tree | 98.908%  | 98.857%   | 98.908%| 98.605%  |
|                    |                | MLP       | 98.834%  | 98.793%   | 98.839%| 98.456%  |
| LATAM-Bot-DDoS-IoT | Binary         | Decision Tree | 99.990%  | 99.990%   | 99.990%| 99.990%  |
|                    |                | MLP       | 99.801%  | 99.859%   | 99.843%| 99.851%  |
|                    | Multiclass     | Decision Tree | 99.989%  | 99.989%   | 99.989%| 99.989%  |
|                    |                | MLP       | 99.793%  | 99.791%   | 99.789%| 99.789%  |
| LATAM-Bot-DDoS-IoT | Binary         | Decision Tree | 98.884%  | 98.916%   | 98.884%| 98.888%  |
|                    |                | MLP       | 98.655%  | 98.840%   | 98.156%| 98.991%  |
|                    | Multiclass     | Decision Tree | 98.867%  | 98.868%   | 98.867%| 98.625%  |
|                    |                | MLP       | 98.654%  | 98.658%   | 98.661%| 98.376%  |

FIGURE 4. Normalized confusion matrix for the Decision Tree binary classification results on the SDN testbed. The numbers in the axes mean 0 for the Normal class and 1 for the Attack class.

host h7 and its victim, server h8. This can be explained as ten seconds were not enough for a single attacker to overflow its victim completely. In this experiment, an average of 860,016 packets were transmitted by each attacker node in just ten seconds to the victim nodes h1 and h8 (as presented in section III).

Fig. 6 shows a fragment of our smart IDS logs while classifying traffic in ONOS: we log the classification results, the model specified in the input JSON request to detect the traffic, and the tuple that conforms to the flow (i.e., the source IP, source port, destination IP, destination port, and protocol fields). The white square in the image shows that our smart IDS can detect simultaneous attacks from different source IPs (i.e., DDoS attacks), which in this case correspond to attackers h4 and h7.

D. COMPARISON WITH PREVIOUS WORKS

Unlike state-of-the-art IoT datasets [8], [24], [27], we addressed the modeling of real user traffic and the representation of DDoS attacks against physical IoT victims simultaneously.

In addition, we performed experiments with our IoT dataset in an SDN architecture with physical and virtual components, running ONOS as a Linux service instead of Docker [51], in contrast to the fully virtualized simulation reported in [37].

Moreover, we present time performance as a metric to measure the quality of our Intrusion Detection System, different from other works [28], [30] that are limited to only classification metrics (e.g., the accuracy, precision, recall, and F1-score values).

Although the deployed IDS in [37] shows the best accuracy of 95.01%, their obtained precision was 95.46%, which is less than our precision of 100%. It indicates misclassification of legitimate traffic and a severe problem against normal users, different from our IDS capability, where all of the flows identified as attacks were correctly classified. Furthermore,
In the next section, we conclude this paper and present future work that can be carried out to explore new directions and extend our current efforts.

V. CONCLUSION AND FUTURE WORK

In this work, we presented a new dataset that includes real normal traffic from actual clients consuming production services from a company and also real attack traffic to physical IoT devices. We call this new dataset LATAM-DDoS-IoT and share a DoS version of it.

To leverage the categories and subcategories present in the LATAM dataset, we conducted binary and multiclass classifications with its balanced DoS and DDoS versions, obtaining an average accuracy of 99.967% and 98.872%, respectively. Then, we combined the balanced version of the Bot-IoT dataset from [3] applying transfer learning, showing how the datasets differ from each other. Additionally, we concatenated both datasets in another experiment to get a higher level of generalization from both domains, achieving strong results such as 99.99% accuracy using Decision Tree binary classification for DoS.

Furthermore, we built an SDN architecture using ONOS and Mininet to deploy and evaluate our IDS and coded a Java app to communicate with our cloud-hosted Flask REST API. 100% of the flows identified as attacks by our smart IDS were correctly classified, and it detected the work in [37] uses the CIC DoS dataset [52] for targeting web servers and not considers an IoT environment.

In the next section, we conclude this paper and present future work that can be carried out to explore new directions and extend our current efforts.
above 90% of the attack flows. Our defense system does not misclassify legitimate traffic and presents an average time performance above 30,000 flows per second, which is fast enough for physical deployment.

This work can be extended by creating and deploying an Intrusion Prevention System as a mitigation management strategy, which integrates and communicates with our Intrusion Detection System. The resulting architecture will not only allow us to detect attacks (as we have done) but also stop the identified attackers, diminishing network damage.

Another direction for further improvement is to test our IDS in an entirely physical SDN architecture. To achieve this, the next component to remove would be Mininet to avoid virtualized hosts and implement real physical devices. Moreover, since our Java functional prototype was designed to capture only TCP traffic, a natural improvement would be to consider UDP traffic as well.

In addition, it will be interesting to see what kind of experiments derive from other colleagues’ use of the LATAM-DDoS-IoT dataset, such as one-class classifiers, since the characteristics of real traffic from actual clients and also attack traffic directed to physical IoT devices make our dataset convenient for real production environments.

ACKNOWLEDGMENT
Genaro Almaraz would like to thank the Tecnologico de Monterrey and Consejo Nacional de Ciencia y Tecnología (CONACYT) for the scholarships during his masters degree studies. Additionally, the authors would like to thank Jaime Vergara for his assistance in deploying the SDN testbed and Brian Martinez, Francisco Muñoz, and Juan Hernandez for their valuable time dedicated to the DDoS testbed in Aligo.

REFERENCES
[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of Things: Vision, applications and research challenges,” Ad Hoc Netw., vol. 10, no. 7, pp. 1497–1516, 2012.

[2] S. T. Zargar, J. Joshi, and D. Tipper, “A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks,” IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2046–2069, Mar. 2013.

[3] J. G. Almaraz-Rivera, J. A. Perez-Diaz, and J. A. Cantoral-Ceballos, “Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models,” Sensors, vol. 22, no. 9, p. 3367, Apr. 2022.

[4] M. Banerjee, J. Lee, and K.-K. R. Choo, “A blockchain future for Internet of Things security: A position paper,” Dirit. Comput. Netw., vol. 4, no. 3, pp. 149–160, Aug. 2018.

[5] J. G. Almaraz-Rivera, J. A. Perez-Diaz, J. A. Cantoral-Ceballos, J. F. Botero, and L. A. Trejo, “LATAM-DDoS-IoT dataset,” IEEE Access, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/10004300/

[6] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” J. Netw. Comput. Appl., vol. 36, no. 1, pp. 16–24, 2013.

[7] I. A. Valdivinos, J. A. Pérez-Díaz, K.-K.-R. Choo, and J. F. Botero, “Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions,” J. Netw. Comput. Appl., vol. 187, Aug. 2021. Art. no. 103093.

[8] N. Korontis, N. Moustafa, E. Sittikova, and B. Turnbull, “Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset,” Future Gener. Comput. Syst., vol. 100, pp. 779–796, Nov. 2019.

[9] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy, 2018, pp. 108–116.

[10] 1998 DARPA Intrusion Detection Evaluation. Accessed: May 9, 2022. [Online]. Available: https://www.ll.mit.edu/rd/datasets/1998-darpa-intrusion-detection-evaluation-dataset.html

[11] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, “A hybrid deep learning-based model for anomaly detection in cloud datacenter networks,” IEEE Trans. Netw. Service Manage., vol. 16, no. 3, pp. 924–935, Sep. 2019.

[12] I. Indre and C. Lemmaru, “Detection and prevention system against cyber attacks and botnet malware for information systems and Internet of Things,” in Proc. IEEE 12th Int. Conf. Intell. Comput. Commun. Process. (ICCP), Sep. 2016, pp. 175–182.

[13] C. Thomas, V. Sharma, and N. Balakrishnan, “Usefulness of DARPA dataset for intrusion detection system evaluation,” Proc. SPIE, vol. 6973, pp. 164–174, Mar. 2009.

[14] KDD Cup 1999 Data. Accessed: May 9, 2022. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[15] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl., Jul. 2009, pp. 1–6.

[16] NSL-KDD Dataset. Accessed: May 9, 2022. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html

[17] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in Proc. Mil. Commun. Conf. Inf. Syst. Conf. (MILCIS), Nov. 2015, pp. 1–6.

[18] Goldeneye Layer 7 (Keepalive+Noack) Dos Test Tool. Accessed: May 2022. [Online]. Available: https://github.com/jseidl/GoldenEye

[19] Last Orbit Ion Cannon (LOIC). Accessed: May 9, 2022. [Online]. Available: https://www.cloudflare.com/learning/ddos/ddos-attack-tools/low-orbit-ion-cannon-loic/

[20] A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018). Accessed: May 9, 2022. [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018

[21] hPing. Accessed: May 11, 2022. [Online]. Available: http://www.hping.org

[22] Ostinato. Accessed: May 11, 2022. [Online]. Available: https://ostinato.org

[23] S. S. Gopalani. (2021). Towards Effective Detection of Botnet Attacks Using Bot-IoT Dataset. Accessed: May 11, 2022. [Online]. Available: https://scholarworks.rut.edu/theses/10069

[24] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, “TON IoT telemetry data set: A new generation dataset of IoT and IOT for data-driven intrusion detection systems,” IEEE Access, vol. 8, pp. 165130–165150, 2020.

[25] Node-RED. Accessed: May 11, 2022. [Online]. Available: https://node-red.org

[26] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Albedhi, and M. Ayyash, “Internet of Things: A survey on enabling technologies, protocols, and applications,” IEEE Commun. Surveys Tuts., vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[27] S. Dadkhi, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, and A. A. Ghorbani, “Towards the development of a realistic multidimensional IoT profiling dataset,” in Proc. 19th Annu. Int. Conf. Privacy, Secur. Trust (PST), Los Alamitos, CA, USA, Aug. 2022, pp. 1–11.

[28] M. A. Ferrag, L. Maglara, S. Moschouyaniss, and H. Janicke, “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study,” J. Inf. Secur. Appl., vol. 50, Feb. 2020, Art. no. 102419.

[29] Y. Zhang, J. Xu, Z. Wang, R. Geng, K.-K.-R. Choo, J. A. Perez-Diaz, and D. Zhu, “Efficient and intelligent attack detection in software defined IoT networks,” in Proc. IEEE Int. Conf. Embedded Softw. Syst. (ICCESS), Los Alamitos, CA, USA, Dec. 2020, pp. 1–9.

[30] A. R. Gad, A. A. Nashat, and T. M. Barkat, “Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset,” IEEE Access, vol. 9, pp. 142206–142217, 2021.

[31] TCPDUMP. Accessed: May 12, 2022. [Online]. Available: https://www.tcpdump.org

[32] Nmap. Accessed: May 12, 2022. [Online]. Available: https://nmap.org

[33] Argus. Accessed: May 12, 2022. [Online]. Available: https://openargus.org

[34] S. S. Khan and M. G. Madden, “One-class classification: Taxonomy of study and review of techniques,” Knowl. Eng. Rev., vol. 29, no. 3, pp. 345–374, Jan. 2014.

[35] L. Bilge and T. Durnitas, “Before we knew it: An empirical study of zero-day attacks in the real world,” in Proc. ACM Conf. Comput. Commun. Secur., New York, NY, USA, 2012, pp. 833–844.
[36] W. Zhijun, L. Wenjing, L. Liang, and Y. Meng, “Low-rate DoS attacks, detection, defense, and challenges: A survey,” IEEE Access, vol. 8, pp. 43920–43943, 2020.

[37] J. A. Pérez-Díaz, I. A. Valdovinos, K.-K. R. Choo, and D. Zhu, “A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning,” IEEE Access, vol. 8, pp. 155859–155872, 2020.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12 no. 10, pp. 2825–2830, 2012.

[39] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, “API design for machine learning software: Experiences from the scikit-learn project,” in Proc. ECML PKDD Workshop, Lang. Data Mining Mach. Learn., 2013, pp. 108–122.

[40] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. Advances in Neural Information Processing Systems, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2019, pp. 8024–8035.

[41] B. Lantz, B. Heller, and N. McKeown, “A network in a laptop: Rapid prototyping for software-defined networks,” in Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw. (Hotnets), New York, NY, USA, 2010, pp. 1–6.

[42] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O’Connor, P. Radoslavov, W. Snow, and A. Parulkar, “ONOS: Towards an open, distributed SDN OS,” in Proc. 3rd Workshop Hot Topics Softw. Defined Netw., vol. 14, New York, NY, USA, 2014, pp. 1–6.

[43] iPerf. Accessed: May 12, 2022. [Online]. Available: https://iperf.fr

[44] C. Zhang, Z. Cai, W. Chen, X. Luo, and J. Yin, “Flow level detection and filtering of low-rate DDoS,” Comput. Netw., vol. 56, no. 15, pp. 3417–3431, 2012.

[45] HTTP DDoS Detector. Accessed: May 12, 2022. [Online]. Available: https://github.com/jatj/httpDetector

[46] Flowbag. Accessed: May 12, 2022. [Online]. Available: https://github.com/DanielArndt/flowbag

[47] Flask. Accessed: May 12, 2022. [Online]. Available: https://palletsprojects.com/p/flask

[48] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, “A GRU deep learning system against attacks in software defined networks.” J. Netw. Comput. Appl., vol. 177, Mar. 2021, Art. no. 102942.

[49] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zha, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” Proc. IEEE, vol. 109, no. 1, pp. 43–76, Jul. 2020.

[50] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, Dec. 2009.

[51] Docker. Accessed: May 12, 2022. [Online]. Available: https://www.docker.com

[52] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, “Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling,” Comput. Netw., vol. 121, pp. 25–36, Jul. 2017.

JOSUE GENARO ALMARAZ-RIVERA received the B.S. degree in computer science from Universidad Autonoma de Nuevo Leon, Mexico, in 2020, and the M.S. degree in computer science from Tecnologico de Monterrey, Mexico, in 2022, where he is currently pursuing the Ph.D. degree in computer science.

He has worked at companies such as Apple and Meta. Also, he was a TEDx speaker at two events at Universidad Autonoma de Nuevo Leon, in 2017 and 2018, with the Social Hacking and Artificial Intelligence topics. His current research interests include self-supervised learning and its potential application to cybersecurity.

JESUS ARTURO PEREZ-DIAZ received the B.Sc. degree in computer science from the Autonomous University of Aguascalientes, in 1995, and the Ph.D. degree in new advances in computer science systems from the Universidad de Oviedo, in 2000. He was a Full Associate Professor at the University of Oviedo, from 2000 to 2002. He is currently a Researcher and a Professor at the ITESM–Campus Queretaro, Mexico. He has published more than 30 scientific papers in several journals and international conferences. His research interests include the mitigation of DDoS attacks in SDN and the design of communications protocols, where he has supervised several master’s and Ph.D. theses in the field. He is a member of the Mexican Researchers National Systems. He received the Best Student Award from the Autonomous University of Aguascalientes. He was recognized by the COIMBRA Group as one of the Best Young Latin-American Researchers, in 2006 and received a Research Stay at Louvain le nouveau University, Belgium. He has been awarded by the CIGRE and by Intel for the development of innovative systems.

JOSE ANTONIO CANTORAL-CEBALLOS (Member, IEEE) received the B.Sc. degree (Hons.) in electronic and communications engineering from the Tecnologico de Monterrey, Queretaro, Mexico, in 2005, and the M.Sc. (Hons.) and Ph.D. degrees from the University of Manchester, Manchester, U.K., in 2009 and 2013, respectively, with a focus on algorithms for tomography imaging in embedded systems. From 2013 to 2014, he was a Postdoctoral Research Associate at the University of Manchester, where he worked on the design and development of a tomography data acquisition system, and imaging algorithms from limited views. From 2015 to 2020, he was at the Advanced Technology Center (CIATEQ A.C.), Mexico, where his research efforts focused on the study of novel deep learning methods to solve digital signal processing and energy problems. He is currently a Research Professor at the Department of Computer Science, Tecnologico de Monterrey, where he continues his research on deep learning solutions to different problems, in particular the study of neurological signals. He was a recipient of the National Instruments Prize for his M.Sc. dissertation, in 2009, and the Best Technical Presentation Award at the Seventh World Congress of Industrial Process Tomography (WCRIPT), Krakow, in 2013.

JUAN FELIPE BOTERO (Member, IEEE) received the Ph.D. degree in telematics engineering from the Technical University of Catalonia, Spain. He is an Assistant Professor at the Electronics and Telecommunications Engineering Department, University of Antioquia, Colombia. In 2013, he joined GITA (Research Group on applied telecommunications) at the Electronics and Telecommunications Engineering Department. His main research interests include quality of service, software defined networking, network virtualization, and resource allocation.

LUIS A. TREJO received the Ph.D. degree in computer science (parallel processing) from the Universidad Claude-Bernard de Lyon, France, in 1993. He is a full-time Professor at the School of Science and Engineering, Tecnologico de Monterrey. Since 2015, he has been a member of CONACyT’s National Research System, Level 1, and a member at the GEE-ML (Machine Learning) Research Group, Tecnologico de Monterrey. His research interests include internetworking, the Internet of Things, information security, intrusion detection and prevention systems, machine learning, data science, and parallel processing.