IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON MULTIMEDIA ANALYSIS FOR INTERNET-OF-THINGS

Received April 24, 2018, accepted May 23, 2018, date of publication June 18, 2018, date of current version June 29, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2842034

INVITED PAPER

An Intelligent System for Video Surveillance in IoT Environments

ALBERT REGO, ALEJANDRO CANOVAS, JOSE M. JIMÉNEZ, AND JAIME LLORET, (Senior Member, IEEE)

Instituto de Investigación para la Gestion Integrada de zonas Costeras, Universitat Politècnica de València, 46730 València, Spain

Corresponding author: Jaime Lloret (jlloret@dcom.upv.es)

This work was supported in part by the Ministerio de Educación, Cultura y Deporte, through the Ayudas para contratos predoctorales de Formación del Profesorado Universitario FPU (Convocatoria 2015) under Grant FPU15/06837, in part by the Programa para la Formación de Personal Investigador de la Universitat Politècnica de València 2014, Subprograma 2, (Codigo del contrato: 884), and in part by the Ministerio de Economía y Competitividad in the Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento within the project under Grant TIN2014-57991-C3-1-P and Grant TIN2017-84802-C2-1-P.

ABSTRACT Multimedia traffic has drastically grown in the last few years. In addition, some of the last paradigms proposed, like the Internet of Things (IoT), adds new types of traffic and applications. Software-defined networks (SDNs) improve the capability of network management. Combined with SDN, artificial intelligence (AI) can provide solutions to network problems based on classification and estimation techniques. In this paper, we propose an artificial intelligence system for detecting and correcting errors in multimedia transmission in a surveillance IoT environment connected through a SDN. The architecture, algorithm, and messages of the SDN are detailed. The AI system design is described, and the test-bed and the data set are explained. The AI module consists of two different parts. The first one is a classifying part, which detects the type of traffic that is sent through the network. The second part is an estimator that informs the SDN controller on which kind of action should be executed to guarantee the quality of service and quality of experience. Results show that with the actions performed by the network, like jitter can be reduced up to 70% of average and losses can be reduced from 9.07% to nearly 1.16%. Moreover, the presented AI module is able to detect critical traffic with 77% accuracy.

INDEX TERMS Artificial inteligence, IoT, multimedia, SDN.

I. INTRODUCTION

In the last few years, video transmission through networks has grown dramatically. Multiple reports state that this trend will increase in the coming years. Cisco, in its Cisco Visual Networking Index, *Forecast and Methodology 2016–2021* report [1], asserts that video traffic in 2021 will be three times that of the 2016 video traffic. Moreover, this traffic will represent 82% of the total Internet traffic in 2021. Furthermore, it says that video surveillance traffic on the Internet will be seven times greater during that same period than the traffic we currently have. In addition, Cisco predicts that 3.4% of the video traffic that will be transmitted though Internet in 2021 will be generated by video surveillance traffic.

The Telecommunication Standardization Sector of ITU (ITU-T) [2], in its recommendation ITU-T Y.4000/Y.2060 (06/2012) [3], defines the Internet of Things (IoT) as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies. The forecasts of IoT devices connected to networks increases day by day. Cisco reported in its White Paper [4] that it expected about 50 billion IoT devices to be connected to the Internet by 2020. Later, the predictions changed. According to Amy Nordrum [5], the predictions for the year 2020 on the amount of connected IoT devices vary from the 30.7 billion, predicted by IHS Markit [6], to the 20.8 billion predicted by Gartner [7] or the 28.1 billion that was predicted by IDC [8]. However, the last two studies do not have in mind smartphones, tablets and computers. Nonetheless, the number of IoT devices that are expected in 2020 is overwhelming. The need to make good traffic management in the network is clear.

Above all, the traffic that requires special treatment, due to its characteristics, is the traffic that comes from critical applications and multimedia.

Networks can have great limitations if they are managed statically, due to the rigidity of these mechanisms. Networks, which are managed statically by commands or static scripts, are less efficient and, furthermore, their resource provisioning is less automatic. In the last few years, the most often proposed technique to improve network management in different studies, is applying Software Defined Networking (SDN). IETF in its RFC 7149 [9] defines SDN as the set of techniques used to facilitate the design, delivery, and operation of network services in a deterministic, dynamic, and scalable manner. Through the SDN Controller, we can apply different rules about traffic flows that traverse the network, which allow us to have adaptive networks. Generally, by applying SDN, we are able to increase the efficiency and reduce the cost of network management, bringing important advantages in the networks where it is applied.

Artificial Intelligence (AI) helps to manage resources and network traffic dynamically. Using AI to study the traffic of the network, we can discover the different types of flow that are being transmitted. Thus, traffic patterns can be obtained, which can then be applied in SDN decision making. Combining AI techniques with SDN, adaptive behaviors are achieved in order to improve the performance of the network.

Due to the predicted change about Internet traffic previously exposed, multimedia traffic must be managed as efficiently as possible. Therefore, and knowing that techniques such as SDN and AI allow to increase efficiency and traffic management dynamically, we propose an intelligent system for guaranteeing QoS and QoE in the video surveillance traffic generated by devices of IoT.

In our study, the treated traffic is generated by multimedia streaming. Our work is directed to the environment of video surveillance, where IoT nodes (smart cameras, among others) send video when anomalies are detected. We propose a SDN core network to manage the different IoT networks. Then, we describe the data-set used to create and train our AI module. After that, the AI system that we have created is detailed. It allows the detection of traffic that is considered critical. In addition, the AI system is also able to estimate the necessary resources to guarantee an adequate level of Quality of Service (QoS) or Quality of Experience (QoE) of the multimedia transmission. This AI system is integrated in the SDN to avoid QoS and QoE problems during the streaming of multimedia traffic. Thereby, it is possible to act when the resources are not enough for having an adequate transmission.

We use an architecture based on SDN, where an OpenFlow switch [10] performs the Cluster Head function, and it is responsible for communicating the IoT networks. The SDN network collaborates with the AI system, which detects critical traffic based on the packets sent by the SDN controller. The statistics collected by the controller from the nodes are sent to the AI system, which estimates the necessary resources for the appropriate transmission. When this point of knowledge of the network and its traffic are reached, the controller is responsible for applying different techniques to guarantee the resources. Depending on the level of resources needed, it will make different actions that will affect different IoT nodes. This provokes an increment of performance in the multimedia streaming and a better network management. Problems such as bandwidth limitation, jitter, delay and packet loss can be reduced or avoided due to the ability of the system to react to the critical situations detected by the AI system. So this system differs from others because, thanks to the results of the classification, it is able to estimate the best action to perform in order to solve the current problem with this critical data. It combines a classification method with an estimator to act and improve the performance of the network.

The rest of this paper is structured as follows. Section 2 presents some of the most relevant works related to our study. Section 3 presents the network architecture, algorithm and messages used to implement our solution. Section 4 displays the test-bed used to train the AI module in the proposed system. Then, in Section 4, the AI module is detailed and its different parts are discussed. In Section 5, the results are shown and the QoS improvements are discussed. Finally, Section 6 concludes the paper and introduces some future works.

II. RELATED WORK

There is a lot of previous work dealing with the study of multimedia transmission, video surveillance, IoT, AI and SDN. Even many of the previous works interact with different technologies. We go on to show some of them.

There are authors who study different areas of IoT applicability. For example, Pal in [11] establishes six areas, called SPACES, which are of interest to companies and organizations when implementing IoT. The areas he presents in his study are: scalability, privacy, affordability, context awareness, ease-of-development, and security. Other authors, such as [12] and [13], propose to use IoT devices and sensors in the field of e-Health or Ambient Assisted Living (AAL). Both papers discuss the possibility of obtaining information through these devices to improve the personalization of medical treatment, facilitate the practice of medicine, and reduce costs. Tan and Wang [14] present the structure of IoT; they also propose IoT architecture and design an application model. Gubbia *et al.* in [15] present a vision of the implementation of IoT in the Cloud. They study key technologies and application domains, to subsequently implement a solution based on both public and private clouds. This implementation is done within a framework that allows Cloud scalability, and that provides capacity for IoT.

Other authors study in the scope of video surveillance. Authors, such as Ajiboye *et al.* in [16], propose a new hierarchical architecture called Fused Video Surveillance Architecture (FVSA). Privately-owned video surveillance systems can increase efficiency in public safety. In their proposal, they define a network adapted to intelligent services of video surveillance, which allows communicating with other

compatible systems in IoT. Lloret *et al.* in [17] present the study of the implementation of a video surveillance system in rural environments. They study codec selection, design and coverage problems, and finally show the results obtained in a public deployment.

There are authors who study multimedia transmission based on SDN technology. In [18], Taha *et al.* present an algorithm for the management of video transmission performance based on SDN. Their algorithm provides a stable video because it distributes bandwidth equally among clients. Besides, they perform tests and present some results that confirm the validity of their algorithm.

There are works done by other authors that relate SDN with IoT. Huang *et al.* in [19] expect that as the SDN networks proliferate, the collection of information and the updating of the network topology and the control of QoS will be facilitated in the IoT environment. Quin et al in [20] extend the Multinetwork Architecture Information Architecture (MINA), achieving different levels of quality for different IoT tasks so that an original SDN IoT controller supports commands to differentiate flows and tasks. They have applied a prototype to an IoT scenario. The performance results indicate that IoT networks can be exploited more efficiently. In [21], Omnes *et al.* present a new multilayer IoT architecture, which includes SDN and NFV, based on network and IT resources. Some authors, such as Bizanis and Kuipers [22], study the state of the art when applying SDN and Network Virtualization (NV) to IoT. They describe the implementation of IoT in both technologies and finally review IoT architectures enabled by SDN-NV together with implementations in real life. Other authors, such as [23] and [24], describe SDN presenting their first fields of application, and analyze the possibility of using it in IoT applications.

There are also authors who propose applying AI to SDN. For example, Matlou and Abu-Mahfouz [25] analyze automatic learning algorithms through AI, which are applied in SDN. They also study the possibility of applying them in Software Defined Wireless Sensor Network (SDWSN). Latah and Toker [26] study the application of AI to SDN. In their conclusions, they indicate that the inclusion of AI in SDN security systems find lower rates in the detection of false positives. Regarding the video transmission, they observe a lower rate of frame losses. They announce that the use of hybrid intelligence will probably improve networks based on SDN. Guibao *et al.* in [27] present a framework called FINE, based on AI. They implement this framework in collaboration with SDN/NFV and demonstrate that its use is feasible in networks and real communication services. Sendra *et al.* in [28] present a proposal to use an intelligent routing protocol in a SDN topology. They designed an intelligent algorithm based on reinforcement learning to improve routing. They used the Quagga suite to implement the routing tasks. The virtual topology is compared with a real one and they announce that their proposal achieves a better RTT but that the convergence time of the OSPF protocol is greater, due to Quagga.

There are some published works where authors have applied AI to IoT. Egea *et al.* in [29] show the combination of Machine Learning with IoT. They propose the modification of the Fast-Fast-Correlation Feature (FCBF) algorithm, with the aim of separating and prioritizing sensed data in multimedia traffic, to avoid damage in emergency situations. Their results show three algorithms based on FCBF, confirming improvements regarding their precision and execution time. Turcu *et al.* in [30] propose the control of traffic signals and environmental parameters by means of a distributed intelligent system, based on IoT.

Our work presents different novelties regarding previous published works. On the one hand, regarding AI, previous techniques have not been used to detect and estimate the parameters on real time. On the other hand, the combination of SDN with IoT and IA is quite novel, and there are few papers published related with them, especially when the aim is to improve QoS in multimedia transmissions. In our proposal, we include an architecture, a message protocol, the required algorithms and an AI system composed by a classifier and an estimator.

III. PROPOSED NETWORK MODEL

In this subsection, the proposal is detailed. First, the architecture is shown. Then, the algorithm is described. Finally, the messages are displayed and the communication process is commented.

A. ARCHITECTURE

In this subsection, the architecture of the network is described. The system is designed for surveillance. The architecture of the proposal is a combination of two network technologies. This combination is shown in Fig. 1. On the one hand, IoT networks work as edge networks. The IoT nodes implement the functionality of the system. On the other hand, a SDN is used to provide the core network. The SDN controller, whose function is to ensure the best QoS, is the central node in the network and it is able to make decisions to interconnect the different IoT networks.

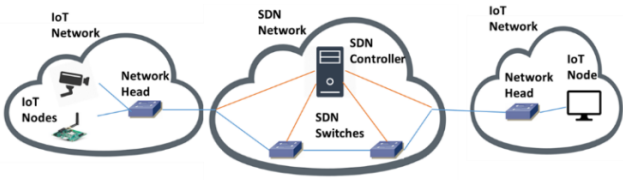

FIGURE 1. Scheme of the architecture proposed.

The two networks are joined by the Network Head (NH) of each IoT network. It is a special node that manages the IoT network communication and sends the data through the SDN network. Moreover, it uses the OpenFlow standard to communicate with the controller and send it statistics about the use of the network. This role is played by an OpenFlow-enable switch. There are also OpenFlow-enabled switches that do not act as the NH of an IoT network. However,

the SDN controller has the AI module and it is in constant communication with it. The AI module is a set of software programs that uses AI techniques to provide the functionalities to the system proposed.

Thereby, the network is composed by IoT networks that implement the functions. There are several different kinds of IoT nodes in the system. Fig. 2 shows the different roles in the system and the communication between them. The tasks that each role implements are painted with different colors. The SDN controller is in charge of network management and sends the statistics that gather from the SDN nodes (NHs and other SDN switches) to the AI module. The AI module uses this set of data in order to apply the AI techniques and inform the controller about the multimedia traffic flowing through the network and its resource requirements. This module is divided into two parts: The traffic classifier, which reports whether the incoming flow is critical or not; and the estimator, which decides the kind of action that should be performed by the SDN controller in order to guarantee the QoS conditions for multimedia transmission.

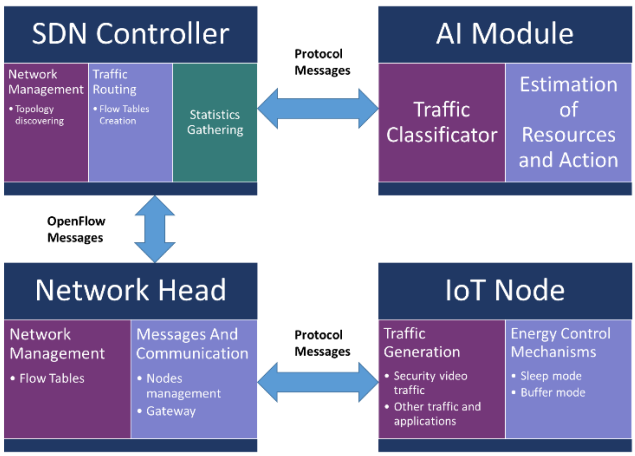

FIGURE 2. Scheme of each actor in the system and their interaction.

The communication between the SDN controller and the AI module is internal, but it is composed by structured messages. The SDN controller performs network management functions. However, the most important parts for this paper are the traffic routing part, where the SDN controller fills the flow table of each switch, and the statistics gathering. In order to do that, the controller communicates with the SDN nodes by using the OpenFlow standard. Nevertheless, in some cases we need to use custom messages defined in the following subsections. The Network Head not only performs network management tasks, but also communicates with the IoT nodes and it is able to manage their behavior. With the messages described in Subsection C, the Network Head is able to operate with the nodes. These nodes are the ones that generate the traffic in the network. Some of this traffic is multimedia traffic, like video surveillance traffic, and its QoS must be guaranteed.

B. ALGORITHM

In this section, the algorithm performed by the SDN controller is detailed, and the different actions that can be executed either by the controller or the nodes are described and classified. The algorithm of the network management is simpler than the one used by the AI module. The controller initializes the AI module. Its main task is to use the standard OpenFlow messages to gather statistics. The algorithm used for network management uses the AI module to detect critical traffic flows being sent through the network. This critical traffic means multimedia traffic in this paper. When multimedia traffic is detected by the AI module, it estimates the resources required and the best action to perform in order to provide an acceptable level of QoE. This estimation uses the statistics provided by the controller to be aware of the current state of the network. With the estimation done by the AI module, the SDN controller chooses an action to perform in the network. Depending on the resources needed to provide enough QoE in the transmission, the AI module labels the level of urgency of the actions to be taken. Therefore, the SDN controller handles the categories of the actions. The SDN controller will perform an action categorized into the same level of needed resources (listed in Table 1) as that of the AI module. With these actions, the SDN controller is able to change priority, queuing policies or making routing decisions in order to guarantee the QoS needed in the transmission. Some problems, such as media access, are managed by the controller. If the action being performed by the SDN belongs to the first three categories, the OpenFlow standard does not contain any messages that could perform it. Therefore, we have designed some custom messages (explained in the next subsection) that implement these actions. The action labeled as IoT has been specifically designed for the architecture maintenance. They allow activating backup nodes in destination and enable buffer mode in source. Moreover, with this ability to put some IoT nodes into sleep mode, the system will avoid QoS problems.

TABLE 1. Possible actions and their category.

Category	Action
Queuing	Change the queuing priority Use QoS queuing
VLAN	Use VLAN PCP for packet priority treating on each switch
BW	Enable an alternative route Enable load balancing Send the traffic through a priority treatment route
BW variation	Enable link aggregation
IoT	Back up nodes in destination Enable buffer mode in source node
NH Congestion	Sleep mode in source IoT network

The algorithm is described in Algorithm 1 and it is graphically shown in Fig. 3, where the Execute_Action subprocess is detailed. The SDN controller starts gathering statistics and providing these statistics to the AI module. When a new

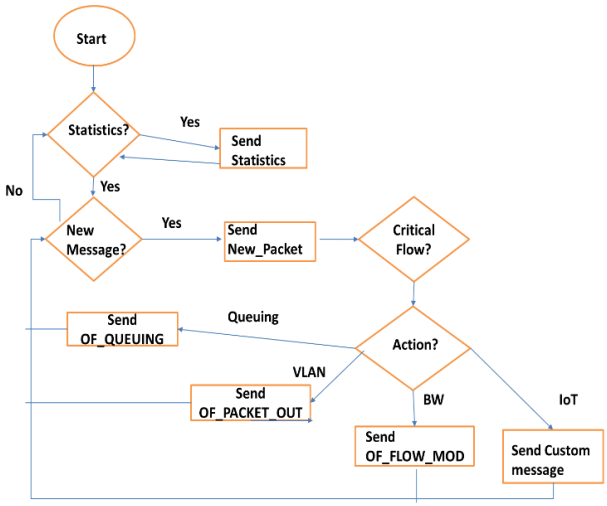

FIGURE 3. SDN controller operation diagram.

packet is reported to the controller, it sends the packet to the AI module. The AI module detects if it is a critical situation. If it is, the AI module also estimates the action to perform. The controller is notified about the action and it sends the messages needed to perform this action in the SDN nodes. The messages needed to be sent change depending on the action to be executed. When the action is done, the controller returns to its usual stats reporting activity.

Algorithm 1 Actions Management

Given: Actions

Initialize_AI(Actions)

Cat_Prev = Cat_initial
Foreach new iteration
    Stats = Get_Statistics()
    AI_Sent_Statistics(Stats)
    If New_Packet **do**
        Cat = AI_Send_Packet(Packet)
        If Cat != No_Crit **do**
            Execute_Action(Cat)
        End If
    End If
End Foreach

C. MESSAGES

The messages sent from the controller to the nodes belong to the OpenFlow standard in almost all cases. In this subsection, the communication processes described are not only those between the SDN controller and the nodes, but also between the SDN controller and the AI module. In addition, the structure of the messages added in order to expand the capabilities of the SDN controller is also detailed.

Since the SDN controller contains the AI module, the messages exchanged between them are not sent through the network. However, it is very important to describe this communication in order to understand how both actors work. The communication process in which the SDN controller receives a new packet is shown in Fig. 4a. The OpenFlow "Packet_In" message is sent from the NH to the controller. The controller sends the packet to the AI module, which classifies the new flow and decides if it is critical or not. Then, if the flow is critical, the AI module estimates the level of criticalness depending on the state of the network. These states have been built up thanks to the messages sent by the SDN controller to the AI module (displayed in Fig. 4b). These messages contain the statistics gathered by the nodes. The AI module reports its classification to the SDN controller. With this info, the SDN controller executes an action that matches with the estimation performed by the AI module.

Depending on the action to be executed, the OpenFlow messages will change. Table 2 shows the actions taken when each message is sent.

TABLE 2. Actions taken when each message is sent.

Action	Messages used
Change the queuing priority	OF_PACKET_OUT (ENQUEUE)
Send the traffic through a route with higher priority.	OF_FLOW_MOD
Enable load balancing	OF_FLOW_MOD
Enable an alternative route	OF_FLOW_MOD
Enable link aggregation	OF_FLOW_MOD
Use VLAN PCP for high priority packets on each switch	OF_PACKET_OUT
Back up nodes in destination	Awake
Enable buffer mode in source node	Buffer_Mode
Sleep mode in source IoT network	Sleep

A special case is when the action is related to the IoT networks. In that case, there are no OpenFlow standard messages to implement the actions, so the messages shown in Fig. 5 are used. They are the following ones:

- Category: Used to communicate with the AI Module and reports which kind of resource has to be improved.
- Sleep: Used to inform the NH that all the networks in the IoT network must be put to sleep when the timer reaches 0 except the video source.
- Awake: Used to inform the NH that all the networks in the IoT network must be awakened when the timer reaches 0. Used also to activate the backup nodes in the destination IoT network.
- Buffer_Mode: Activates the buffer in the video source indicated by its ID.

The communications steps differ when an OpenFlow message is used and when one of the custom messages are used. This happens because the custom messages are focused on IoT network management. The communication process of each case is shown in Fig. 6 and Fig. 7. This process is the continuation of the one described in Fig. 4 (when the AI module indicates that the flow is critical and the action that should be executed). Fig. 6 shows the message exchange when the action is focused on the SDN nodes. The SDN controller

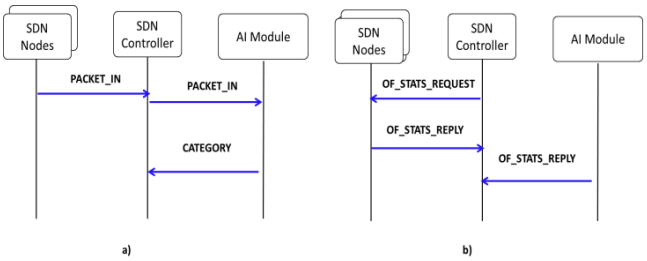

FIGURE 4. Message communication process when: a) A new packet arrives; and, b) Statistics are demanded.

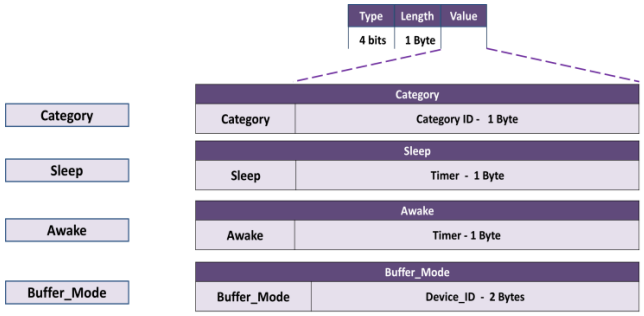

FIGURE 5. Custom messages used when actions in the IoT networks are required.

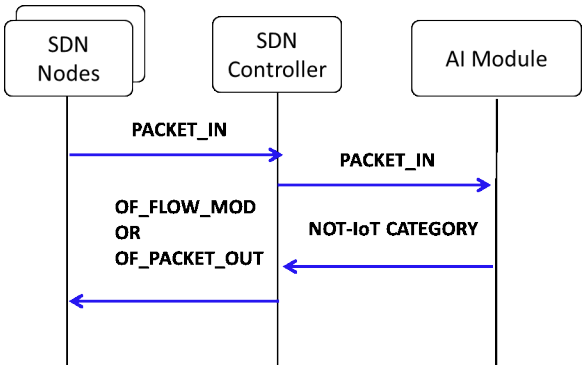

FIGURE 6. Message communication process when there is a SDN-node-based action.

uses the OpenFlow messages to inform the SDN nodes of the network on which action must be performed. Fig. 7 describes the process when the IoT network is involved and the NH is the destination of the messages sent by the controller.

IV. TEST-BED

In this section, the study performed to design the AI system is presented. Different scenarios have been tested. They are based on the architecture previously described. Mininet emulator has been used to perform the simulations and gather the data to design the system. With this test-bed, we are looking for the most complete set of data. So we need to check which characteristics have influence over the QoS obtained, and consequently over the QoE. Therefore, we have analyzed how the video resolution and the frame rate affect some network

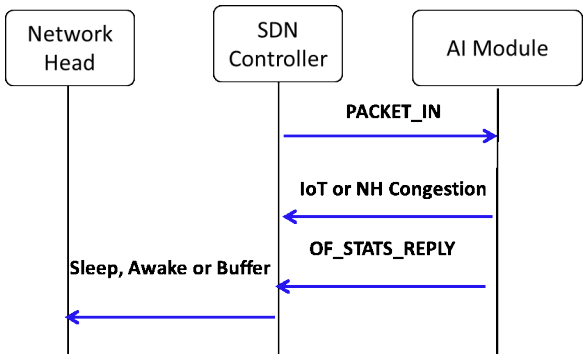

FIGURE 7. Message communication process when the IoT network is affected.

parameters. These network parameters are jitter, delay and loss rate. Moreover, we have also checked how the network status affects the video quality perceived by the end user.

In order to perform the study, we have used several compression formats and different networks. The video formats used have been MJPEG, MPEG4 and H264, with different video resolutions. The video resolutions have been chosen based on the fact that IP surveillance video cameras use sensors from 0.4MP (Megapixels) to 8MP. Some of them reach higher video resolutions like 704 × 576 or UHD. Therefore, we have used three types of resolution for testing: a medium resolution, a high resolution and a very high resolution. These resolutions are 928 × 576, 2592 × 1920 and 3840 × 2160, respectively. The number of frames per second is also taken into account. Rates between 15fps and 30fps are usual in these kinds of cameras. The networks used in the test have been both wired and wireless.

Once the different scenarios and video characteristics have been set, the simulations are executed and the data are gathered. On each simulation, the network parameters (jitter, delay, loss rate and bandwidth) of the links of each node are captured. In the end, the data is collected and the dataset is built. This dataset bring us the possibility of testing our system once it is developed. So, the performance of the AI system can be measured.

A. DEVELOPMENT AND ANALYSIS OF QOS RESULTS

In this subsection, the results obtained from the test-bed to build our dataset are shown and analyzed. In order to analyze the maximum possible number of scenarios for video transmission under the system proposed, we executed several tests that can be classified into:

- Congested networks
- Networks with high loss rate
- Networks with jitter

Each one of these cases is analyzed.

First, we have analyzed how different frame rates (30-60 fps) and resolutions (800 × 600, 1280 × 720 and 1920 × 180) affect to jitter, delay and loss rate. This test has been performed on a wired network with loss rates from 0% to 1%.

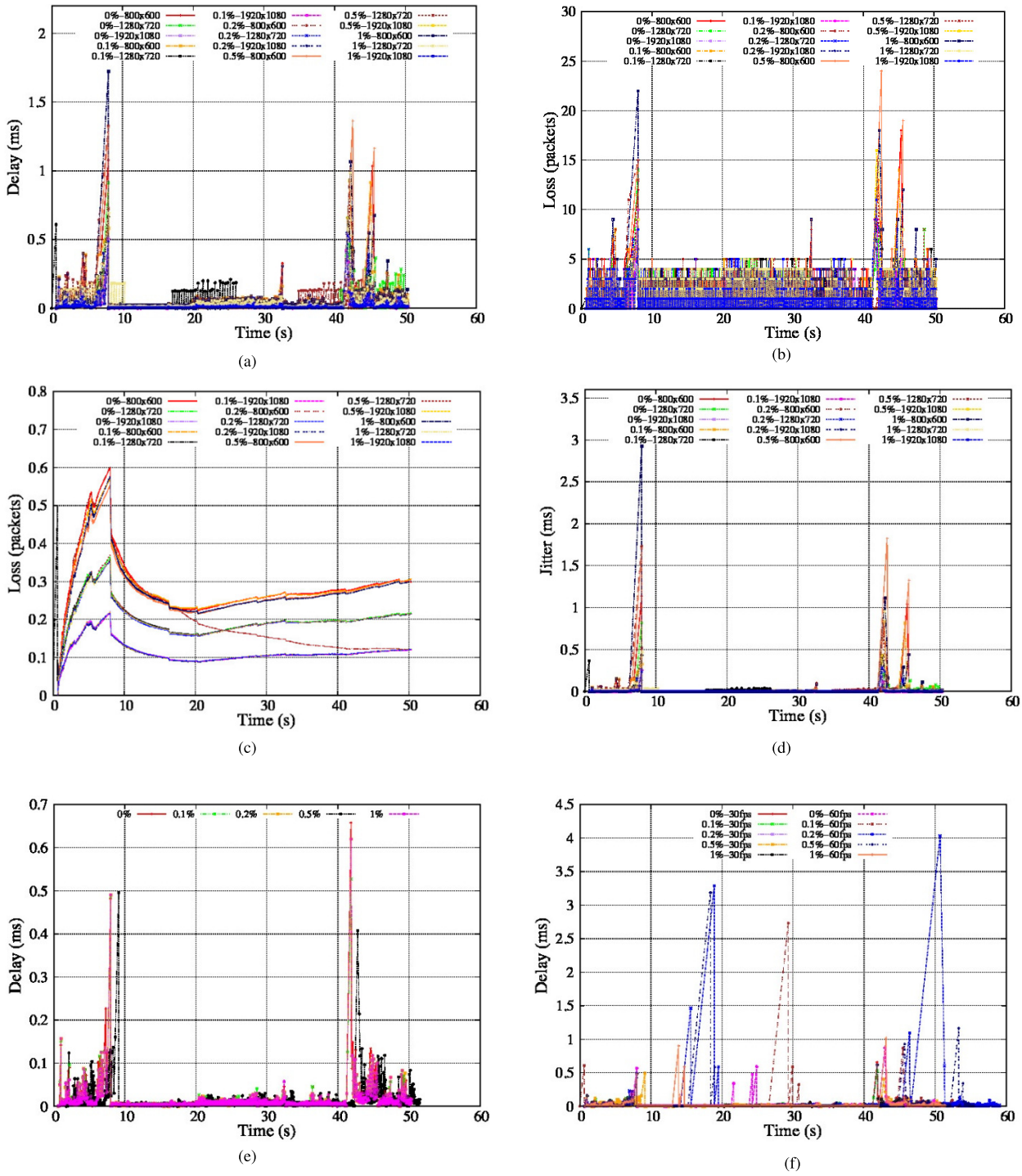

FIGURE 8. QoS parameters depending on loss rate and video resolution. From left to right and up to down: a) delay with loss rate from 0% to 1% and different resolutions with 30fps, b) number of lost packets with the same conditions as in the previous graph, c) average of lost packets, d) Jitter in ms, e) delay with loss rate from 0% to 1% and a 1920 × 1080 30fps video streaming, and f) delay in the same conditions as the previous case but also with 60fps videos.

Fig. 8 shows the results of this test. It shows how loss rate affects the QoS parameters when the video resolution changes. Delay, lost packets, average lost packets and jitter are measured. As displayed in Fig. 8c, the losses obtained are higher when the video resolution is low, reaching more than 20 packets as maximum, as can be observed in Fig. 8b. This pattern is also presented with jitter and delay, shown in Figs. 8a-d. In addition, QoS is degraded at the beginning of the transmission because of the increment of jitter, delay and loss rates. This increment is the highest increment of

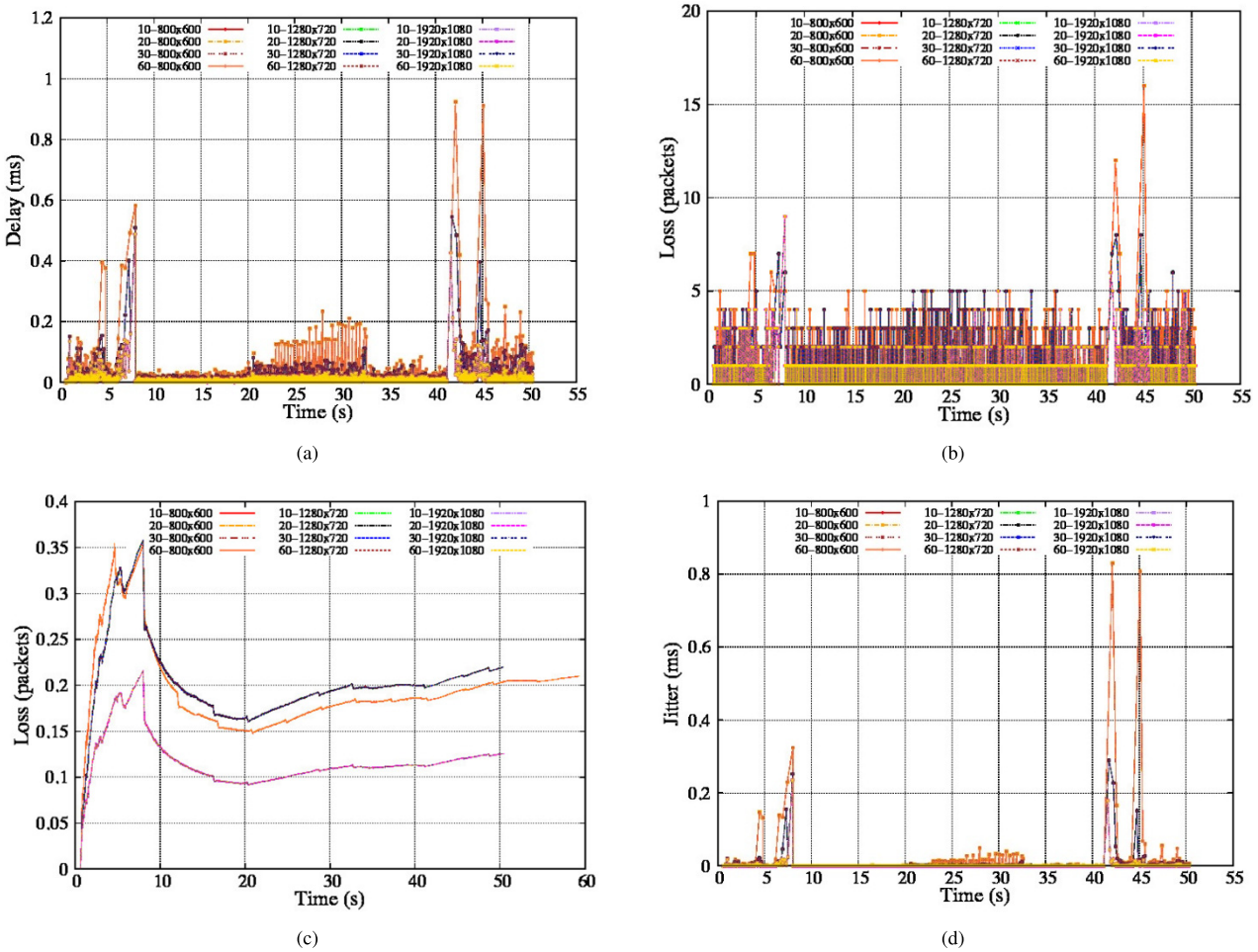

FIGURE 9. QoS parameters depending on jitter, from 10ms to 60ms, and video resolution with 30fps streaming. From left to right and up to down: a) delay, b) number of lost packets, c) average of lost packets, d) jitter.

the transmission. The increments occur at 8s, at 41s and also at 45s, when there are changes on the video scene. The delay increases up to 1.7ms of maximum and, during the rest of the transmission, it varies from 0.2ms to 0.3ms. Then, at the end of the transmission and, as shown in Fig. 8a, it increases again up to 1.4ms. The number of lost packets presents a maximum of 23 packets at the beginning of the transmission and then decreases to 4-5 packets. At the end of the streaming, it raises to 24 packets for an 800 × 600 video resolution and a 0.5% loss rate (Fig. 8b). Regarding jitter, it reaches 3ms at the beginning and 1.8 at the end (Fig. 8d). In Fig. 8e, the manner in which the loss rate affects the delay when a 1920 × 1080 30fps video is being streamed can be observed. The highest changes on delay happen when the video scene changes. Delay increases to 0.5ms at the beginning and to 0.65ms at the end of the streaming. The loss rate presented in this test is not very significant because the different measures with 0% and 1% of loss rate are similar. In order to show how frame rate affects transmission, Fig 8f is presented. The highest the frame rate gives the highest delay. The maximum delay is 4ms at the end of the transmission with 0.2% loss rate. The peaks of delay happen on different timestamps due to the different video framerate.

Secondly, we have analyzed how different values of jitter, from 10ms and 60ms, affect the QoS parameters under streaming with different rates (30-60fps) and resolutions (800 × 600, 1280 × 720 and 1920 × 180). The results are displayed in Fig. 9.

In Fig. 9a, we can observe how the increment of jitter affects the delay on transmission with different resolutions. Again, when the video changes the scene, the increment of delay is higher. It is incrementally increased to 0.6ms at the beginning and 0.9ms at the end of the 800 × 600 video transmission. The higher the resolution, the lesser the delay is. With the same video resolution, the increment of jitter does not affect the delay. However, in Fig. 9b, we can observe that, with higher resolutions, we obtain more lost packets. With 1920 × 1080 video resolution, the maximum number of lost packets is 9 at the beginning of the transmission. Nevertheless, with an 800 × 600 video resolution, we obtain 16 lost packets of maximums at the end. The peaks of loss are placed in the last two changes of video scenes, on 41s and 45s.

In Fig. 9c, we have the same results in terms of average lost packets. We observe the same pattern, with increments at the end of the streaming. In Fig. 9d, we observe that jitter is affected in the same way as delay. We obtain lower values, with maximums of 0.3ms at the beginning of the 800 × 600 transmission and 0.8-0.9ms at the end of it.

Finally, we have analyzed how QoS parameters vary when there are different videos with different rates and resolutions. These videos are sent through networks with different bandwidth from 10Mb/s to 100Mb/s and provoking congestions. In Fig. 10a-b, the delay obtained for different transmissions through a path with 10Mb/s of available bandwidth. In order to congest the network, an 800 × 600 30fps ''disturbing'' video is streamed as displayed in Fig.10a, and a 1600 × 1200 30fps video is streamed in as displayed in Fig. 10b. They started at different moments. This process is repeated after 30 and 40 seconds of transmission. As we can observe in Fig.10, the delay obtained is higher for videos with lower resolutions. For instance, with 800 × 600 streaming, the delay reaches 1.7ms of maximum at the beginning of the transmission, but with 1024 × 768, it is lower than 1.6ms. Moreover, as it can be observed in Fig. 10b, the delay increases up to 1.1ms at the end. We have repeated the process, increasing the available bandwidth to 100Mb/s. In Fig. 10c, we observe that this increment of available bandwidth reduces the impact on delay, being 1.3ms during the beginning of the 1024 × 768 transmission. In Fig. 10d, it is shown how changes in the number of lost packets occur when the 10Mb/s network gets congested. The video transmitted has an 800 × 600 resolution and 30fps. The video is transmitted at 10s(s1), 20s(s2) and 40s(s3). When the 1920 × 1080 video is being streamed, there are losses of 50 packets. In Fig. 10d, we can observe a frame of the video when the peak of lost packets happens. That frame would decrease the QoE obtained. This error happens at 48s. Fig. 10e shows that there is not a relationship between loss rate and the size of the video, but there is a relationship with the video resolution. The transmission, in terms of packet loss, is more affected with less video resolution. This can be observed in the red, green and orange lines in the graph. These lines are the ones related to 800 × 600 video streaming. However, the errors produced when a high-resolution video is being streamed affects the QoE more, as we can observe in the 1920 × 1080 (s2) case. If the network is congested with a 1024 × 768 video, we obtain the results that are presented in Fig. 10f. With these results, when the resolution is low, the transmission is more sensitive to loss rate. This can be observed in the pink, orange and blue lines for 1024 × 768, and these results are compared with the lines related to 1920 × 1080. However, this does not affect the MOS, as we have observed when analyzing the received videos.

B. DEVELOPMENT AND ANALYSIS OF QOE RESULTS

We have performed 495 experiments. On each one, we have saved the video received on the destination. Thereby, we have both, the original video and the video with transmission errors. So, we can perform an objective study of the image quality received. As a result of this study, we have obtained PSNR, NQI, VQM, SSIM and MSE measurements. The goal was to obtain an approximation of the level of quality perceived by the user. Attending to the literature, we have chosen those measurements that have more correlation with the subjective quality, in terms of MOS, perceived by the user.

The Peak Signal to Noise Ratio (PSNR) parameter is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Normally, higher PSNR indicates that the reconstruction is of higher quality [31], [32]. According to the mathematical equations for calculating MSE and PSNR, it can be inferred that they represent similar error values (i.e., the calculated error is of the same degree). Because of this, PSNR can be considered as an unofficial representative of all the above mentioned video quality metrics and still the most widely used metric for video quality estimation in many video processing systems [33]. The Human Visual System (HVS) is highly adapted to extract the structural information from the area of viewing. SSIM metric uses this characteristic of the HVS in the estimation of the quality of the processed digital video. Structural information of an image can be defined by those characteristics that represent the structure of the objects in the scene – independently of the mean brightness and contrast [33], [34]. These measurements are based on three components: luminance comparison, contrast comparison and structure comparison [35]. NQI works in a similar manner as the SSIM index. NQI defines picture distortion as a combination of three factors: difference in mutual characteristics, difference in luminance, and difference in contrast. Human eye sensitivity to spatial-temporal patterns decreases with high spatial and temporal frequency. Based on this difference in sensitivity, high spatial or temporal information can be represented with less data and less precision, while human eyes are more or less insensitive to the loss of this information. This characteristic of HVS is exploited by DCT quantization, which is the base for VQM [31]. The values of VQM start from 0 and, in real situations, they can reach around 12. The VQM value of 0 represents minimum distortion and maximum quality [35]. In conclusion, the SSIM metric has a quite better performance compared to PSNR and, in most cases, performs very similarly to the Human Visual System. But, imperfections are also present. SSIM is almost insensitive to changes in brightness, contrast and hue such that, when these changes are bigger, SSIM values can become largely inverted. VQM mostly considers the changes that are more noticeable to the human eye.

With all the previous discussion, we have chosen VQM, SSIM and PSNR to set the output parameters of the dataset. We have chosen these metrics because every one of them can provide us some kind of characteristic to correlate that metric with the QoE. VQM is similar to the subjective quality perceived by the user. Therefore, this metric has more weight in the settings. The equation used to calculate the subjective

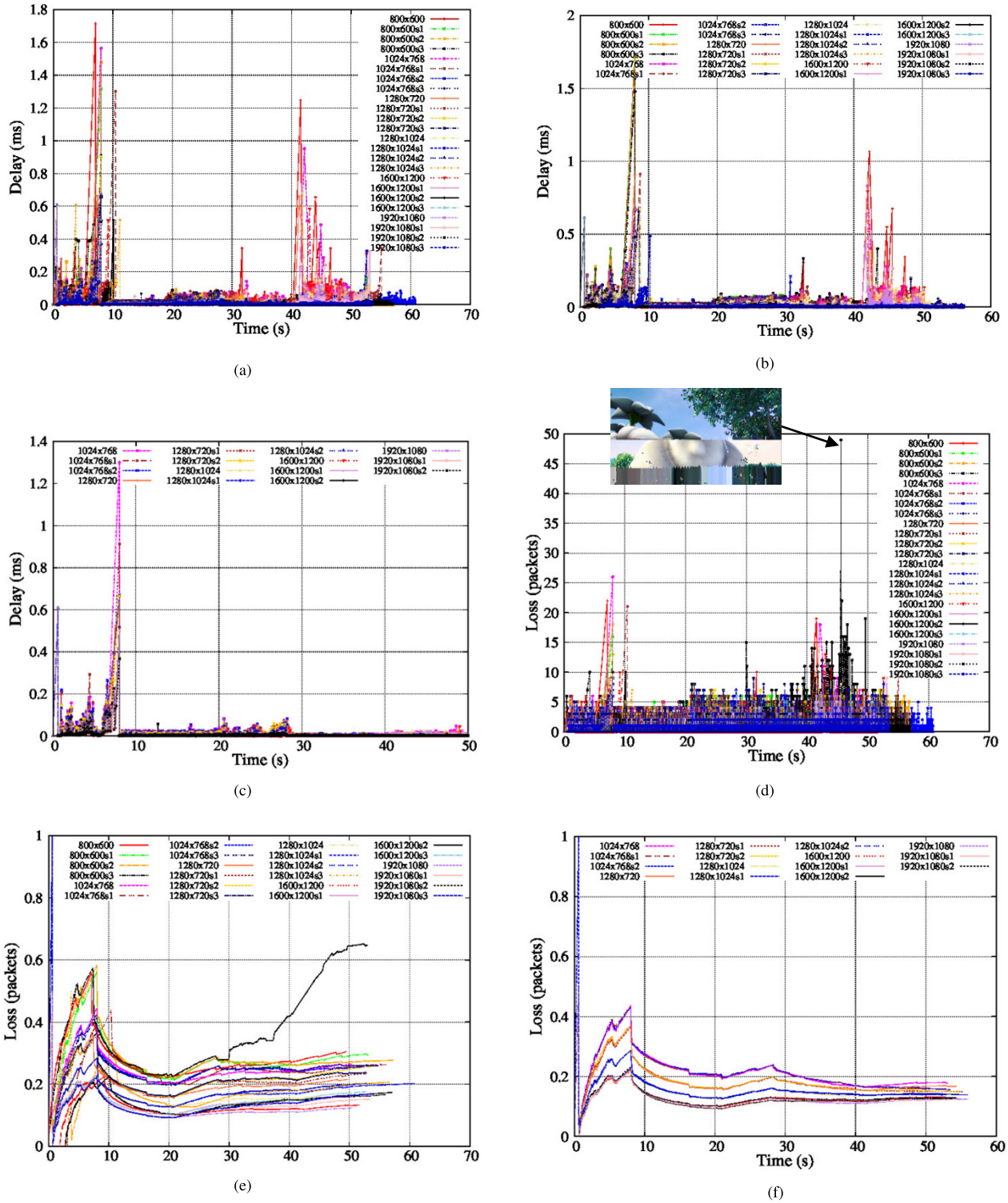

FIGURE 10. QoS parameters depending on network congestion and video resolution. From left to right and up to down: a) delay obtained in a 10Mb/s network with 800 × 600 30fps disturbing traffic, b) delay obtained in a 100Mb/s network with 1920 × 1080 30fps disturbing traffic, c) delay obtained in a 100Mb/s network with 1920 × 1080 30fps disturbing, d) number of packets lost in a 10Mb/s network with 800 × 600 30fps disturbing traffic e) average of lost packets obtained in a 10Mb/s network with 800 × 600 30fps disturbing traffic, f) average of lost packets obtained in a 10Mb/s network with 1920 × 1080 30fps disturbing traffic.

values from objective measurements of each frame is:

$$QoE_{\approx s_i} = \delta_{i,VQM} R_{i,VQM} + \delta_{i,SSIM} R_{i,SSIM} + \delta_{i,PSNR} R_{i,PSNR} \quad (1)$$

Where the subjective approximate QoE value for each frame *i*, *QoE_aprox_s_i*, is the combination of three factors. The result of each metric *M* for the frame *i (R_(i,M))* adjusted with a specific weight ∂_(i,M). These values correspond with

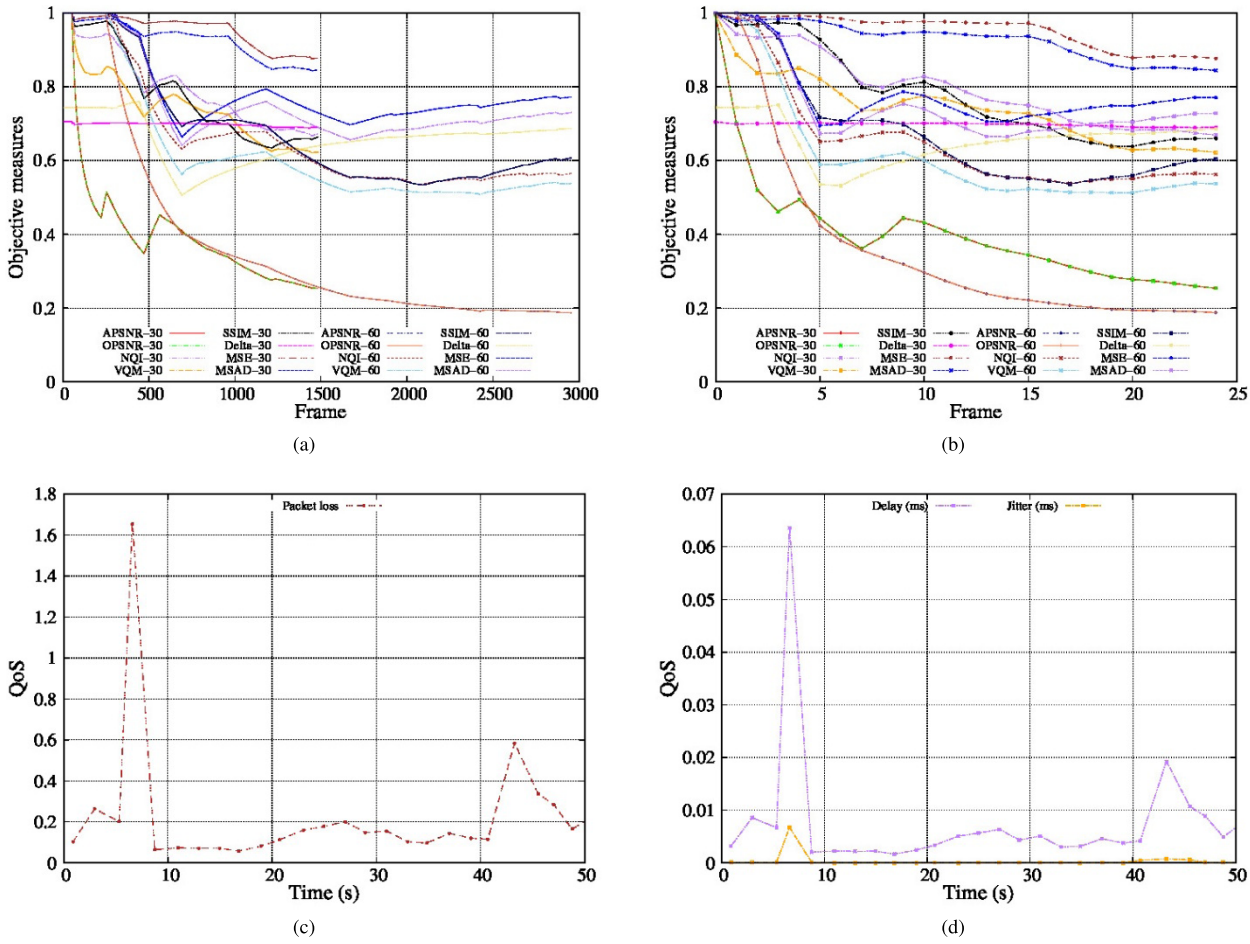

FIGURE 11. Preprocessing done with the data obtained from the QoS and QoE analysis. From left to right and up to down: a) dataset related to the subjective measurements after analyzing both, a 30fps and a 60fps video streaming., b) preprocessing base on sampling for each frame, c) preprocessing based on sampling the lost packets each time instant, d) preprocessing based on sampling jitter and delay each time instant.

∂_(i,VQM)=0.5, ∂_(i,SSIM)=0.35 and ∂_(i,PSNR)=0.15, which are determined based on the literature.

C. DATA PREPROCESING

In order to set the data used in the traffic classifying model, we first performed data preprocessing. Thereby, we grouped the data in GOPs of 2 seconds. That means that if the obtained data was frame-based, we would preprocess them each 60 frames when the frame rate is 30fps, or each 120 frames when it is 60fps. Results are presented in Fig. 11.

As we can observe in Fig.11a, we converted the approximated 3000frames for 60fps videos and 1500 frames for 30fps videos into 25 samples as shown in Fig. 11b. Each sample was formed by calculating the average of the values within the 2-second interval. The same process has been executed for the data packets. In that case, they are presented depending on the time instant, not frames. As it can be observed in Fig. 11c and Fig.11d, the average data was taken every 2 seconds in order to obtain 25 samples from 50 seconds of transmission. Each sample corresponded to 2 seconds of video. In those 2 seconds, we obtained the average of different QoS parameters like jitter and delay (Fig.11d) or number of lost packets (Fig. 11c). Regarding lost packets, there was a peak with more than 1.6 packets at the beginning of the transmission and another with 0.6 at the end. However, during the transmission, the average was below 0.2 packets. As in the previous study, these peaks happen when the video changes the scene. In terms of delay, at 8s, it reached 0.06ms and, at 42s, it reached 0.02ms. The average during the transmission was below 0.01ms. The jitter suffered an increment of 0.01ms at 8s, but it was insignificant during the rest of the transmission.

Once the preprocessing was done, we obtained 25 samples for each experiment. Jitter, delay, lost packets, bandwidth, resolution and fps were measured. The average value of the different objective metrics for each sample and time interval was used for labeling. Consequently, we obtained 6 QoS-or-video characteristics for each 2 seconds as an input for our system. As output, we obtained the label that corresponded to the average of the objective QoE metrics. The last step in the preprocessing process was to divide the spectrum of the possible objective QoE values into 5 ranks. Thereby, the labels

used were discrete values, not continuous. Managing only 5 possible labels is much easier. We associate the kind of traffic to those labels as we describe in the next subsection.

V. ARTIFICIAL INTELLIGENCE SYSTEM

The artificial intelligence system is composed of two main processes. The first process is a multimedia traffic classifier and the second is an estimator of the network resources. This last process is used for guaranteeing the most adequate network condition for multimedia transmission. For the development of the classifier, we have used convolutional and recurrent neural networks. A statistical model has been used to design the estimation process. In the following subsections, both processes are detailed.

A. QOE-BASED MULTIMEDIA TRAFFIC CLASSIFICATION

We performed 237 experiments and obtained 25 samples for each one. Therefore, a total amount of 5,925 samples have been used to develop the classifying model. First, a learning process is carried out and, then, a test process to check the model. The learning process is a supervised process because the labeling process has been manually executed for each characteristic array. There are 5 different types of labels that represent the kind of traffic in the transmission:

- Non-critical traffic
- Little critical traffic
- Rather critical traffic
- Critical traffic
- Very critical traffic

As it is mentioned in the previous subsection, these values are obtained from the objective QoE. They cover a Rank from 1 to 5, with 1 being a really bad objective QoE and 5 being the best possible. Thereby, a score of 5 corresponds to non-critical traffic and a score of 1 to very critical traffic. Consequently, an output of 2 means critical traffic, 3 means rather critical traffic, and 4 means little critical traffic. Therefore, the classifier model has QoS values and video characteristics as an input and discrete objective QoE value as an output. We must take into account that some video errors like black pixels, color errors, tiling, noise, ghosting, soft focus, and flickering can affect the QoE. Errors like the ones aforementioned can affect the quality of the video received by the surveillance video system and reduce its efficiency. This could cause an incident to go undetected by the system, and it would not accomplish its function. For this reason, it is very important for our classifier system to rapidly and efficiently detect the critical traffic, from scores of 2 to scores of 5, so that the system can provide a solution as soon as possible using the estimator.

When the samples are labeled and preprocessed, the learning process can start. We used 80% of the samples for learning and 20% for testing. The group selection, learning and testing, were randomly chosen from the sample set. In the learning process, the stopping criteria used has been based on the mean square error and the number of cycles.

We have analyzed several learning methods in order to determine which one presents the best performance and adapts better to the problem. The methods used are based on neural networks, array processing machines, statistics and k neighbors. The statistic method (Kernel) is based on discriminant analysis [36]. This is a term that is broadly used to include problems associated with the (statistical) separation between distinct classes or groups. It includes a wide range of problems in statistical pattern recognition, where a pattern is considered as a single entity and is represented by a finite dimensional vector of features of the pattern. The neural network (NN) method is based on adaptive networks. A class of adaptive networks is identified which makes the interpolation scheme explicit. This class has the property that learning is equivalent to the solution of a set of linear equations. These networks thus represent nonlinear relationships while having a guaranteed learning rule. Another learning machine used is support vector machine (SVM). The machine conceptually implements the following idea: input vectors are nonlinearly mapped to a very high dimension feature space. In this feature space, a linear decision surface is constructed. Special properties of the decision surface ensure high generalization ability of the learning machine. Finally, we have used classifying methods based on distances – specifically on the closest neighbor (KNN).

Since the optimal model is found, the accuracy of the system is measured. In order to calculate it, we have used, for all the items discussed previously, Precision (P), Recall (R), Accuracy (A) and F1. All these parameters used the following results to carry out the classifier model:

TP – The model classifies the traffic as critical or very critical and its success.

TN – The model classifies the traffic as non-critical and its success.

FP – The model classifies the traffic as critical or very critical and its failures.

FN – The model classifies the traffic as non-critical and it failures.

For obtaining the accuracy measurements, we calculate:

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{5}$$

In Fig. 12, the results obtained from the measurements previously explained are presented. This provides us a way to analyze the accuracy of our system.

As it can be observed in Fig.12, the SVM model is the one that presents the best results. Its accuracy is 84%. That measurement is based on the total set of analysis. However, other measures are even more important to our system because we should avoid the false negatives. That means that

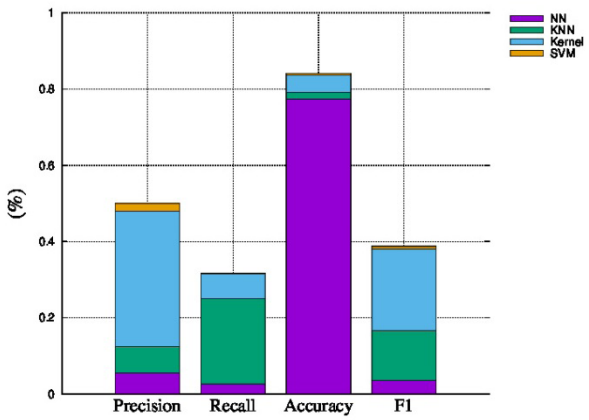

FIGURE 12. Precision measurements for model checking.

the system should not classify critical traffic as non-critical traffic. The measure that takes more into account the false negatives is recall. In this case, the statistic model and the SVM model present similar results, with 31%. KNN presents 25% of recall and the neuronal network model is far away from obtaining such a high result, with only 2%. For the rest of the measurements, both SVM and Kernel present similar results. For SVM, we obtained 50% precision and 38% of F1. For Kernel, we obtained 48% precision and 38% of F1. The worst results were obtained with NN: 5% precision and 3% of F1. However, precision is not as important as recall for our system because it takes into account the false positives, which are not as important as the false negatives. F1 takes into account both, the FN and the FP, becoming an illustrative measurement; but it is not as important as recall. In order to illustrate how each one of the classifier methods works, Fig.13 is presented.

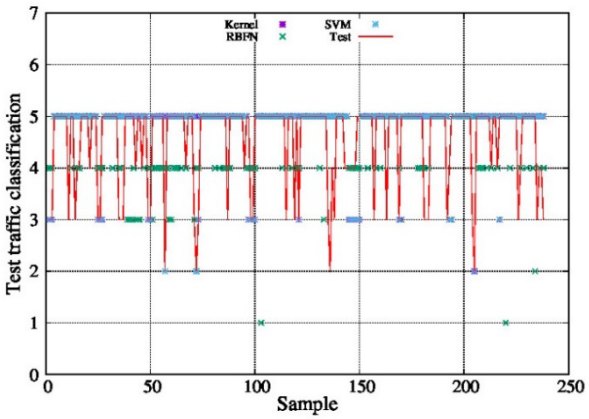

FIGURE 13. Test results for the evaluated classifier methods.

Fig. 13 shows that the SVM model is the one that performs better in the test, detecting 75% of critical traffic. The NN model does not perform very well, obtaining 3 FP, which is important in our system. Moreover, it has not detected any critical traffic. The distribution of the dataset has probably affected this method because there is not the same quantity of critical traffic as non-critical traffic.

In conclusion, the SVM model is the one that best fits the problem. Therefore, in order to improve its performance, some parameters are refined. They are: activating function, weighting factor C, nu and ε. C determines the compensation between the error in training and the VC dimension of the model [37]. Modifying ε, we are able to vary the quantity of support vectors which affects the smoothness of the SVM output. The complexity and the generalizing capability of the network depend on that value [38]. The *nu* parameter allows us to control efficiently the number of support vectors.

Table 3 shows the classifier results after we refined the SVM model. With these data, we can statistically compare the new results with the ones obtained in the previous study.

B. NETWORK RESOURCES ESTIMATION FOR QOE GUARANTEEING

The resource estimation model is based on the Bayes statistic model. In this model, the most adequate resource for solving the critical problem is estimated from the network status and the kind of node. The possible resources are:

- Traffic priority (queuing management/theory)
- Route traffic
- BW variation
- Buffer management
- Sleeping nodes in the source network
- Activate backup nodes

The statistic model has been obtained from the previous study. Situations from the test-bed, where the traffic is critical, have been chosen. Then, all the resources are tested to check which one improves more efficiently the traffic conditions and the results are obtained. Some cases, where the traffic is not critical, are also studied to have some scenarios with false positives. Thereby, the system learns in which cases it should do nothing. The cases studied are the following ones:

1) Congested networks due to the increment of video flows either with a lot of bandwidth or with little bandwidth
2) Congested networks due to the increment of data traffic either with a lot of bandwidth or with little bandwidth
3) Wired or wireless networks with high loss rate
4) Wired or wireless networks with high jitter

The data is preprocessed as in the classifier model and the result is the input of the system. This input is received for each node in the network so that we can identify if the critical traffic is being generated in the access network (IoT network) or in the core network (SDN network). Once all the cases are studied, the estimation model – given the network parameters as an input – provides which resource is the most adequate to be developed. Mathematically, the probability that one event occurs – given some input parameters – is defined by the Bayes statistic estimator:

$$\Pr(r|j,d,p,b,n) = \frac{\Pr(r)\Pr(j,d,p,b,n|r)}{\Pr(j,d,p,b,n)} \quad (6)$$

TABLE 3. Learning methods comparison.

Method	Classification Error	stddev Global Classification Error	Correctly classified (Accuracy)	Recall	F1	Precision
NN	0.3792	0.1290	0.6207	0.0263	0.0357	0.0556
SVM	0.1590	0.0769	0.8410	0.3158	0.3871	0.5000
KNN	0.2083	0.7708	0.7917	0.2500	0.1667	0.1250
Kernel	0.1632	0.0749	0.8368	0.3158	0.3810	0.4800
New-SVM	**0.0865**	**0.0119**	**0.9135**	**0.5382**	**0.6118**	**0.7745**

where *Pr(j,d,p,b,n|r)* is the maximum likelihood function. It estimates the probability that a resource r is chosen. That probability is calculated, given some network parameters *j, d, p, b* (jitter, delay, lost packets and delay), and if the node is NH or not (n). As the denominator does not depend on the numerator, the resource estimation problem can be presented as:

$$\Pr(r) = \Pr(r)\Pr(j, d, p, b, n|r) \tag{7}$$

where r is one of the previously described resources and

$$\Pr(r) \approx \Pr(r)\Pr(j, d, p, b, n|r) \tag{8}$$

We can assume that the variables are independent in order to estimate the probability *Pr(j,d,p,b,n|r)*. Describing $p(x)$ as the model parameters, we can define the probability as in (9):

$$\Pr(j, d, p, b, n|r) \approx p(j|r)p(j|r)p(p|r)p(b|r)p(n|r) \tag{9}$$

Depending on the quantity of available data for learning, we can make another supposition. For example, we could estimate, at the same time, several variables like *j, d* and *p*. Then we obtain:

$$\Pr(j, d, p, b, n|r) \approx p(j, d, p|r)p(b|r)p(n|r) \tag{10}$$

In this way, given the network status, with the SDN node where it is being measured and the bandwidth, we can calculate the conditional probability of use of the r_x resource. According to Equation 10, *b* and *n* are independently estimated and *j,d,* and *p* are jointly taken into account.

Some network situations have been simulated, and the percentage of times that some resource has been chosen has been calculated in order to validate the model. In Fig. 14, the percentage of times that some resource has been chosen in each one of the following situations is shown:

1) Network with low BW congested during video streaming
2) Network with high BW congested during video streaming
3) Network with low BW congested during data transmission
4) Network with high BW congested during data transmission
5) High loss wired network
6) High loss wireless network
7) High jitter wired network
8) High jitter wireless network
9) Source IoT network congested
10) Destination IoT network congested

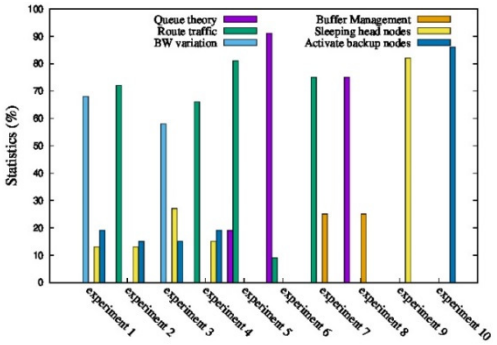

FIGURE 14. Statistical results of the different experiments.

Fig. 14 indicates that, when the SDN is congested during video streaming, the system chooses to increase the available bandwidth, with 58% of probability. However, if the available bandwidth is not low, the best option is to route the traffic, with 72% of probability. Results are similar with data packets, with 66% of probability. If the congestion is originated in the source network, the system chooses to sleep the nodes in that network. Thereby, the maximum bandwidth is assigned to the video generated by the surveillance system. This happens in 86% of the cases. The action most often chosen for the case in which the congestion happens in the IoT destination network is to activate the backup nodes in that same network, with 91% of probability. Regarding losses, in wired networks, the system chooses to route the traffic through another route with 81% of probability. If the network is wireless, the solution is to use priority with queuing (91% of probability). In 25% of the times, the system activates the buffer mode when the problem is the jitter. Thereby, the congestion in the nodes side is reduced. In wireless networks, the system manages the node to operate the multimedia traffic with more priority. The buffer is not being used to reduce energy consumption of the IoT nodes. So, other solutions have a greater probability (75%) of being chosen in those situations.

Once the experiments have been finished, the model has been training based on Equation 10. The results of the test are shown in Table 4. In the table, the estimation error and the accuracy of the system can be observed. Although the values

TABLE 4. Estimation model results.

	Estimation Error	Correctly Estimated (Accuracy)
Estimation Model	0.3066	0.6934

obtained are acceptable, further studies can be carried out to improve the performance.

VI. METHODOLOGY AND RESULTS

Once the AI system is trained and explained, its application to the SDN must be measured. In this section, the experiments performed to measure the improvement during multimedia transmission are detailed. First, the topology and the software used in the experiments are described. Then, the results are shown and discussed.

A. METHODOLOGY AND TOPOLOGY

The experiments ran over the emulator Mininet. This emulator provides SDN emulation by using Linux Hosts as PC and Switches. The experiments consist of sending video streaming through the core network in different scenarios. The streaming is performed using the VLC software. Both the source and the destination network are emulated as Linux hosts in Mininet; so we manage them as hosts in the network. Mininet allow us to modify the network conditions and, along with the multimedia traffic sent, to simulate different scenarios. These conditions are defined in a script and the network is built with those characteristics. The topology used in the experiments is the one shown in Fig. 15. This topology allows us to use different techniques, such as alternative routing, thanks to the path redundancy. The streaming source is H1, and the destination, H2. The path chosen for the delivery is S1-S3-S5 because the others present higher delay. Moreover, the link between S1 and S3 is marked in Fig. 15. This is because we are able to use link aggregation in that link in order to increase the available bandwidth.

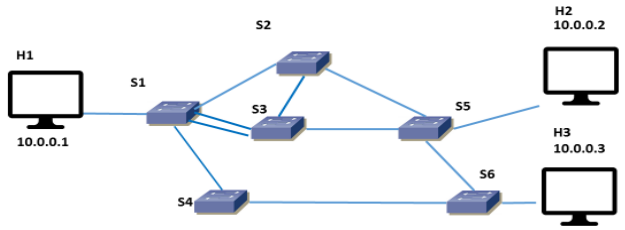

FIGURE 15. Topology used in the experiments.

Attending to Fig.14, the different scenarios tested are the following ones: In the first scenario (Scenario 1), there is a bandwidth problem and the action indicated by the AI module is to use an alternative path, with an increment of delay. In the second one, the problem is the same, but the action to perform is to use link aggregation in order to increase the bandwidth. In Scenario 3, the AI module suggests using queuing to modify the priority and reduce the loss rate. Finally, in Scenario 5, there is congestion in the source IoT network and the AI module reports the custom message that is necessary to put the nodes to sleep. In the next subsection, the results of each scenario are presented.

B. RESULTS

The results of the experiments are displayed and discussed in this subsection. Not only are QoS parameters measured, but also QoE has been analyzed, and it is discussed at the end of the subsection.

In Fig. 16, the comparison between the bandwidth used by the multimedia streaming in both cases, with and without the proposed system, is displayed. Without any system that performs actions to improve the QoS, the bandwidth has a maximum of 1.83Mb/s. However, with the proposal, that maximum increases up to 3.08Mb/s. The minimums are 16.4kb/s and 104.12kb/s, respectively. The average bandwidths are similar, 1.19Mb/s and 1.08Mb/s.

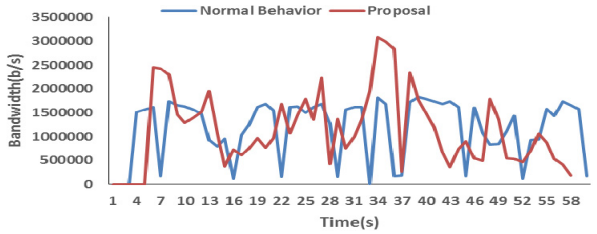

FIGURE 16. Bandwidth obtained in Scenario 1.

In terms of jitter, the performance is compared in Fig. 17. The average jitter without the proposed system is 2.47ms. However, by using the alternative path, the average is 8.19ms. The maximums are 11.13ms and 47.62ms, respectively. Finally, the minimum jitter also increases from 0.01ms to 0.13ms with the proposed action.

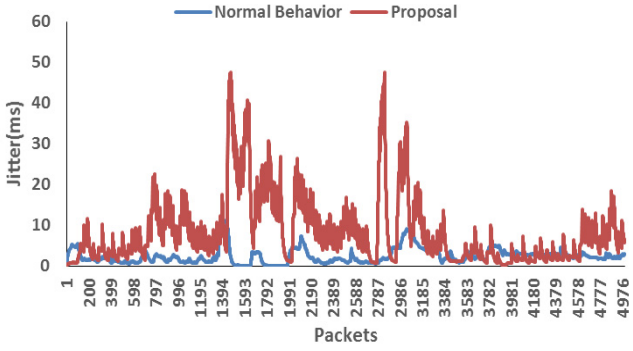

FIGURE 17. Jitter obtained in Scenario 1.

In Scenario 2, due to the network status, the action to perform is link aggregation. Fig. 18 shows the changes produced in terms of bandwidth when the system is used. Without it, the average bandwidth is 1.12Mb/s. There was a maximum

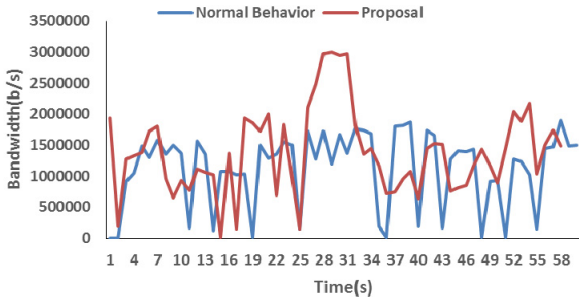

FIGURE 18. Bandwidth obtained in Scenario 2.

of 1.89Mb/s and a minimum of 10.9kb/s. When the system is used, the average bandwidth is 1.42Mb/s, the maximum is 2.99Mb/s, and the minimum is 16.4kb/s.

Regarding the jitter, there is a reduction when the system performs the action indicated by the AI module, as shown in Fig. 19. The average jitter is reduced from 2.4ms to 0.49ms. The maximums are similar: 8.89ms when the proposal is not being used, and 7.63ms when it is. The minimums are 0.22ms and 0.03ms, respectively.

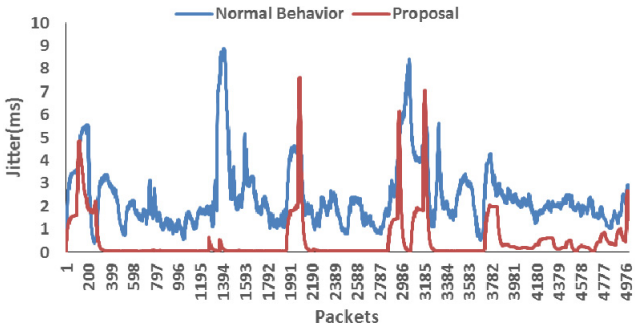

FIGURE 19. Jitter obtained in Scenario 2.

In Scenario 3, the loss rate is measured. The AI module decides to use priority techniques using queues. Fig. 20 shows that the bandwidth does not change like in the previous scenarios. The average bandwidths are 1.01Mb/s, without applying the action, and 1.07Mb/s when applying it. The maximum is 1.95Mb/s in both cases and the minimum is 16.4kb/s.

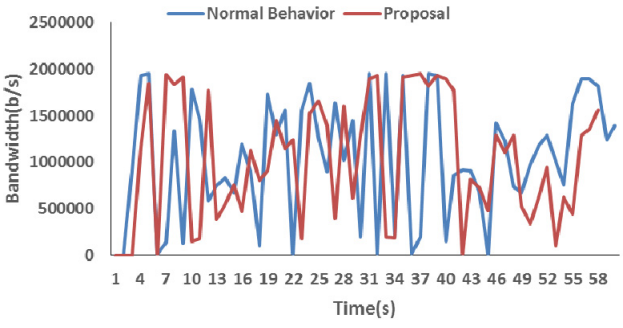

FIGURE 20. Bandwidth obtained in Scenario 3.

However, the jitter presents some differences. Fig. 21 shows that the average jitter without the proposal is 4.77ms, and it is reduced to 1.35ms when it is used. The average jitter is quite different: 10.2ms for the proposal, and 43.05ms when it is not used. The minimums are 0.01ms with the proposal and 0.13ms without it.

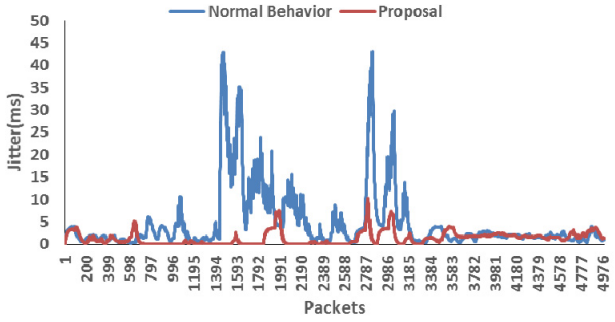

FIGURE 21. Jitter obtained in Scenario 3.

Fig. 22 shows the reduction of the loss rate from 9.07% to nearly 1.16%. This loss rate reduction works to improve the QoE, as is shown at the end of the section.

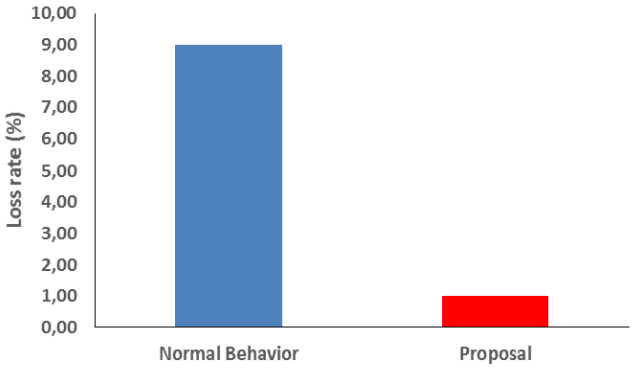

FIGURE 22. Loss rate obtained in Scenario 3.

The last scenario is used to test the custom characteristic for QoS improvement using IoT characteristics. The congestion in the source network is handled by putting the rest of the nodes to sleep. So, the QoS of the multimedia transmission is improved, as shown in Fig. 23. It shows that the maximum bandwidth consumed by the multimedia flow is increased from 1.79Mb/s to 2.92Mb/s by using the proposed solution. The average bandwidth is also increased from 0.85Mb/s to 1.24Mb/s. The minimum bandwidth consumed is 8.98kb/s with the system and 104.1kb/s without it.

The jitter also changes with this action. It is displayed in Fig. 24. The congestion in the source network produces an average jitter of 9.93ms. There is a maximum of 48.08ms and a minimum of 0.12ms. Nevertheless, with the action performed, this jitter is reduced to 0.27ms of the average, 6.19ms of the maximum and 0.01 of the minimum.

After testing these scenarios, the video obtained in the destination network is watched by 11 users, 8 males and

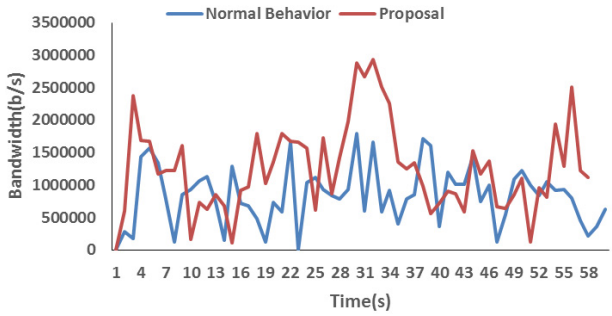

FIGURE 23. Bandwidth obtained in Scenario 4.

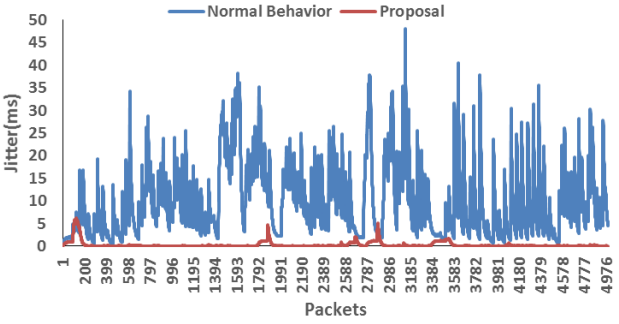

FIGURE 24. Jitter obtained in Scenario 4.

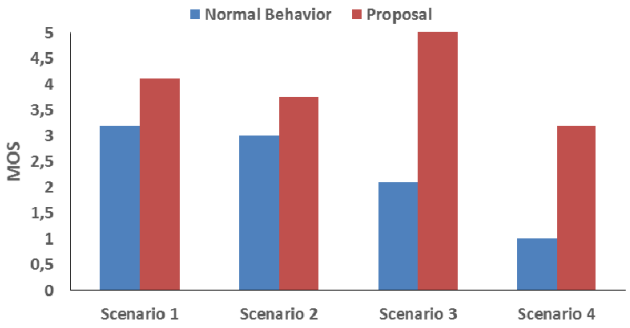

FIGURE 25. QoE obtained in each Scenario.

3 females. They chose for each player a score from 1 to 5 – 5 being the best quality and 1 being the worst one. The results are shown in Fig. 25. All the actions performed by the controller increase the MOS obtained by the users. On the one hand, the greatest improvements are those produced in Scenario 3, where the MOS increases from 2.1 to 5, and in Scenario 4, where it increases from 1 to 3.2. On the other hand, in Scenario 2, the increment is from 3 to 3.75 and, in Scenario 2, the MOS increases from 3.2 to 4.1.

VII. CONCLUSION

We have proposed an artificial intelligence system to detect problems and correct errors in multimedia transmission in surveillance IoT environments connected through a SDN. The system performs some actions to guarantee the Quality of Service (QoS) and Quality of Experience (QoE). The system has been tested in several scenarios.

With the proposed system, the QoS can be improved in different cases when the network suffers problems like congestion or high loss rates. Some QoS parameters are improved in the test performed, like bandwidth and jitter, and then, the QoE increases. Moreover, the presented AI module is able to detect critical traffic with 77% accuracy. This is the main model limitation, due to the classification method. Improving this classification method, by using a more complete data-set would allow us to improve this accuracy.

As future work, we can improve the system accuracy by using the end users' interaction. So, during the transmission, if the QoE experienced by the user is not satisfactory, they can interact with the software from the destination. This interaction can be implemented through a checkbox or through some command. This would be detected by the system and marked as a FN. Another possible improvement would be to enhance the estimation model to a node-level one. This would allow us to select the best resource or action (depending on the network status) for each link in the path, not only for the entire network.

Moreover, in future works, we will analyze the correlation between the objective QoE metrics and MOS or DMOS. Thereby, this study could be applied to future research in order to improve the performance. Furthermore, some other statistical methods will be studied in order to improve the results in the estimation process for network resources selection.

REFERENCES

[1] *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*. Accessed: Dec. 27, 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[2] *ITU Telecommunication Standardization Sector*. Accessed: Dec. 27, 2017. [Online]. Available: https://www.itu.int/en/ITU-T/Pages/default.aspx

[3] *ITU-T Recommendations*. Accessed: Dec. 27, 2017. [Online]. Available: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060

[4] *The Internet of Things. How the Next Evolution of the Internet Is Changing Everything*. Accessed: Dec. 27, 2017. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[5] *Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated*. Accessed: Dec. 27, 2017. [Online]. Available: https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

[6] *Tech Companies Creating Strategic Platforms to Support the Internet of Things, IHS Says*. Accessed: Dec. 27, 2017. [Online]. Available: http://news.ihsmarkit.com/press-release/technology/tech-companies-creating-strategic-platforms-support-internet-things-ihs-say

[7] *Gartner Says 6.4 Billion Connected 'Things' Will Be in Use in 2016, Up 30 Percent From 2015*. Accessed: Dec. 27, 2017. [Online]. Available: https://www.gartner.com/newsroom/id/3165317

[8] *Worldwide and Regional Internet of Things (IoT) 2014–2020 Forecast: A Virtuous Circle of Proven Value and Demand*. Accessed: Dec. 27, 2017. [Online]. Available: https://www.business.att.com/content/article/IoT-worldwide_regional_2014-2020-forecast.pdf

[9] *Software-Defined Networking: A Perspective From Within a Service Provider Environment*. Accessed: Dec. 27, 2017. [Online]. Available: https://tools.ietf.org/html/rfc7149

[10] *Open Datapath*. Accessed: Dec. 27, 2017. [Online]. Available: https://www.opennetworking.org/projects/open-datapath/

[11] A. Pal, "Internet of Things: Making the hype a reality," *IT Prof.*, vol. 17, no. 3, pp. 2–4, May/Jun. 2015. [Online]. Available: https://doi.org/10.1109/MITP.2015.36

[12] M. Hassanalieragh *et al.*, "Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges," in *Proc. IEEE Int. Conf. Services Comput.*, New York, NY, USA, Jun./Jul. 2015, pp. 285–292. [Online]. Available: https://doi.org/10.1109/SCC.2015.47

[13] L. Parra, S. Sendra, J. M. Jiménez, and J. Lloret, "Multimedia sensors embedded in smartphones for ambient assisted living and e-health," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13271–13297, Nov. 2016. [Online]. Available: https://doi.org/10.1007/s11042-015-2745-8

[14] L. Tan and N. Wang, "Future Internet: The Internet of Things," in *Proc. 3rd Int. Conf. Adv. Comput. Theory Eng. (ICACTE)*, Chengdu, China, Aug. 2010, pp. V5-376–V5-380. [Online]. Available: https://doi.org/10.1109/ICACTE.2010.5579543

[15] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013. [Online]. Available: https://doi.org/10.1016/j.future.2013.01.010

[16] S. O. Ajiboye, P. Birch, C. Chatwin, and R. Young, "Hierarchical video surveillance architecture: A chassis for video big data analytics and exploration," *Proc. SPIE*, vol. 9407, p. 94070K, Mar. 2015. [Online]. Available: https://doi.org/10.1117/12.2083937

[17] J. Lloret, P. V. Mauri, J. M. Jimenez, and J. R. Diaz, "802.11g WLANs design for rural environments video-surveillance," in *Proc. Int. Conf. Digit. Telecommun. (ICDT)*, Aug. 2006, p. 23. [Online]. Available: https://doi.org/10.1109/ICDT.2006.1

[18] M. Taha, L. Garcia, J. M. Jimenez, and J. Lloret, "SDN-based throughput allocation in wireless networks for heterogeneous adaptive video streaming applications," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Valencia, Spain, Jun. 2017, pp. 963–968. [Online]. Available: https://doi.org/10.1109/IWCMC.2017.7986416

[19] J. Huang, Q. Duan, Y. Zhao, Z. Zheng, and W. Wang, "Multicast routing for multimedia communications in the Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 215–224, Feb. 2017. [Online]. Available: https://doi.org/10.1109/JIOT.2016.2642643

[20] Z. Quin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the Internet-of-Things," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, Krakow, Poland, May 2014, pp. 1–9. [Online]. Available: https://doi.org/10.1109/NOMS.2014.6838365

[21] N. Omnes, M. Bouillon, G. Fromentoux, and O. Le Grand, "A programmable and virtualized network & IT infrastructure for the Internet of Things: How can NFV & SDN help for facing the upcoming challenges," in *Proc. 18th Int. Conf. Intell. Next Gener. Netw. (ICIN)*, Paris, France, Feb. 2015, pp. 64–69. [Online]. Available: https://doi.org/10.1109/ICIN.2015.7073808

[22] N. Bizanis and F. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," *IEEE Access*, vol. 4, pp. 5591–5606, Sep. 2016. [Online]. Available: https://doi.org/10.1109/ACCESS.2016.2607786

[23] Á. L. V. Caraguay, A. B. Peral, L. I. B. López, and L. J. G. Villalba, "SDN: Evolution and opportunities in the development IoT applications," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 5, Jan. 2014, Art. no. 735142. [Online]. Available: https://doi.org/10.1155/2014/735142

[24] K. Sood, S. Yu, and Y. Xiang, "Software-defined wireless networking opportunities and challenges for Internet-of-Things: A review," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 453–463, Aug. 2016. [Online]. Available: https://doi.org/10.1109/JIOT.2015.2480421

[25] O. G. Matlou and A. M. Abu-Mahfouz, "Utilising artificial intelligence in software defined wireless sensor network," in *Proc. 43rd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Beijing, China, Oct./Nov. 2017, pp. 6131–6136. [Online]. Available: https://doi.org/10.1109/IECON.2017.8217065

[26] M. Latah and L. Toker, "Application of artificial intelligence to software defined networking: A survey," *Indian J. Sci. Technol.*, vol. 9, no. 44, pp. 1–7, Nov. 2016. [Online]. Available: https://doi.org/10.17485/ijst/2016/v9i44/89812

[27] G. Xu, Y. Mu, and J. Liu, "Inclusion of artificial intelligence in communication networks and services," *ITU J., ICT Discoveries, Special* , no. 1, pp. 1–6, Oct. 2017.

[28] S. Sendra, A. Rego, J. Lloret, J. M. Jimenez, and O. Romero, "Including artificial intelligence in a routing protocol using software defined networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Paris, France, May 2017, pp. 670–674. [Online]. Available: https://doi.org/10.1109/ICCW.2017.7962735

[29] S. Egea, A. Rego, B. Carro, A. Sánchez-Esguevillas, and J. Lloret, "Intelligent IoT traffic classification using novel search strategy for fast based-correlation feature selection in industrial environments," *IEEE Internet Things J.*, to be published. [Online]. Available: https://doi.org/10.1109/JIOT.2017.2787959

[30] C. E. Turcu, V. G. Gaitan, and C. O. Turcu, "An Internet of Things-based distributed intelligent system with self-optimization for controlling traffic-light intersections," in *Proc. Int. Conf. Appl. Theor. Electr. (ICATE)*, Craiova, Romania, Oct. 2012, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICATE.2012.6403461

[31] P. Symes, *Digital Video Compression*. New York, NY, USA: McGraw-Hill, 2004, p. 394.

[32] A. Bovik, *Handbook of Image and Video Processing*. San Diego, CA, USA: Academic, 2000, p. 1384.

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004. [Online]. Available: https://doi.org/10.1109/TIP.2003.819861.

[34] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.

[35] Z. Kotevski and P. Mitrevski, "Performance assessment of metrics for video quality estimation," in *Proc. Int. Sci. Conf. Inf. Commun. Energy Syst. Technol. (ICEST)*, Ohrid, Macedonia, vol. 1, Jun. 2010, pp. 693–696.

[36] G. J. McLachan, *Discriminant Analysis and Statistical Pattern Recognition*. Hoboken, NJ, USA: Wiley, 2004, p. 526. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/0471725293

[37] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. Cambridge, MA, USA: MIT Press, 2001, p. 576.

[38] J. A. K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, Eds., *Advances in Learning Theory: Methods, Models and Applications* (NATO Science Series III: Computer & Systems Sciences), vol. 190. Amsterdam, The Netherlands: IOS Press, 2003, p. 436.

ALBERT REGO was born in Valencia, Spain, in 1991. He received the bachelor's degree in computer science and in telecommunications technology engineering in 2015 and the master's degree in telecommunications from the Polytechnic University of Valencia in 2016. He is currently pursuing the Ph.D. degree under the FPU National Scholarship. He has authored several papers and has cooperated in some international conferences, both by reviewing papers and being a part of a committee. His research interests include software defined networks.

ALEJANDRO CANOVAS received the M.Sc. degree in telecommunications engineering from the Polytechnic University of Valencia in 2009 and the Master en Inteligencia Artificial, Reconocimiento de Formas e Imágen Digital degree from the Department of DSIC, Polytechnic University of Valencia. He finished his Ph.D. thesis in 2016. He was a Software Programmer in several enterprises. He has been involved in several research projects related to public and private pattern recognition and artificial intelligence applied to multiple subjects. He is a member of the Integrated Management Coastal Research Institute. His research interests include statistical translation, artificial intelligence, pattern recognition, and sensors networks. His main research interests include IPTV and IPTV stereoscopic. He has several scientific papers published in international conferences and journals with impact factor. He has been involved in several organizations, for example CHINACOM, eLmL, ICC, MARSS, EDAS, SCPA, and SSPA. He is an Assistant Editor of the *NPA Journal*.

JOSE M. JIMÉNEZ received the degree in computer science engineering in 1997 and the University Master's degree in corporate networks and system integration and the University Master's degree in digital postproduction from the Polytechnic University of Valencia, Spain. He has been a Professor with the Polytechnic University of Valencia since 2005 and with the Technical School of Telecommunications Engineering since 2011. He has been a Cisco CCNP Instructor since 2004. He was a Network Designer and Administrator with several enterprises. He has several scientific papers in international conferences and international journals with JCR. He has been involved in several program committees and organization of international conferences. His research interests include wireless sensor networks, ad-hoc and 2P networks, cloud computing, and network security. He has been involved in the private sector for 18 years focused on design and management of corporative networks.

JAIME LLORET (SM'17) received the B.Sc. and M.Sc. degrees in physics in 1997, the B.Sc. and M.Sc. degrees in electronic engineering in 2003, and the Dr. Ing. degree in telecommunication engineering in 2006. He is a Cisco Certified Network Professional Instructor. He was a Network Designer and Administrator in several enterprises. He is currently an Associate Professor with the Polytechnic University of Valencia. He is the Chair of the Integrated Management Coastal Research Institute and he is the Head of the Active and Collaborative Techniques and use of Technologic Resources in the Education (EITACURTE) Innovation Group. He is the Director of the University Diploma Redes y Comunicaciones de Ordenadores and he was the Director of the University Master Digital Post Production from 2012 to 2016. He was the Vice-Chair of the Europe/Africa Region of Cognitive Networks Technical Committee (IEEE Communications Society) from 2010 to 2012 and the Vice-Chair of the Internet Technical Committee (IEEE Communications Society and Internet society) from 2011 to 2013. He was the Internet Technical Committee Chair (IEEE Communications Society and Internet Society) from 2013 to 2015. He has authored 22 book chapters and has over 400 research papers published in national and international conferences and international journals (over 180 with ISI Thomson JCR). He has been a co-editor of 40 conference proceedings and a guest editor of several international books and journals. He has been involved in over 400 program committees of international conferences, and over 150 organization and steering committees. He has led many local, regional, national, and European projects. He is currently the Chair of the Working Group of the Standard IEEE 1907.1. He has been the general chair (or the co-chair) of 40 international workshops and conferences. He is an ACM Senior and an IARIA Fellow. He is the Editor-in-Chief of the *Ad Hoc and Sensor Wireless Networks* (with ISI Thomson Impact Factor), the international journal *Networks Protocols and Algorithms*, and the *International Journal of Multimedia Communications*. He is the IARIA Journals Board Chair (eight journals) and he is (or has been) an Associate Editor of 46 international journals (16 of them with ISI Thomson Impact Factor).