IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 18, 2019, accepted August 28, 2019, date of publication September 17, 2019,
date of current version September 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941978

IoT-Guard: Event-Driven Fog-Based Video Surveillance System for Real-Time Security Management

TANIN SULTANA AND KHAN A. WAHID, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada

Corresponding author: Tanin Sultana (tanin.sultana@usask.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the Canada First Research Excellence Fund (CFREF).

ABSTRACT In this paper, we design and implement a distributed Internet of Things (IoT) framework called IoT-guard, for an intelligent, resource-efficient, and real-time security management system. The system, consisting of edge-fog computational layers, will aid in crime prevention and predict crime events in a smart home environment (SHE). The IoT-guard will detect and confirm crime events in real-time, using Artificial Intelligence (AI) and an event-driven approach to send crime data to protective services and police units enabling immediate action while conserving resources, such as energy, bandwidth (BW), and memory and Central Processing Unit (CPU) usage. In this study, we implement an IoT-guard laboratory testbed prototype and perform evaluations on its efficiency for real-time security application. The outcomes show better performance by the proposed system in terms of resource efficiency, agility, and scalability over the traditional IoT surveillance systems and state-of-the-art (SoA) approaches.

INDEX TERMS IoT, edge, fog, video surveillance, convolutional neural network, motion detection, gun-knife detection, real-time security, message queuing telemetry transport (MQTT).

I. INTRODUCTION

A smart home environment (SHE) consists of different applications of ubiquitous computing that integrates smartness into dwellings for comfort, healthcare, safety, security, and energy conservation. An SHE is monitored by ambient intelligence to provide context-aware services and to facilitate safety and security management [1]. A security management system is designed to provide complete safety from robbery, sabotage, and intrusion by monitoring the internal and external SHE, using surveillance cameras [2]. Various cyber-physical systems widely adopt the use of intelligent video surveillance (IVS) [3], [4] for automatic and accurate identification of events and objects in a target scene. IVS enables video-analytics to predict and interpret the activity of a scenario without human intervention [3]. Meanwhile, with the development of artificial intelligence (AI) and machine learning (ML), surveillance applications and security procedures are being improved with enhanced functions and accuracy [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Antonino Orsino.

According to the Uniform Crime Reports published by the Federal Bureau of Investigation (FBI), the 2017 statistics [6] show that in USA, burglaries of residential properties accounted for 67.2 percent of all burglary offenses, and the victims of these offenses suffered an estimated $3.4 billion in property losses. In addition, 15.5 percent of all robberies in 2017 occurred at commercial properties while residences experienced 16 percent. Due to this rise in property crime, the research community is paying attention to smart home security

Protective services and authorities often fail to respond to crime incidents efficiently. They tend to follow a reactive approach which relies mostly on witness reports or closed circuit television (CCTV) footage after the crime takes place. Therefore, in most cases, when an event occurs, authorities visit the location of the incident, retrieve the content manually from the camera, and then proceed to identify relevant footage either by watching the full length of the video or by processing it through specialized video analytics algorithms [7]. Thus reactive approach is naturally inefficient for preventing crimes [8]. An efficient crime predictive system could

enable robust security management in an SHE by identifying preventative procedures. Thus, the authorities could reduce crime incidents and losses. In addition, modern multimedia surveillance systems comprise a wide range of sensors, distributed over multiple sites [9]. The video surveillance system in an SHE consists of many cameras that can produce a large amount of surveillance data, both photo and video. This may result in heavy network congestion and impose complicated processing load on individual devices and systems [10]. In this paper, we will discuss an IoT-integrated intelligent video surveillance framework to provide an effective solution to this problem.

The internet of things (IoT) is the internetworking of physical objects, virtual objects, living beings, analytics, user interfaces, and network connectivity that allows these objects to collect and exchange data over an internet-based infrastructure [11]. This internetworking enables advanced IoT applications (e.g., environment monitoring, smart-city, intelligent transportation, healthcare, surveillance, and smart homes) [10], [12]. The IoT opens the door to advanced innovations to facilitate modern interactions and provides new opportunities for infrastructures and services that improve the quality of life [10]. Hence, an IoT-based smart-surveillance system can be adapted to reduce the crime rate, especially in a smart building. Although cloud-based IoT architectures are used for processing and storing essential surveillance data [9], they have issues regarding bandwidth and latency-sensitive video surveillance applications which require IoT nodes near the source of visual data to meet their delay requirements [11]. Fog computing [13] has been introduced to address these issues.

Fog, by residing in between cloud and edge devices, provides a decentralized computing infrastructure to perform a substantial amount of communication, control, storage, and management [14]. It may utilize one or more IoT end devices or near-user edge devices collaboratively. An edge device (also known as a terminal/end), on the other hand, tends to be limited to computing at the edge of the network [14]. The data-generating sensors and IoT devices are located at or near an edge node. Fog nodes can reduce the processing burden on resource-constrained edge devices [11], overcome bandwidth constraints for centralized services, and meet latency requirements of delay-sensitive applications [10]. Fog computing has the ability of responding quickly, and therefore provides on-demand services by storing and processing data locally [10]. This criterion encourages researchers to integrate fog computing in time-critical IoT applications (e.g., smart home security [SHS]) to improve real-time crime prevention.

Unlike the classical approach, where the camera sensors remain active regardless of the presence of target events or anomalies, an event-driven approach can provide better surveillance services by monitoring patterns and surveillance activity in the field of view. In this approach, an end/edge node will forward the surveillance data to the fog whenever it identifies an event (e.g., motion) in its input data streams. This approach can significantly reduce energy consumption and bandwidth due to the minimal amount of data transmission to the fog [9]. Edge computing enables this event-driven approach in a target IoT surveillance application by delegating simple processing to camera-connected, constrained IoT-edge-node devices [15]. Fog computing, on the other hand, enables AI into the system to make decisions based on previously gathered information or prior inputs which have made the system more automated [16]. Deep learning (DL), also known as deep structured learning, hierarchical learning, deep-feature learning, and deep-representation learning, is a branch of ML that represents high-level data abstractions [17]. DL enriches AI fields, such as transfer learning, computer vision, semantic parsing, and language processing. DL algorithms are more popular than the old ML algorithms, especially, in the area of computer vision [5] (i.e., object recognition, driverless cars, and AI gaming [17]). Integrating DL in the trademark of AI into the fog node will enable it to predict possible events, and decide and act on its own beforehand, which is very necessary for implementing a predictive approach for automatic and accurate identification of crime events and eventually to avoid crime incidents at an SHE.

Based on all the discussions above, we propose IoT-guard, an event-driven edge-fog-integrated video surveillance framework, to perform real-time security management by aiding in crime prevention and predicting crime events at an SHE. The proposed IoT-guard approach provides a three-layer architectural framework that orchestrates event-driven edge devices in an SHE and DL-implemented fog computing nodes to address increasing human security concerns. The system also provides an alert by sending the crime data instantly to the police or protective service, and thus, it ensures a quick response.

The main contributions of the proposed system are: (i) a resource-efficient smart-edge-node implementation to detect human intrusion and initiate fog processing; (ii) a fog-enabled infrastructure for the detection and confirmation of a crime; and (iii) an event-driven crime data reporting service to the police station to deal with a detected crime. Therefore, the proposed framework integrates image processing, AI computer vision, and network communication methods for real-time crime event detection, ensuring resource efficiency and good distribution of the processing load in an IoT-based video surveillance system. The rest of the paper is organized as follows: Section II discusses the most relevant background works; Section III provides a detailed description of the proposed IoT-guard framework; Section IV and Section V present performance evaluation of the IoT-guard laboratory prototype and comparison with the state-of-the-art architectures, respectively and finally, Section VI concludes the paper and outlines future research.

II. RELATED BACKGROUND SURVEY

This section surveys previous works on smart surveillance and analyzes techniques leveraging security and safety

services in a smart-city environment, including transportation, healthcare, industry, and residences.

Shih [18] developed an occupancy detection and tracking system for automatic monitoring and commissioning of a building with the help of an image-based depth sensor and a programmable pan–tilt–zoom camera. A device free occupant-activity sensing system using Wi-Fi-(IEEE 802.11x)-enabled IoT devices for smart homes is proposed by Yang *et al.* [19]. Lee *et al.* studied an on-road pedestrian tracking system across multiple moving cameras [20] and in another article, developed a technique for vehicle tracking and localization based on 3-D constrained multiple-kernel tracking [21]. In [22], Chen *et al.* proposed a quality-of-content-based joint source and channel coding system for detecting humans in a mobile surveillance cloud. Ajiboye *et al.* [23] proposed Fused Video Surveillance Architecture (FVSA) that enhances the public safety by utilizing data from privately-owned cameras.

Cloud-based IoT architectures are used for processing and storing essential surveillance data where each camera/node sends the data directly to a cloud for all sorts of decision making. The authors of [24] discussed the contribution of cloud technology and its secured integration into IoT architectures. Hossain [9] proposed a framework for a cloud-based multimedia surveillance system that supports the processing overload, storage requirements, access, security, and privacy in large-scale surveillance settings. These studies reveal the capability of cloud computing to satisfy many IoT requirements (e.g., monitoring, sensor stream processing, and visualization tasks). However, the large amount of real-time media data sent by the end devices using high-speed fiber networks leads to a high network deployment cost [25]. Although the situation has changed in recent years with the internetworking ability of IoT, still IoT-cloud architecture has issues regarding bandwidth, energy, and latency in real-time video surveillance applications [11], [25].

Consequently, fog computing paradigm emerged and fog–based solutions can now facilitate real-time processing and fast response time, and reduce latency issues, thus extending cloud computing and services closer to the end of the network [11], [25]. Fog, however, can be distinguished from the cloud by its proximity to the end users, the geographical distribution, and its mobility support [26]. Ni *et al.* [10] explained the architecture, features, and role of fog computing. Distributed and efficient object-detection architecture in edge computing for real-time surveillance application is also proposed in [25]. The authors in [27] explained an edge-computing framework to enable cooperative video processing on resource-abundant mobile devices for delay-sensitive multimedia IoT tasks.

To provide intelligent applications, researchers combined IoT with AI. An AI and software-defined network-(SDN)-based system for detecting and correcting multimedia transmission errors in a surveillance IoT environment is described by [28]. A DL-based pedestrian detection and face recognition technique for surveillance application in a fog-enabled IoT environment is proposed by [5]. Li *et al.* [29] showed the design of a novel offloading strategy to optimize IoT DL applications with an edge-computing environment. Cao *et al.* [30] described the design of a self-optimizing, context-driven, and energy-aware IoT wireless video sensor node for surveillance applications. A fog framework for intelligent video surveillance to enhance crime assistance and safety in public transportation is presented by [8]. Fan *et al.* [31] described a novel visualization mechanism which fuses multimodal information for large-scale intelligent video surveillance, utilizing an event-driven approach. Some architectures and frameworks for event-driven video surveillance approaches are also described in [32], [33], and [34], along with energy-aware, event-driven video surveillance solutions, such as [35] and [36].

The authors of [8] described the critical application requirements of an efficient smart-surveillance system, such as real-time and accurate detection of an event, reliable and agile prediction of crime events, and high-performance service deployment. Resource-efficient approaches add benefits in the IoT-based video surveillance architectures because of the ever-increasing number of surveillance nodes [11]. A BW-and energy-aware video compression algorithm for IoT-based video surveillance applications is proposed by [37]. However, to deploy a resource-efficient and proactive surveillance system, the previously discussed proposals may be insufficient. Therefore, this paper proposes IoT-guard, which successfully addresses all requirements. It also achieves significant efficiency compared to SoA or traditional surveillance architectures.

III. IOT-GUARD FRAMEWORK

This section describes the security management framework of the proposed IoT-guard. The system provides crime detection and proactive alerts using edge-and fog-integrated approaches. Fig. 1 illustrates the high-level architecture of the system for security management at an SHE. Camera-connected several edge nodes are set at different points to cover the inside and outside of a residential unit. The event-driven feature deployed in each of the edge nodes keeps them on standby unless any significant movement from the human intrusion is detected. If motion is detected, the edge node will capture motion-detected images and forward them and its own location to a fog node. A single fog node controls several edge nodes within one single unit or building. Several fog nodes to cover an entire residential area consisting of several buildings.

With the help of AI, each fog node can detect and identify a possible crime event and crime object by processing the motion-captured images sent by an edge node. If a fog node identifies and confirms the presence of a human and weapon, it will classify the type of weapon and immediately dispatch crime event information (i.e., a labeled image and location) to the nearest crime prevention unit (i.e., police or protective service in that area) instantly. Each fog node is also able to dispatch crime data simultaneously in the form of a

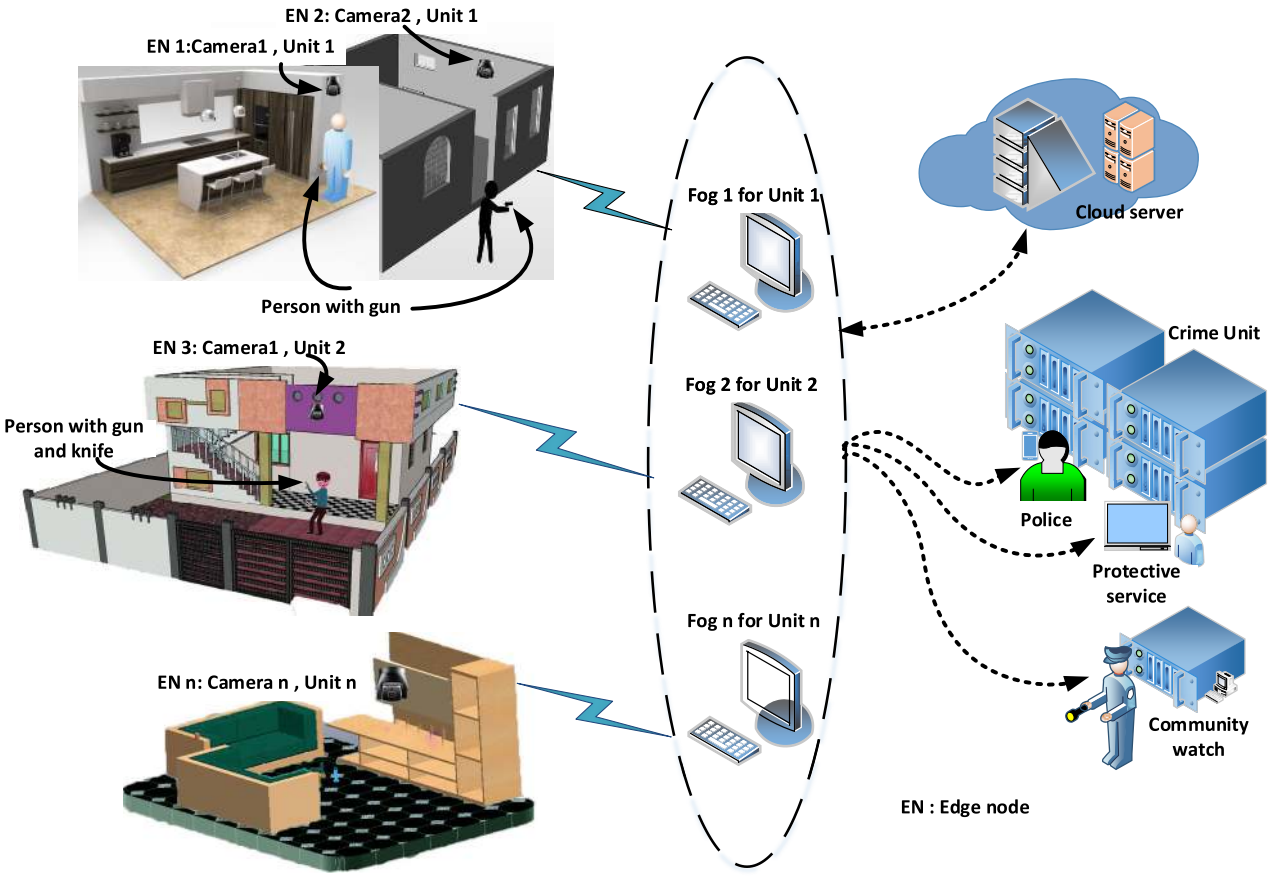

FIGURE 1. Illustration of high-level view of IoT-guard-enabled security management system.

mobile phone alert message. Using the crime data sent by the fog node, the crime prevention unit can ensure real-time crime prevention before the crime actually takes place. The AI-enabled event-driven fog node also nullifies any false positive result registered by the edge node. Each crime prevention unit may receive a crime notification from several fog nodes covering a residential area. All the fog nodes maintain bidirectional communication with a central cloud server within a smart city for receiving system updates, crime event data mining, statistical analysis, and periodic information storage.

Fig. 2 shows a general workflow diagram of the IoT-guard framework. The three-layer IoT-guard framework is discussed in detail in the following subsections.

A. EDGE-NODE PROCESSING

The system locates an edge node at the possible crime scene. The edge node, containing a camera sensor, will detect any motion, and capture motion-object images and transmit them to the fog node. The edge node divides its task into three categories: image processing, motion detection, and motion image dispatching. The image-processing functionality finds moving objects in dynamic image sequences by making use of a pixel-based change detection technique. The most intuitive method to detect change is the simple differencing of pixels, followed by thresholding. The detection of pixel-level change requires little computational cost [38]. Hence, the system will utilize the pixel-based background subtraction technique for detecting changes or motion in real-time video sequences at a constrained IoT edge node. An advantage of the pixel-based background subtraction technique is that it compensates for the lack of spatial consistency by a constant updating of the model parameters [39]. In addition, the background subtraction method enables robust segmentation by applying a threshold to the individual pixel's difference in subsequent frames, which helps to separate the moving object or foreground from the background [40]. We implemented this approach at the edge node to identify human motion. We tested the algorithm and set the threshold and sensitivity level for the system to avoid false positive results.

While the edge node continuously scans for motion using the motion detection algorithm, it does not perform any transmission of media data. Whenever it detects motion, it immediately captures motion-detected images and dispatches them to a specific fog node along with its location information. Then, it continues its function of identifying motion as before.

B. FOG-NODE PROCESSING

Deep neural network (DNN) is a popular deep learning (DL) structure that consists of multiple-layered models of inputs

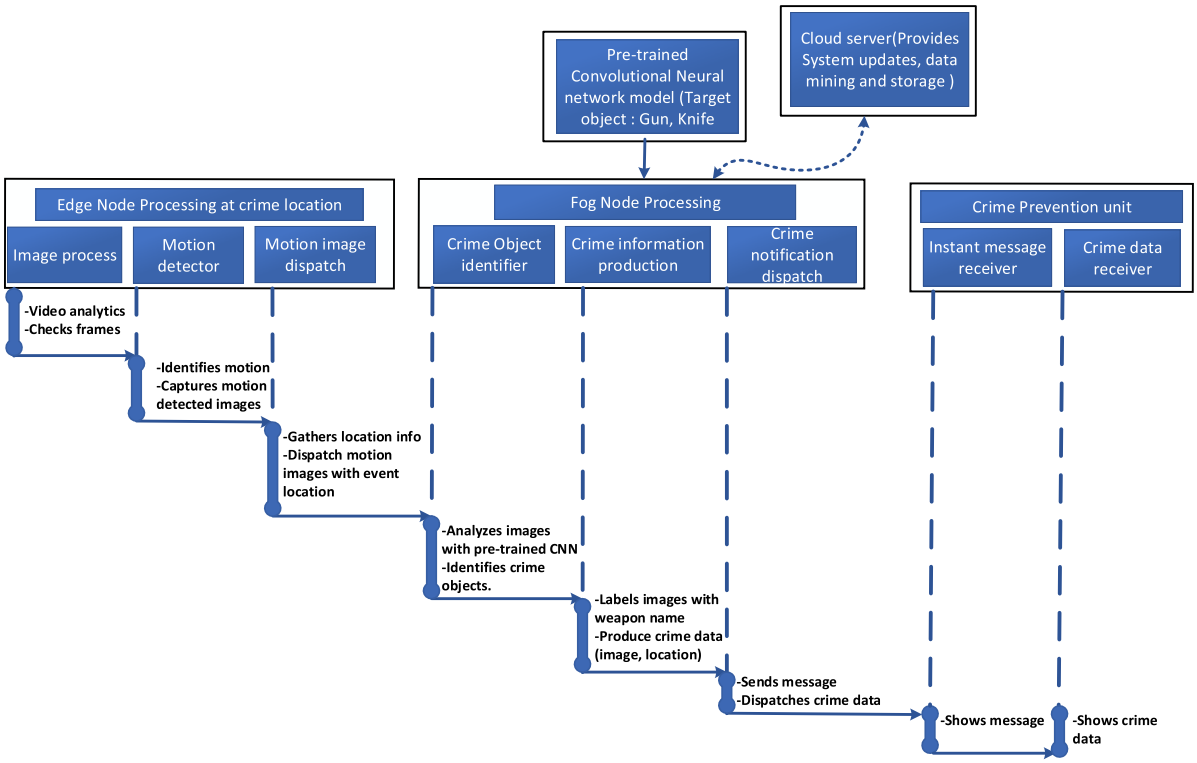

FIGURE 2. Workflow diagram of the IoT-guard framework.

that can be used as a feed-forward, convolutional, or recurrent neural network [17]. The convolutional neural network (CNN) is implemented by forming layers based on convolution, where input data is convolved to a smaller area, detecting important part within that area. Each of the convolutional layers applies nonlinear activation functions and filters in the order of hundreds to thousands and combine their results to compute the output [17]. A fog node, controlling several edge nodes, receives motion-detected images from them. Using intelligent computational methods, it then applies an object detection algorithm using a pre-trained CNN model.

The CNN architecture that we used to train and build the classifier model is a simplified version of VGGNet [41]. The model can predict crime objects. In the robberies in USA for which the Uniform Crime Reporting (UCR) Program received weapons information in 2017, firearms were used in 40.6 percent, and knives or cutting instruments in 8.1 percent [6]. Therefore, we collected gun[1] [42] and knife[2] datasets to train and build a CNN model that is capable of detecting guns and knives. Hence, the CNN model running at a fog node detects and labels the images with the name of the crime objects having the highest probability, and saves those images. The fog node then assembles and sends crime data (i.e., the labeled image, crime event location, and camera position) to the nearest crime assistance or police unit and also sends an alert message to the protective service in real-time. On the other hand, the cloud is responsible for generating updated CNN crime data models (i.e., by using transfer learning methods [43]), so that the fog node can download them whenever they are available.

C. CRIME PREVENTION UNIT

The crime unit receives the crime image, alert message, and crime location information. The alert message consists of the crime data and location, and the labeled-image verifies and confirms the crime weapon. Finally, the location information tells the police where the crime might occur so they can take necessary steps to prevent it.

IV. EVALUATION OF THE IOT-GUARD FRAMEWOK

In this section, we describe the deployment of the testbed architecture in a laboratory environment. The camera-connected edge node and the fog node are implemented using a Raspberry Pi 3 (RPi) device and a personal computer (PC), respectively. The hardware and software configurations, communication network, and the real-time experimental evaluation of the prototype are discussed elaborately in this section.

A. DEVICE AND HARDWARE CONFIGURATION

The Raspberry Pi[3] 3 model B (RPi) is one of the most common devices for emulating an IoT edge node [44].

[1]https://sci2s.ugr.es/weapons-detection
[2]https://github.com/Hvass-Labs/knifey-spoony

[3,4]https://www.raspberrypi.org

Its quad-core 64-bit ARM Cortex A53 operating Raspbian stretch (clocked at 1.2 GHz) has a built-in 802.11 n Wireless LAN and 400 MHz Video Core IV. These features enable it to encode/decode visual data and apply a motion detection algorithm to real-time video input. The Pi camera-board[4] plugs directly into a dedicated 15 pin MIPI camera serial interface (CSI) on the RPi through a 15 pin Ribbon cable. This 5 MP camera sensor can capture a maximum of 2592 × 1944 resolution static images as well as support the 1080@30fps, 720@60 fps, and 640 × 480@60/90 fps video recordings. Hence, the pi camera connected with the RPi will serve as a visual edge node for this experiment and the prototype.

An Intel(R) Core (TM) i7 processor clocked at 3.40GHz with 24GB RAM and 64-bit Windows 7 operating system will be functioning as a fog node for the experimentation. The training, validation, and testing of the CNN model are also done using the CPU of this device. Another PC with a Linux operating system will be used to receive crime data (i.e., acting as a protective service unit), while a smartphone will receive the Short Message Service (SMS) alert.

B. NETWORK AND SOFTWARE CONFIGURATION

Based on the evaluation and experimentation of [11], we set IEEE 802.11(WLAN) as the physical layer protocol, IPv4 as the network layer protocol, and, finally, MQTT as the application layer protocol for the proposed system. The MQTT client publishes multimedia data (text, image, and video) through a specific topic to the broker/server, and, then, the broker forwards the data to the clients who have subscribed to that topic [11]. The publisher/subscriber side scripts for the prototype are all written using Python[5] (version 2.7.13 and 3.5.6). Wireshark[6] (version 2.6) and tcpdump[5] are used to monitor and analyze the generated network traffic between the edge node and the fog node, and from the fog to the protective service unit. The system utilizes Eclipse mosquitto[7] message broker as the dedicated MQTT broker. With the help of an MQTT Python client library called Paho-mqtt[8], it creates applications for a multimedia publisher and subscriber at different nodes. It also connects to the mosquitto broker to route the data to the subscriber using the specified topic. In addition, the system uses the Twilio[9] API for producing SMS notifications. All these software programs and libraries are open-source and free.

C. PROTOTYPE EVALUATION

We evaluated the implemented laboratory prototype of the system in several steps. The first step evaluates the motion detection algorithm and data transmission performance of the edge node. The second step refers to the evaluation of the trained CNN model and its prediction accuracy at the fog node. Finally, the third step includes the crime data, and SMS alert transmission and reception performance of the prototype. The following subsections describe the evaluation steps.

[5]https://www.python.org
[6]https://www.wireshark.org
[7]https://mosquitto.org
[8]https://pypi.org/project/paho-mqtt
[9]https://www.twilio.com

TABLE 1. Experimental parameter at edge node.

During scanning for motion	After motion detection
Scanning image resolution: 224x160 (RGB)	Captured motion image resolution: 640x480 (JPEG)
Video FPS: 30	
Threshold (pixel to pixel intensity difference): 40	Image capturing and transmission time: 10 sec
Sensitivity: (total number of pixel difference): 100	

FIGURE 3. Image with an intruder holding a knife to be sent to the fog device, after motion is detected (image taken in a lab setting).

1) EVALUATION AT THE EDGE NODE

We implemented and tested the motion detection algorithm at the edge node. Table 1 shows the experimental parameters used to get the best output from the algorithm. The camera at the edge successfully scanned and detected an intruder, and then captured and transmitted images instantly according to the parameter described in Table 1. Fig. 3 shows a motion-detected image captured at the edge node.

2) CNN MODEL EVALUATION

We trained the fog device using around 1,000 knife images and 800 gun (i.e., pistol) images. The ratio of the training-to-testing images was 8:2. The images were categorized into two sub-labels, gun and knife, and under two labels, weapon and auto-weapon. The validation loss after 25 epochs was negligible. However, training the machine with a large number of datasets ensures better accuracy of prediction. Figs. 4 and 5 show the evaluation result of predicting crime objects using the CNN model. The model detected a knife and a gun (shown in Fig. 4) and labeled them on the images. The model is equally efficient on real-world CCTV images as well, as shown in Fig. 5.

3) CRIME DATA TRANSMISSION AND RECEPTION

The fog node increases the resolution of the crime image (800x540), then successfully dispatches the crime data (i.e., the object labeled image and location) and SMS alert to the protective service in real-time. Figs. 6 and 7 show the received crime information at the crime unit of the system. The IoT-guard can insert the name and highest probability of the crime object in the SMS notification as well.

FIGURE 4. (a) Knife and (b) gun detected and labeled on the images at the fog (Lab environment).

FIGURE 5. Detection on real-world CCTV images (image source: (a) [45], (b) [46], (c) [47], (d) [48], (e) [49], and (f) [50].

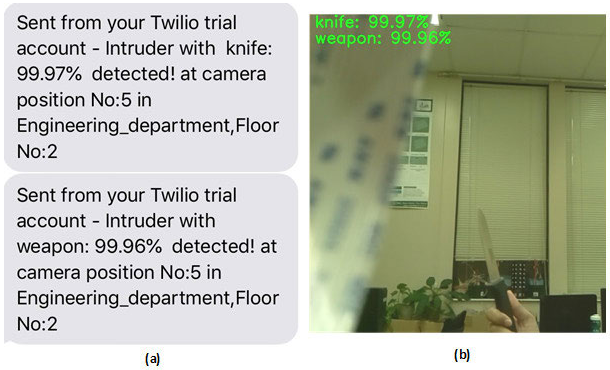

FIGURE 6. Crime information received at the crime unit, (a) SMS alert and (b) knife-detected image.

Fig. 8 shows the prototype configuration, operation, and the communication protocol stack of the proposed system as a testbed. This prototype architecture is used to evaluate the function and performance of the IoT-guard in real-time. We utilized the method described in [11] to measure different parameters of the system, such as system latency at different

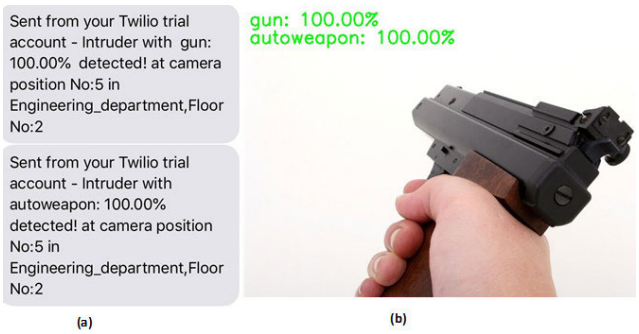

FIGURE 7. Crime information received at the crime unit, (a) SMS alert and (b) gun-detected image.

TABLE 2. Performance observation of IoT-guard.

Node name	Edge	Fog
Motion image detection time (s)	Instant	N/A
Object detection time (s) (maximum)	N/A	15
%CPU usage (maximum)	19%	12%
%Memory usage (maximum)	6%	2%
Energy consumption (J)	0.95	-
Crime data (1 Image) transmission time (s) (average)	1.00 (Edge to fog)	1.27 (Fog to police)
Crime data (location data) transmission time (s) (average)	0.4 (Edge to fog)	0.4 (Fog to police)
Crime image encoding time(s)	0.04	0.026
Maximum time (s) required by the system to report	≈ 18	

nodes, data transmission latency, percentage of CPU, and memory usage, and energy consumption.

Table 2 shows the performance of the system. Although the system was implemented in a lab environment, its performance was quick enough to report the potential crime beforehand with negligible latency.

The energy measurement at the fog node was out of the scope of this research, because the proposed architecture has focused on real-time data processing using the high-computational CNN model at the fog node to save computational cost, energy, and bandwidth at the resource-constrained edge nodes of the IoT-based surveillance system. Therefore, we performed energy measurements only at the edge node for performance evaluation and comparison.

V. PERFORMANCE COMPARISON AND DISCUSSION

In this section, we compare the performance of the proposed IoT-guard with traditional surveillance architectures and the SoA framework. Because of the importance of the resource-efficiency, video compression techniques are considered in order to save a significant amount of transmission energy during deployment in the IoT surveillance nodes [37]. Therefore, we compare the performance of our proposed system with a similar IoT-based video surveillance architecture [25], with and without an IoT video compression algorithm [37]. The parameters compared include storage

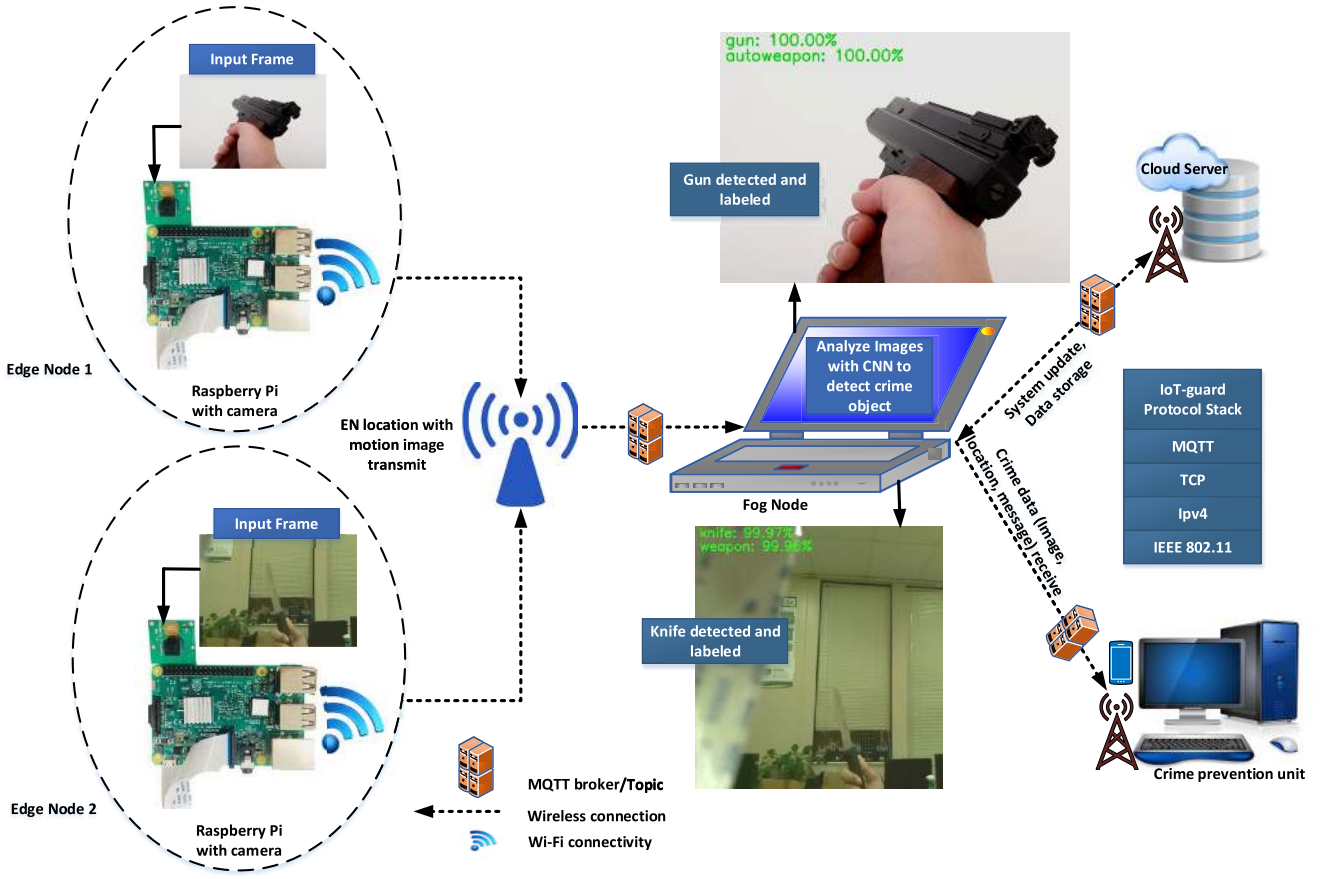

FIGURE 8. Testbed configuration used in the evaluation of the deployed IoT-guard.

requirement, transmission BW, energy consumption, and memory and CPU usage.

We implemented and compared four different IoT surveillance architectures: architecture 1, architecture 2, architecture 3, and architecture 4, which communicate from an edge/end node to a fog node for video surveillance applications. Architecture 1 [25] continuously transmits captured images/video from an end node to a fog node, where the deep-learning model processes them for object detection. Architecture 2 resembles architecture 1, except for an additional inter-frame/video compression [37] technique appended to the video captured at its edge node. Architecture 3 is the proposed IoT-guard system, where the end node utilizes a lightweight motion detection algorithm only. Architecture 4 resembles the IoT-guard but with the inter-frame/video compression [37] technique added to its edge node. We applied an intra-frame compression (i.e., JPEG [Joint Photographic Experts Group]) technique to all these four architectures as a general IoT characteristic. The evaluation was made using similar hardware in the lab environment for better comparison among these architectures.

Figs. 9 to 17 present graphical contrasts among these systems in terms of storage, BW, and energy saving and CPU usage. Tables 3 and 4 show numerical measurements of the parameters of interest of these architectures. We observed these parameters at the end node (i.e., edge) of these structures. The proposed IoT-guard transmitted only those events triggered by anomaly/motion detection incidents, while the traditional architectures, 1 and 2, continuously transmitted multimedia data throughout the day. Therefore, we varied the number of anomaly events (i.e., 1, 5, 10, 50, and 100) at the edge node of the IoT-guard to compare its performance with the traditional architectures.

It is evident from Table 3 that, although the maximum percentage of memory usage of these architectures is similar, they significantly vary in percentage of CPU usage. In the case of structures 1 and 2, the camera-connected edge nodes capture images, and process and transmit them continuously. Consequently, the percentage of CPU usage of these architectures is higher if we compare them with 3 and 4. In contrast, the proposed architecture uses a lightweight event-driven approach and transmits only if an anomaly/event occurs. Therefore, its percentage of CPU usage is significantly less than the other structures. Even though architecture 4 uses the same features as architecture 3, its additional computational burden due to the video compression algorithm makes the CPU usage higher compared to that of architecture 3. Fig. 9 presents the graphical contrast among these

TABLE 3. Comparison among different IoT-based video surveillance architectures.

IoT Architecture	Architecture 1: End device-fog [25]	Architecture 2: End device-fog with video compression	Architecture 3: Proposed IoT-guard (End/edge device-fog)	Architecture 4: Proposed IoT-guard with video compression
Intra frame compression (JPEG)	Present	Present	Present	Present
Video compression [37]	Absent	Present	Absent	Present
Motion detection time (s) (max)	N/A	N/A	instant	instant
%CPU usage (max)	58	52	19	47
%Memory usage (max)	5.9	5.9	6	6
Average storage (MBytes) required at fog (1 day)	33.86	25.4	0.072 (for 100 events)	0.0576 (for 100 events)
Transmission bandwidth consumption (Kbits/s)	2172 (for 10 sec transmission only)	1913 (for 10 sec transmission only)	250 (1 event transmission)	190 (1 event transmission)
Bandwidth consumption (Kbits/s) for 24 hours (approx.)	18.766×10^6	16.5×10^6	2500 (10 events) 12500(50 events) 25000 (100events)	1900 (10 events) 9500 (50 events) 19000 (100 events)

❖ Considering the common presence of Cloud in all the architectures, all the above parameters are measured at the end/edge node

TABLE 4. Energy consumption comparison.

IoT Architecture	1. End device- fog [25]	2.End device-fog with video compression	3.Proposed IoT-guard (End device-fog-crime Unit)		4.IoT-guard with video compression	
			During Scanning	During detection& transmission	During Scanning	During detection & transmission
Voltage (V)	5	5	5		5	5
Current consumption for the process (A) (max)	0.25	0.38	0.19	0.19	0.19	0.33
Power consumption (W = V*A)	1.25	1.9	0.95	0.95	0.95	1.65
Energy consumption (J = W*s)	1.25	1.9	0.95	0.95	0.95	1.65
Energy consumption for a whole day (86400 sec) (J)	108000J/ 108kJ	164160J/ 164.16kJ	82080J/ 82.08kJ	1.016J (1 event) 10.16J (10 events) 50.8J (50 events) 101.6J (100 events)	82080J/ 82.08kJ	1.78J (1 event) 17.8J (10 events) 89J (50 events) 178J (100 events)

❖ Normal operating voltage: 5V; operating current: 0.27A; all the above parameters are measured at the end device.

architectures in terms of the percentage of CPU usage savings. The proposed IoT-guard provides around 60 percent savings in CPU usage compared to others.

From Table 3, it is clear that the average storage requirements of structures 1 and 2 differ from the others significantly. Fig. 10 shows the efficiency of the IoT-guard in terms of savings in storage requirement. Whether the IoT-guard architecture is deployed with or without the video compression, the storage requirement is reduced by 99 percent at the fog node, even though the number of anomalies/events is 100 (Fig. 10). On the other hand, as shown in Fig. 10, associating the compression algorithm with the proposed architecture helps the IoT-guard save 20 percent more storage at the receiving fog node, which is required for day-long data transmission.

Similarly, the proposed architecture is substantially BW efficient compared to architectures 1 and 2. The event-driven feature helps the system to maintain constantly the BW savings above 99 percent. This BW efficiency is related neither to the number of events nor the integration of the video compression (Figs. 11 and Fig. 12). Although architecture 4 saves lightly more than architecture 3, especially when the number of events increases (as shown in Fig. 13), it does not impact much on the overall BW savings compared to the traditional architectures, i.e., 1 and 2 (shown in Figs. 11 and 12).

In addition, the proposed architecture is equally energy efficient either it is deployed with or without the video compression technique compared to architecture 1 and 2. Table 4 shows the comparative numerical analysis of the energy consumption of these four IoT architectures.

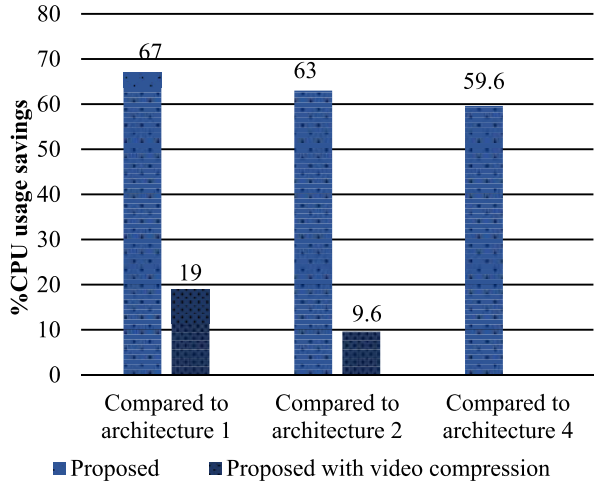

FIGURE 9. Comparison of percentage of CPU usage savings.

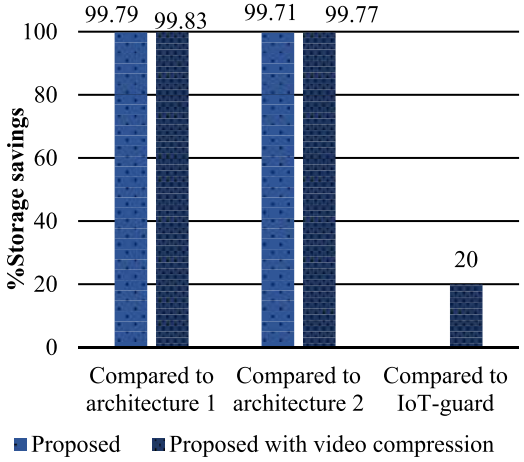

FIGURE 10. Comparison of percentage of storage savings at the fog node.

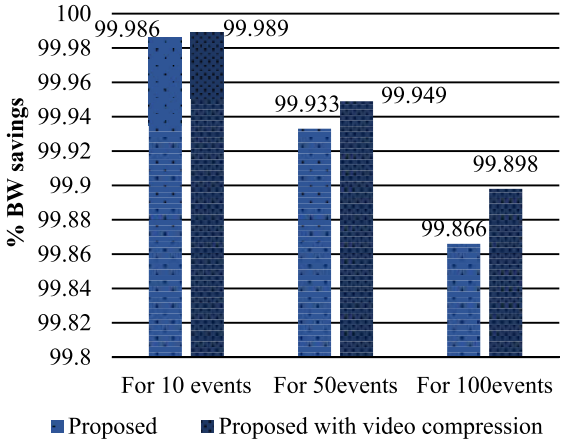

FIGURE 11. Percentage of BW savings compared to architecture 1.

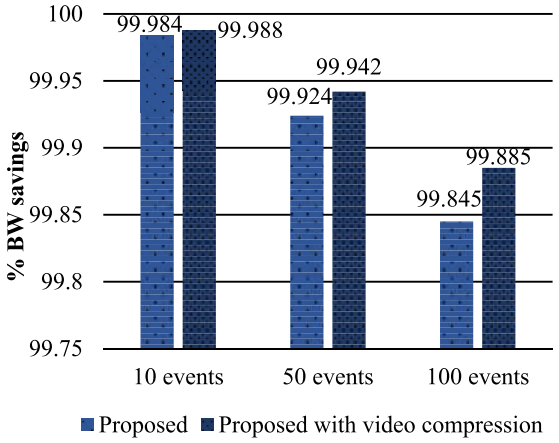

FIGURE 12. Percentage of BW savings compared to architecture 2.

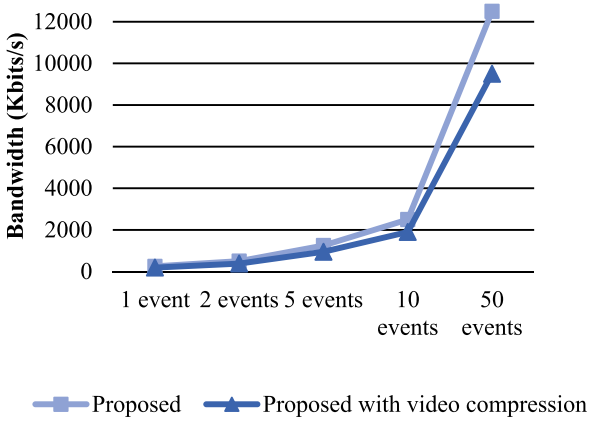

FIGURE 13. Difference in BW consumption between architectures 3 and 4 based on the varying number of events.

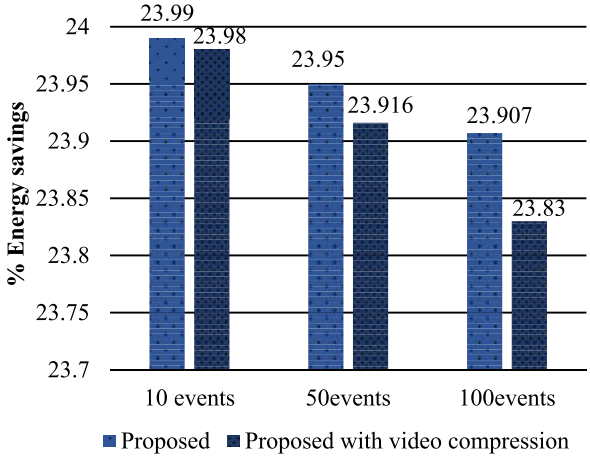

FIGURE 14. Percentage of energy savings compared to architecture 1.

Figs. 14 and 15 show that the IoT-guard can save 24 percent energy and 50 percent energy compared to architectures 1 and 2, respectively, which is unaffected by the integration of the video compression or the number of events. Moreover, the proposed system can save 42.9 percent more energy if it omits the video compression algorithm (Fig. 16). Fig. 17 shows the difference in energy consumption based on the varying number of events, between architectures 3 and

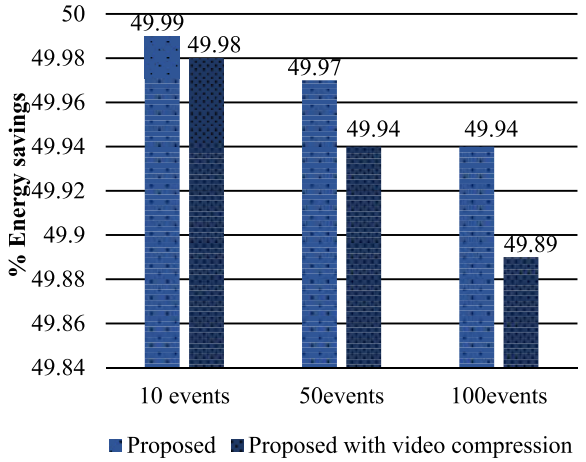

FIGURE 15. Percentage of energy savings compared to architecture 2.

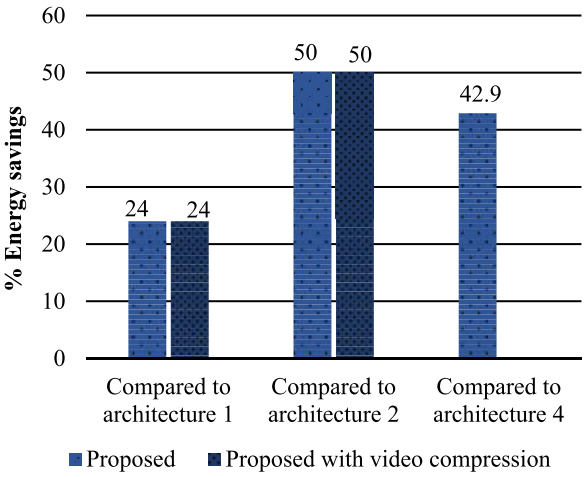

FIGURE 16. Overall comparison of percentage energy savings.

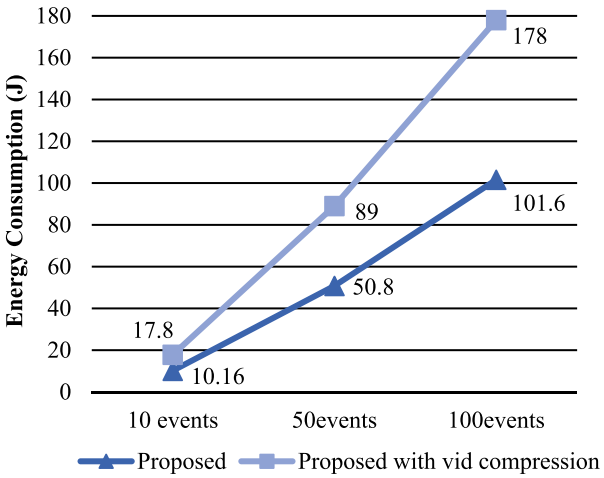

FIGURE 17. Difference in energy consumption between architectures 3 and 4 based on the varying number of events.

TABLE 5. Comparison between proposed and SoA architecture.

IoT Architecture	Proposed IoT-guard	Edge-fog unit [8]	Edge-fog-crime unit [8]
Feature	Motion detection at edge; object detection using CNN and decision at fog	Object detection and decision making at end device	Object detection using CNN at edge and fog; decision making at fog
Motion image transmission time	1sec	N/A	N/A
Total time of system operation	18sec	1 min 10s	10 min
Energy consumption for 1 event detection and transmission (max)	0.95J	81.65J	N/A
%CPU usage (max)	At edge: 19% At fog: 12%	At edge: 125%	N/A
%Memory usage (max)	At edge: 6% At fog: 2%	At edge: 60% (exceeded 10% of system memory)	N/A

architecture 4 during the detection and transmission of multimedia data.

Later, we compared our proposed system with a state-of-the-art (SoA) architecture [8]. In the SoA architecture, the authors incorporated a deep learning (DL) based real-time object-detection module in the edge node (RPi), aimed at human vision ability. According to this architecture, the edge node detects the crime object, using the DL-based object-detection module and the fog node provides all the crime object templates. The fog also performs second-level processing to confirm the crime object and then informs the crime services. Therefore, the SoA architecture conducts two-levels of processing and detection. We implemented the SoA architecture as a laboratory prototype to compare it with our proposed system by categorizing the processing into two parts: the edge-to-fog unit and the edge-to-fog-to-crime unit. Then the performance of this SoA system was evaluated and compared to the proposed one.

The implementation of the SoA architecture resulted in some significant outcomes and findings. The architecture performed poorly, when we ran the trained CNN model at the RPi edge node to detect objects in real-time. The edge node froze several times due to the computational workload imposed on it by the DL model even though the frame rate was too low (5fps). Therefore, we checked the performance using some pre-saved images and also using a single image at a time. Still, the performance in terms of memory and CPU usage, and detection time did not improve that much. Table 5 shows the results and comparison.

It is clear that running a deep-learning object-detection model requires very high CPU and memory usage in a constrained IoT device, reducing the scalability at the edge node. In addition, the low agility of the SoA system increases energy consumption significantly. Moreover, large computations can reduce system performance or freeze it at the edge node, which is very inefficient for real-time surveillance applications. On the other hand, the IoT-guard utilizes a lightweight motion detection algorithm at the constrained

TABLE 6. Qualitative comparison among different real-time surveillance systems.

System/design	[8] **2018**	[19] **2018**	[25] **2018**	[30] **2017**	[51] **2018**	**Proposed IoT-guard**
Application	Smart transportation crime detection	Smart home human occupancy and activity detection	Object detection	Detection of human presence	Detection of threats	Smart home crime detection and security
Pro-active crime management	Possible	Absent*	Absent*	Absent*	Possible	Possible
Intelligence/ Advanced AI techniques/algorithm	Deep learning	Machine learning	Faster R-CNN	Multiple machine learning algorithm	AI techniques	Deep learning (CNN)
Distributed IoT architecture to balance workload	Three-tier approach	IoT devices-cloud	End device-edge-cloud	Video node-cloud	No	Edge-fog-cloud
Context/Event-aware approach	Deep learning-based algorithm	CSI-based	No	Movement detection based	AI-based	Motion detection-based lightweight algorithm
Computational load at constrained IoT end/edge node	Heavy (deep learning)	Intermediate	Low (frame compression)	Heavy (multiple stages of computations including machine learning)	Due to heavy computational load constrained end device is not possible	Lightweight
Scalability for constrained IoT edge nodes	Absent*	Scalable	Absent*	Absent*	Scalability requires highly resourceful end node	Scalable
Transmission BW aware at constrained IoT edge nodes	BW aware	BW aware	Absent*	Absent*	Absent*	Highly BW aware
Energy aware at constrained IoT end/edge nodes	High energy consuming due to the computational burden at end node	Absent*	Absent*	Energy-aware	Not possible	Highly energy saving
Deployment cost-effectiveness	Cost-aware	Cost-aware	Cost-aware	Absent*	Expensive (also requires monthly cost per end node)	Highly cost-effective
Aware of the demand of large-scale/ever-growing surveillance nodes	Not aware	Aware	Aware	Aware	Not-aware	Highly aware

* Not mentioned/discussed

edge node and runs the heavy computational CNN model at its unconstrained fog node to detect crime objects. This approach distributes the workload quite efficiently among different nodes of an IoT surveillance system and achieves high savings in CPU (84.8 percent) and memory usage (90 percent). It also leaves room for scalability to add other computationally lightweight intelligence at the edge node and highly computational algorithms at the fog node, while keeping the proper balance in the workloads. This balance makes the proposed system faster and enables real-time crime detection. In addition, the lightweight motion detection algorithm greatly reduces the energy consumption of its edge node by about 98 percent compared to the SoA architecture, which utilizes a heavy CNN model for the same purpose at its edge node.

Therefore, the choice of effective communication architectures and protocols [11], proper distribution of computational workload, and appropriate algorithms enhance the efficiency, agility, and scalability of the system, while reducing the energy consumption significantly. The comparison results, shown in Table 5 and Fig. 18, also illustrate the greater efficiency of the system in terms of time, energy, and memory and CPU usage compared to the SoA architecture.

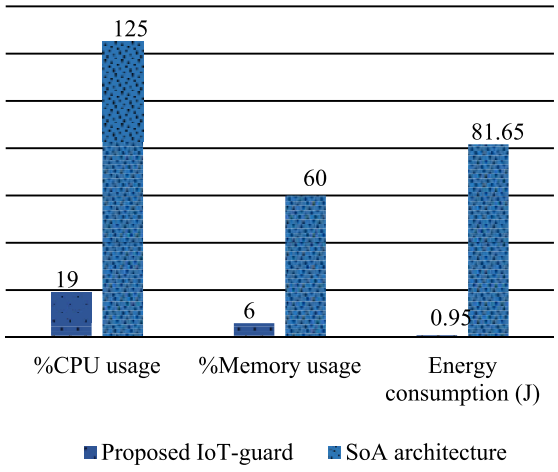

FIGURE 18. Comparison of CPU and memory usage, and energy with SoA at edge.

Finally, a generic and comparative analysis between the proposed architecture and other IoT-based architectures is made, as shown in Table 6. The comparison shows the superiority of the proposed IoT-guard architecture over others.

It incorporates criteria to provide a proactive crime detection and management system using a decentralized edge-fog-cloud-based surveillance architecture. The proper distribution of computational loads from a constrained edge node to a resourceful fog node enables the system to provide real-time operation and service. In particular, shifting the heavy processing and computational burden to the fog node significantly reduces cost, and saves energy and bandwidth consumption. The event-aware lightweight algorithm reduces the computational load at the constrained edge node, allowing scalability in order to integrate more intelligence in the future. Moreover, the high resource-awareness at the constrained edge node benefits the IoT-based video surveillance architecture to cope with the growing number of surveillance nodes [11]. Feeding context-aware critical multimedia data to the fog node and utilizing low resource-consuming constrained edge nodes allows a drastic reduction in the overall cost of the system deployment as well.

VI. CONCLUSION

In this article, we have presented the design, deployment, and performance evaluation of the IoT-guard, which is an event-driven and fog-based smart-surveillance system for real-time crime detection and security management. The application targets security management within a smart home environment under the smart-city paradigm. We evaluated the suitability and feasibility of the proposed system by deploying a laboratory testbed of the IoT-guard and observed its performance. We implemented the proposed architecture with and without a video compression algorithm and compare the performance between them along with other IoT-based video surveillance architectures. Our system outperformed others with greater efficiency in terms of energy, bandwidth, and percentage of CPU usage. Although the video compression algorithm helped our proposed IoT-guard architecture to save 20 percent more storage, it reduced its efficiency, notably in terms of energy and percentage of CPU usage. Therefore, the proposed system is far more efficient even without the video compression algorithm. Then, we performed a quantitative analysis between the proposed architecture and the SoA architecture. The outcomes proved the superiority of our proposed system in terms of agility, scalability, energy, and CPU and memory usage. Finally, an overall comparative analysis concluded the pre-eminence of the proposed IoT-guard over others, given the requirements of present and future video surveillance. We can upgrade the proposed system in the future by adding other types of crime objects or threat events to the model without changing the system configuration. Moreover, it can be further trained to detect more features in the future, for instance, utilizing transfer learning, and thus enabling it to differentiate between resident members and intruders using facial recognition features. This system could include more intelligence and services in the future for other video surveillance applications by utilizing its efficient workload management ability.

REFERENCES

[1] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1190–1203, Nov. 2012.

[2] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous Internet of Things build our future: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2011–2027, 3rd Quart., 2018.

[3] V. Gouaillier and A.-E. Fleurant, "Intelligent video surveillance: Promises and challenges," CRIM Technopôle Defence Secur., Montreal, QC, Canada, Technol. Commercial Intell. Rep., Mar. 2009, vol. 456, p. 468. [Online]. Available: http://docshare04.docshare.tips/files/7051/70518279.pdf

[4] T. Sultana, M. W. Alam, and K. A. Wahid, "Reliability analysis of Io VT based intelligent video surveillance system," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Aug. 2018, pp. 1–4.

[5] S. Din, A. Paul, A. Ahmad, B. B. Gupta, and S. Rho, "Service orchestration of optimizing continuous features in industrial surveillance using big data based fog-enabled Internet of Things," *IEEE Access*, vol. 6, pp. 21582–21591, 2018.

[6] *FBI—Robbery*. Accessed: Apr. 12, 2019. [Online]. Available: https://ucr.fbi.gov/crime-in-the-u.s/2017/crime-in-the-u.s.-2017/topic-pages/robbery

[7] G. Kioumourtzis, M. Skitsas, N. Zotos, and A. Sideris, "Wide area video surveillane based on edge and fog computing concept," in *Proc. 8th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Aug. 2017, pp. 1–6.

[8] A. J. V. Neto, Z. Zhao, J. J. P. C. Rodrigues, H. B. Camboim, and T. Braun, "Fog-based crime-assistance in smart IoT transportation system," *IEEE Access*, vol. 6, pp. 11101–11111, 2018.

[9] M. A. Hossain, "Framework for a cloud-based multimedia surveillance system," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 5, May 2014, Art. no. 135257.

[10] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 601–628, 1st Quart., 2018.

[11] T. Sultana and K. A. Wahid, "Choice of application layer protocols for next generation video surveillance using Internet of video things," *IEEE Access*, vol. 7, pp. 41607–41624, 2019.

[12] M. W. Alam, M. H. A. Sohag, A. H. Khan, T. Sultana, and K. A. Wahid, "IoT-based intelligent capsule endoscopy system: A technical review," in *Intelligent Data Analysis for Biomedical Applications*, 1st ed. New York, NY, USA: Academic, 2019, pp. 1–20.

[13] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[14] M. Chiang, S. Ha, C.-L. I, F. Risso, and T. Zhang, "Clarifying fog computing and networking: 10 questions and answers," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 18–20, Apr. 2017.

[15] B. Skrbic, D. Radovanovic, S. Tomovic, L. Lazovic, Z. Zecevic, and I. Radusinovic, "A decentralized platform for heterogeneous IoT networks management," in *Proc. 23rd Int. Sci.-Prof. Conf. Inf. Technol. (IT)*, Feb. 2018, pp. 1–4.

[16] S. Wu, J. B. Rendall, M. J. Smith, S. Zhu, J. Xu, H. Wang, Q. Yang, and P. Qin, "Survey on prediction algorithms in smart homes," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 636–644, Jun. 2017.

[17] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, 1st Quart., 2017.

[18] H.-C. Shih, "A robust occupancy detection and tracking algorithm for the automatic monitoring and commissioning of a building," *Energy Buildings*, vol. 77, pp. 270–280, Jul. 2014.

[19] J. Yang, H. Zou, H. Jiang, and L. Xie, "Device-free occupant activity sensing using WiFi-enabled IoT devices for smart homes," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3991–4002, Oct. 2018.

[20] K.-H. Lee and J.-N. Hwang, "On-road pedestrian tracking across multiple driving recorders," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1429–1438, Sep. 2015.

[21] K.-H. Lee, J.-N. Hwang, and S.-I. Chen, "Model-based vehicle localization based on 3-D constrained multiple-kernel tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 1, pp. 38–50, Jan. 2015.

[22] X. Chen, J.-N. Hwang, D. Meng, K.-H. Lee, R. L. de Queiroz, and F.-M. Yeh, "A quality-of-content-based joint source and channel coding for human detections in a mobile surveillance cloud," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 1, pp. 19–31, Jan. 2017.

[23] S. O. Ajiboye, P. Birch, C. Chatwin, and R. Young, "Hierarchical video surveillance architecture: A chassis for video big data analytics and exploration," *Int. Soc. Opt. Photon.*, vol. 9407, Mar. 2015, Art. no. 94070K.
[24] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018.
[25] J. Ren, Y. Guo, D. Zhang, Q. Liu, and Y. Zhang, "Distributed and efficient object detection in edge computing: Challenges and solutions," *IEEE Netw.*, vol. 32, no. 6, pp. 137–143, Nov./Dec. 2018.
[26] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency Comput. Pract. Exper.*, vol. 28, no. 10, pp. 2991–3005, 2016.
[27] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018.
[28] A. Rego, A. Canovas, J. M. Jiménez, and J. Lloret, "An intelligent system for video surveillance in IoT environments," *IEEE Access*, vol. 6, pp. 31580–31598, 2018.
[29] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan. 2018.
[30] N. Cao, S. Bin Nasir, S. Sen, and A. Raychowdhury, "Self-optimizing IoT wireless video sensor node with *in-situ* data analytics and context-driven energy-aware real-time adaptation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 9, pp. 2470–2480, Sep. 2017.
[31] C.-T. Fan, Y.-K. Wang, and C.-R. Huang, "Heterogeneous information fusion and visualization for a large-scale intelligent video surveillance system," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 4, pp. 593–604, Apr. 2017.
[32] D. Kieran and W. Yan, "A framework for an event driven video surveillance system," in *Proc. 7th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug./Sep. 2010, pp. 97–102.
[33] A. Dimou, A. Axenopoulos, D. Matsiki, and P. Daras, "A user-centric approach for event-driven summarization of surveillance videos," in *Proc. 6th Int. Conf. Imag. Crime Prevention Detection*, Jul. 2015, pp. 1–6.
[34] A. Sakaushi, K. Kanai, J. Katto, and T. Tsuda, "Edge-centric video surveillance system based on event-driven rate adaptation for 24-hour monitoring," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2018, pp. 651–656.
[35] N. Al-Yamani, S. Qaisar, A. Alhazmi, S. Mohammad, and A. Subasi, "An event driven surveillance system," in *Proc. 5th Int. Conf. Electron. Devices, Syst. Appl. (ICEDSA)*, Dec. 2016, pp. 1–4.
[36] T. Mekonnen, E. Harjula, A. Heikkinen, T. Koskela, and M. Ylianttila, "Energy efficient event driven video streaming surveillance using SleepyCAM," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, Aug. 2017, pp. 107–113.
[37] R. Hasan, S. K. Mohammed, A. H. Khan, and K. A. Wahid, "A color frame reproduction technique for IoT-based video surveillance application," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
[38] L. Li and M. K. H. Leung, "Integrating intensity and texture differences for robust change detection," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 105–112, Feb. 2002.
[39] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.
[40] J. Kang, D. V. Anderson, and M. H. Hayes, "Face recognition for vehicle personalization with near infrared frame differencing," *IEEE Trans. Consum. Electron.*, vol. 62, no. 3, pp. 316–324, Aug. 2016.
[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[42] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," *Neurocomputing*, vol. 275, pp. 66–72, Jan. 2018.
[43] Z. Huang, Z. Pan, and B. Lei, "Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data," *Remote Sens.*, vol. 9, no. 9, p. 907, 2017.
[44] A. Sammoud, A. Kumar, M. Bayoumi, and T. Elarabi, "Real-time streaming challenges in Internet of video things (IoVT)," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.
[45] *CCTV Image of Man Wanted in Connection with Scunthorpe Armed Robbery Credit: Humberside Police*. Accessed: Jul. 17, 2019. [Online]. Available: https://www.itv.com/news/calendar/topic/armed-robbery/
[46] *Armed Robbery at Shop in Manchester Caught on CCTV | Daily Mail Online*. Accessed: Jul. 17, 2019. [Online]. Available: https://www.dailymail.co.uk/video/news/video-1083572/Armed-robbery-shop-Manchester-caught-CCTV.html
[47] *Knife-Wielding Robber Sentenced to Nearly 12 Years' Jail | ABC News*. Accessed: Jul. 17, 2019. [Online]. Available: https://www.abc.net.au/news/2015-05-15/man-sentenced-to-jail-over-spate-of-canberra-robberies/6473788
[48] *Video Shows Saturday stabbing Attack at Damascus Gate | The Times of Israel*. Accessed: Jul. 3, 2019. [Online]. Available: https://www.timesofisrael.com/video-shows-saturday-stabbing-attack-at-damascus-gate/
[49] *CCTV of Armed Robber Getting His Comeuppance Goes Viral | Daily Mail Online*. Accessed: Jul. 17, 2019. [Online]. Available: https://www.dailymail.co.uk/news/article-4505308/CCTV-armed-robber-getting-comeuppance-goes-viral.html
[50] *How Tunnel Vision Impacts First Responders—Law Enforcement Today*. Accessed: Jul. 17, 2019. [Online]. Available: https://www.lawenforcementtoday.com/tunnel-vision-impacts-first-responders/
[51] *Why Athena Security Camera Systems are Different | Athena Security Camera System | Gun Detection*. Accessed: Apr. 10, 2019. [Online]. Available: https://athena-security.com/about-us

TANIN SULTANA received the B.Sc. degree in electrical and electronic engineering from the Chittagong University of Engineering and Technology, Chittagong, Bangladesh, in 2013. She is currently pursuing the M.Sc. degree in electrical engineering with the University of Saskatchewan, Saskatoon, SK, Canada.

She was an Instructor with the Bay Maritime Institute, Chittagong, in 2014. She was a Research Assistant with Ajou University, Suwon, South Korea, in 2015. She has been a Lecturer with East Delta University, Bangladesh, since 2016. She has been a Research Assistant with the Multimedia Processing and Prototyping Laboratory (MPP lab), since May 2017. She is the author of five conference articles and three articles and contributed to one book chapter. Her current research interests include, but not limited to, the Internet-of-Things (IoT)-based smart systems, video surveillance and analytics, the IoT-based health informatics, capsule endoscopy, and wireless sensor networks.

Ms. Sultana received the prestigious Deans Scholarship from the University of Saskatchewan, the BK 21 Scholarship in South Korea, and the Undergraduate Level Scholarship from the Chittagong University of Engineering and Technology.

KHAN A. WAHID (S'02–M'07–SM'13) received the B.Sc. degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2000, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Calgary, Calgary, AB, Canada, in 2003 and 2007, respectively.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK, Canada. He has coauthored two book chapters and more than 140 peer-reviewed journals and international conference papers in the fields of video and image processing, embedded systems, the Internet of Things, medical imaging, and health informatics. He holds two patents.

Dr. Wahid received many prestigious awards and scholarships, including the Most Distinguished Killam Scholarship and the NSERC Canada Graduate Scholarship for his doctoral research and the Award of Innovation from Innovation Place, in 2016. He is also a registered Professional Engineer in the Province of Saskatchewan.