IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 21, 2021, accepted July 31, 2021, date of publication August 5, 2021, date of current version August 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102598

FogSurv: A Fog-Assisted Architecture for Urban Surveillance Using Artificial Intelligence and Data Fusion

ARSLAN MUNIR[1], (Senior Member, IEEE), JISU KWON[2], (Member, IEEE), JONG HUN LEE[3], (Student Member, IEEE), JOONHO KONG[3], (Member, IEEE), ERIK BLASCH[4], (Fellow, IEEE), ALEXANDER J. AVED[5], (Senior Member, IEEE), AND KHAN MUHAMMAD[6], (Member, IEEE)

[1]Department of Computer Science, Kansas State University, Manhattan, KS 66506, USA
[2]Samsung Electronics Co., Ltd., Suwon, Gyeonggi-Do 443743, South Korea
[3]School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, South Korea
[4]AFRL Air Force Office of Scientific Research (AFOSR), Arlington, VA 22203, USA
[5]United States Air Force Research Laboratory (AFRL), Information Directorate, Rome, NY 13441, USA
[6]Visual Analytics for Knowledge Laboratory (VIS2KNOW Lab), School of Convergence, College of Computing and Informatics, Sungkyunkwan University, Seoul 03063, Republic of Korea

Corresponding author: Arslan Munir (amunir@ksu.edu)

This work was supported in part by the Air Force Research Laboratory (AFRL) Information Directorate (RI) through the Air Force Office of Scientific Research (AFOSR) Summer Faculty Fellowship Program® under Contract FA8750-15-3-6003, Contract FA9550-15-0001, and Contract FA9550-20-F-0005.

ABSTRACT *Urban surveillance*, of which airborne urban surveillance is a vital constituent, provides *situational awareness* (SA) and timely response to emergencies. The significance and scope of urban surveillance has increased manyfold in recent years due to the proliferation of unmanned aerial vehicles (UAVs), Internet of things (IoTs), and multitude of sensors. In this article, we propose **FogSurv**—a fog-assisted surveillance architecture and framework leveraging artificial intelligence (AI) and information/data fusion for enabling real-time SA and monitoring. We also propose an AI- and data-driven information fusion model for FogSurv to help provide (near) real-time SA, threat assessment, and automated decision-making. We further present a latency model for AI and information fusion processing in FogSurv. We then discuss several use cases of FogSurv that can have a huge impact on multifarious fronts of national significance ranging from safeguarding national security to monitoring of critical infrastructures. We conduct an extensive set of experiments to demonstrate that FogSurv using AI and data fusion help provide near real-time inferences and SA. Experimental results demonstrate that FogSurv provides a latency improvement of 37% on average over cloud architectures for the selected benchmarks. Results further indicate that combining AI with data fusion as in FogSurv can provide a speedup of up to 9.8× over AI without data fusion while also maintaining or improving the inference accuracy. Additionally, results show that AI combined with fusion of different image modalities obtained through UAVs in FogSurv results in improved average precision of target detection for surveillance as compared to AI without data fusion for different target scales and environment complexity.

INDEX TERMS Urban surveillance, situational awareness, fog computing, unmanned aerial vehicles, information fusion, artificial intelligence, deep neural networks.

I. INTRODUCTION AND MOTIVATION

Recent push towards urbanization and smart cities is accompanied by the desire for enhancing the quality of life and sustainability of cities, a holistic understanding of dynamic city elements, increased *situational awareness* (SA), and the ability to react instantly to emergencies. *Urban surveillance* and *monitoring*, of which airborne surveillance is an integral part, help realize sustainability and safety of smart cities by providing SA, and help enable timely response to threats and emergencies. The significance of urban monitoring and thus the architectures enabling urban surveillance has been

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

increased in recent years due to incidents of domestic terrorism and violence. For example, on the night of October 1, 2017, a gunman opened fire on a crowd of concertgoers on the Las Vegas Strip in Nevada. The perpetrator fired more than 1,100 rounds from his suite on the 32nd floor of a nearby hotel, killing 58 people and leaving 851 injured from gunfire and the resulting panic [1]. This and other similar events [2] necessitate the need of comprehensive surveillance and monitoring of cities, in particular, critical infrastructures, public places, hotels, and tourist attractions to detect threats and provide proactive safety measures for public assurance.[1]

The significance of urban surveillance has further increased manyfold in recent years due to the proliferation of *gray zone* warfare and conflicts where the gray zone actors use techniques such as misinformation, intimidation, and disruptions of services and critical infrastructures to destabilize nations and potentially produce auspicious conditions for military engagements. Although urban surveillance and monitoring mechanisms have been introduced previously, most of these mechanisms are not able to perform in real-time. Real-time urban surveillance and monitoring are imperative to enhance SA, public safety, and national security, and thus this article focuses on real-time surveillance and monitoring.

This article proposes **FogSurv**—a fog-assisted architecture and framework leveraging artificial intelligence (AI)/machine learning (ML), information fusion, and dynamic data-driven methods for enabling real-time urban surveillance and monitoring. FogSurv avails advances in *fog computing* (a novel trend in computing that pushes applications, services, data, computing power, knowledge generation, and decision-making away from the centralized nodes to the logical extreme edges of a network) to realize real-time surveillance and monitoring. Since the surveillance and monitoring systems, including airborne surveillance vehicles, need to be steered according to environmental conditions, such as time of operation, weather conditions, geographical information, and the target (e.g., vehicle, person) activity, FogSurv leverages dynamic data-driven applications systems (DDDAS) methods to furnish dynamic measurement and control to the surveillance and monitoring system.

Our main contributions in this article are as follows:

- We propose FogSurv, a framework for real-time urban surveillance, which comprises of an architecture that integrates Internet of things (IoT) devices, fog, and cloud for surveillance, a framework for AI/ML processing, and AI- and data-driven information fusion, and analytics.
- We propose a novel integrated fog, cloud, Internet of things (IFCIoT) architecture for surveillance and monitoring (IFCIoT-Surv) within the FogSurv framework that integrates IoT devices with cloud servers via intermediary fog nodes.
- We propose the usage of DDDAS methods for enabling IoT and unmanned aerial vehicles (UAVs) in FogSurv to adapt their sensing, processing, and navigation (routing) dynamically based on live sensor data for better monitoring and tracking of the target and enhancing SA. We further propose an AI- and DDDAS-inspired information fusion model in FogSurv that integrate advances in ML and DDDAS methods in information fusion to steer the sensor mode selection and measurement process in accordance with environmental conditions, topological and contextual information, and mission objectives.
- We propose a framework for AI/ML processing in FogSurv to assist with surveillance tasks.
- We discuss several use cases for FogSurv that can have a huge impact on multifarious fronts of national significance.
- We have mathematically modeled latency of AI and information (data) fusion processing in FogSurv considering transmission, processing, and queueing delays at IoT devices, fog/edge, and cloud. Based on the latency model, we also establish conditions for computation offloading in IoT-fog-cloud hierarchy.
- We experimentally evaluate timeliness advantages of FogSurv over contemporary cloud only architectures for urban surveillance.
- We experimentally evaluate the effectiveness of combining data fusion with AI for surveillance by providing a latency and accuracy comparison between convolutional neural networks (CNN) processing with and without data fusion.
- We experimentally evaluate the improvements in average precision and latency provided by combined multimodal data fusion and AI for target detection in surveillance by UAVs in FogSurv as compared to AI without data fusion for different target scales and environment complexity.

We would like to clarify that the IFCIoT-Surv architecture refers to integrated fog, cloud, IoT architecture for urban surveillance whereas FogSurv refers to the overall architecture and framework for real-time urban surveillance, which comprises of the IFCIoT-Surv architecture, framework for AI/ML processing, AI- and data-driven information fusion, and analytics. The remainder of this article is organized as follows. Section II discusses previous works related to urban surveillance. Section III describes our proposed IFCIoT-Surv architecture. Section IV elaborates the usage of UAVs for enhancing SA in FogSurv. A framework for AI/ML processing in FogSurv is presented in Section V. Section VI presents a framework for AI- and data-driven information fusion in FogSurv. Latency of AI and information fusion processing in FogSurv is mathematically modeled in Section VII. Use cases for FogSurv are discussed in Section VIII. Experimental results demonstrating the superiority of FogSurv over only cloud based systems for urban surveillance are presented in Section IX. Section IX further experimentally establishes the effectiveness of combining data fusion with AI for surveillance in FogSurv. Finally, Section X concludes this article.

[1]The U.S. Department of the Air Force notes that the U.S. Department of Defense mission does not include surveillance of cities in the United States.

II. RELATED WORK

This section discusses previous works in literature related to urban surveillance systems, information fusion for urban surveillance systems, and computation offloading and resource management in fog-cloud computing environments.

A. URBAN SURVEILLANCE SYSTEMS

Urban surveillance has been studied in previous works, and surveillance systems leveraging IoT, fog, and cloud have been proposed in literature. Carminati *et al.* [3] discussed the prospects of wireless sensor networks for urban monitoring. The authors highlighted the significance of miniaturized and low-power sensors for both fixed and mobile platforms. Motlagh *et al.* [4] proposed a UAV-based IoT platform and a single-camera based surveillance architecture for crowd surveillance. They used local binary pattern histogram algorithm for facial recognition. The work compared two approaches to video data processing for facial recognition application: first where the video data was processed locally onboard the UAVs, and the second where the video data was offloaded by the UAV to a mobile edge computing (MEC) node. Results revealed that the second approach wherein the data was offloaded by UAVs to an MEC significantly improved energy efficiency and system responsiveness.

Chen *et al.* [5] used a single-camera based surveillance architecture comprising of a UAV, a ground controller and a fog computing node. They used their system to monitor the speed of vehicles on different roads. In their proposed system, the UAV captured videos and then offloaded the video processing to the fog node. The work utilized a single-camera based single vehicle tracking algorithm and extended this to multiple vehicle tracking by using parallel instances of the single vehicle tracking algorithm. Wang *et al.* [6] described a three-tiered edge computing system architecture for urban surveillance. Their proposed system architecture consisted of data tier (bottom-most), edge cloud tier (middle) and application tier (top-most). Their system architecture integrated network function virtualization (NFV) and software defined networking (SDN) for hardware resource virtualization and programmable virtual networks in the edge cloud tier, which allowed creation of dedicated virtual machine instances at the edge cloud nodes for each surveillance task. Results indicated that systems with edge computing nodes provided around 10× faster services than the central cloud-based systems.

Pavlidis *et al.* [7] proposed a prototype urban surveillance system named DETER (Detection of Events for Threat Evaluation and Recognition) for inferring and reporting threats. The DETER system was an integration of computer vision and threat assessment technologies. The computer vision technology consisted of moving object segmentation and tracking, and multi-camera fusion. The threat assessment consisted of feature assembly, offline training and threat classification. The DETER system could be implemented using commercial off-the-shelf (COTS) equipment, which made it compatible with legacy systems. Hu and Ni [8] proposed an IoT-driven urban surveillance system focusing on vehicle recognition and vehicle license plate recognition (VLPR). The work described a filtering-based algorithm for vehicle recognition and VLPR that reduced the image size to minimize data transmission and data storage overheads. Such size reduction filters could be helpful in reducing the cost and overhead of massive surveillance systems. In our earlier work [9], we proposed a correlation filter-based object tracker that could work under varying light conditions. Our proposed technique carried out detection of multiple objects' positions frame-by-frame by performing cyclic shift operations at the objects' positions of the previous frame, which generated a series of candidate windows. Afterwards, correlation operation was performed on the candidate windows with the filter which had been trained on the previous frame, and then the candidate window with the highest confidence was selected as the target. The work also utilized fog computing for the acceleration of the proposed tracker. This work and other previous works related to fog-assisted surveillance; however, did not utilize data fusion. The performance and accuracy of these works can be further improved by mapping the tracking algorithms to the IFCIoT-Surv architecture and leveraging data fusion.

Although there exist works in literature related to urban surveillance, real-time urban surveillance has received less attention. This work proposes a framework for real-time urban surveillance. Furthermore, most of the previous works did not quantify the advantages and speedups of using AI and information fusion. The emphasis of this work is to experimentally quantify the advantage of using fog computing, AI, and information fusion for urban surveillance over cloud-only approaches.

B. INFORMATION FUSION FOR URBAN SURVEILLANCE SYSTEMS

Information fusion for urban surveillance has received little attention in prior works. Liu *et al.* [10] discussed information fusion in distributed computing paradigms. The authors compared four different distributed computing paradigms, viz., cluster computing, peer-to-peer computing, grid computing, and cloud computing, in terms of parallelism, scalability, elasticity, availability, power efficiency, and security. The work also conducted a preliminary performance evaluation of a video tracking application in cloud. An earlier work by one of the authors [11] presented an unmanned airborne system (UAS) based emitted monitoring (UEM) system comprising of two separated airborne direction of arrival (DOA) sensors and a control station. The UEM system employed coarse and fine frequency scanning to enable near real-time detection. Furthermore, the fusion of DOAs from the two sensors and the terrain map provided localization accuracy. Simulation results verified that the proposed system could localize emitters within 100 meters (m) of its real location in an area of 2000-m-radius. This work is complementary to our work on urban surveillance because such UEM systems can be utilized in FogSurv to enhance the detection of

various radio frequency (RF) emitters in urban areas. Since information fusion for urban surveillance has received little attention in literature, this work demonstrates the advantage of information fusion with AI in a fog computing-based system for real-time surveillance application.

C. COMPUTATION OFFLOADING AND RESOURCE MANAGEMENT IN FOG-CLOUD COMPUTING ENVIRONMENTS

Some previous works have considered computation offloading and resource management in fog-cloud computing environments. Shukla and Munir [12] proposed a computation offloading architecture where the data generated by IoT devices was offloaded to fog servers for computation. The proposed fog-based computation offloading architecture was compared with cloud-based computation offloading models. The results indicated that the fog-based computation offloading could provide a performance improvement of 21.4% over cloud-based computation offloading schemes. The work, however, did not focus on surveillance application nor considered information fusion during computation offloading. Eshratifar and Pedram [13] studied the efficiency of offloading only some layers of deep neural networks (DNNs) to the cloud. They developed a computation offloading framework for DNN inferences considering the battery usage constraints on the mobile side and available resources on the cloud. The work; however, did not consider computation offloading to fog nodes and servers. Our work develops a latency model for AI and information fusion in FogSurv framework, and establish conditions for computation offloading in IoT-fog-cloud hierarchy.

Mahmud *et al.* [14] proposed a latency-aware application module management policy for fog computing environment that was designed to meet latency requirements for different applications. The policy's objective was to ensure application's quality of service in meeting service delivery deadlines and to optimize resource usage in fog computing environments. The policy also suggested to relocate modules in order to optimize the number of computationally active fog nodes. The policy was evaluated in iFogSim-simulated fog environment. Naha *et al.* [15] proposed resource allocation and provisioning algorithms in a fog-cloud environment to satisfy deadline-based dynamic user behavior. They first ranked the available resources in fog devices, fog servers, and cloud based on their available processing, bandwidth, and latency. After that, they provisioned resources in a hierarchical fashion from the ranked list by considering the deadline provided by the users. The provisioning algorithm first checked the ranked fog devices to see if fog devices had enough resources to assign the task submitted by the user. If fog devices did not have sufficient resources to meet the submitted task's requirements, then first fog servers alone, and then fog devices and fog servers combined, were considered for task allocation. If even fog servers, or fog servers and fog devices combined, did not have required resources, then the cloud only, or cloud and fog devices combined, or cloud and fog servers combined, were provisioned for task execution. The evaluation was conducted by extending the CloudSim simulation toolkit, and showed promising results in terms of overall processing time, delay, and estimated monetary cost. In another work, Naha and Garg [16] proposed a multi-criteria-based resource allocation policy with resource reservation in a fog environment to minimize overall delay, processing time, and service-level agreement (SLA) violations. The authors leveraged multiple objective functions to determine suitable resources for executing time-sensitive tasks in a fog environment while considering fog environment characteristics, such as device heterogeneity, resource constraints, mobility, and dynamic changes in user requirements.

The resource management works discussed above; however, evaluated the resource management policies in simulation environments. Our proposed FogSurv framework can be used for experimental evaluation of these resource management policies as our proposed framework supports handoff between different fog nodes as an IoT device moves from the vicinity of one fog node to another. Also, these resource management policies did not focus on surveillance applications. Furthermore, these works did not benchmark the performance of fog-based surveillance application on physical edge devices, whereas our work focuses on quantifying the advantages of information fusion on AI-based surveillance applications. Finally, these computation offloading and resource management works are complementary to this work as computation offloading and resource management for surveillance applications can be integrated in FogSurv framework, but is not the focus of this work.

Although some previous works have proposed urban surveillance architectures, the previous works have deficiencies that make them inefficient for real-time urban surveillance and monitoring. The **FogSurv** proposed in this article incorporates additional features and services, such as AI/ML processing, AI- and DDDAS-inspired adaptive sensing and information fusion, which help enable real-time, energy-efficient, accurate, and reliable urban surveillance and monitoring. Furthermore, previous works on computation offloading and resource management were evaluated on simulations, whereas this work benchmarks AI and information fusion on IoT devices, fog nodes, and cloud platforms.

III. IFCIoT-SURV: INTEGRATED FOG CLOUD IoT ARCHITECTURAL PARADIGM FOR SURVEILLANCE

Most of the existing surveillance systems struggle to detect, identify, and track targets in real-time mainly due to the latency involved in transmitting the raw data from sensors, performing the information/data fusion, computation, and analytics at a distant central platform (e.g., cloud), and sending the commands back to the actuators. Hence, there is a need to propose an architectural paradigm that has the ability to perform real-time surveillance and monitoring.

We propose **IFCIoT-Surv**—a novel integrated fog, cloud, Internet of things (IFCIoT) architecture for urban surveillance and monitoring. The proposed IFCIoT-Surv architecture is

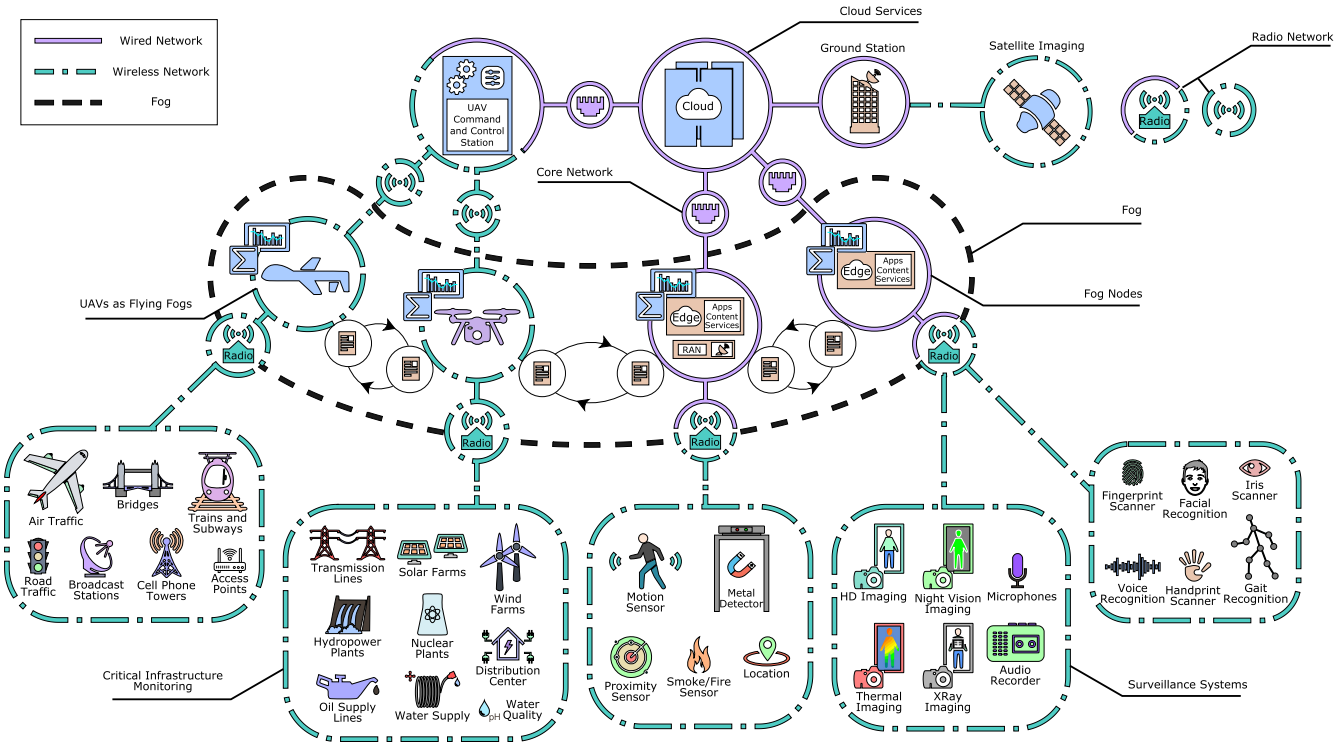


FIGURE 1. IFCIoT-Surv: a novel architecture for urban surveillance and monitoring.

depicted in Figure 1. The IFCIoT-Surv integrates surveillance and monitoring sensors/IoT devices at the edge of the network with a centralized cloud server through an intermediary layer of fog nodes. The fog enables computation near the edge of the network and helps in reducing the communication burden on the core network. The proposed IFCIoT-Surv architecture comprises of three tiers as discussed below.

The bottom-most tier of IFCIoT-Surv consists of sensors/IoT devices which sense and gather data from the physical world. To meet *surveillance* goals, data can be gathered by means of (i) biometric sensing devices, such as face detection and recognition, fingerprint scans, iris scans, gait recognition, etc., (ii) sophisticated video and imaging equipment, such as cameras including night vision imaging cameras, thermal imaging cameras, etc., and/or (iii) activity and environment sensors, such as motion detectors, proximity sensors, etc. For *critical infrastructure monitoring*, sensors/IoT devices are installed at critical infrastructures, such as electricity generation and distribution network, water supply network, telecommunication network, transportation network, etc. Furthermore, airborne vehicles equipped with sensors also sense and gather data from critical infrastructures. The multitude of sensors, IoT devices, and airborne vehicles that make up the surveillance and critical infrastructure monitoring network continuously collect massive amounts of data from the physical world. This massive data, which can be categorized as *big data* because of it characteristics [17], requires significant computing and communication resources to process, filter, fuse and analyze this data to extract meaningful and useful information. The edge-of-network sensors/IoT devices are outfitted with limited computational capabilities that are not sufficient to perform complex data operations. Furthermore, the communication overhead and interoperability issues involved in each sensor/device sharing data with its neighboring sensors/devices to get local contextual information complicates the system implementation.

In order to enhance the computational capabilities at the edge of the network and to make communication between sensors/IoT devices effective, IFCIoT-Surv includes a geo-distributed network of fog nodes. The fog nodes are also referred to as edge servers and fog computing is essentially similar to edge computing in concept with subtle differences as outlined in [18]. Both fog and edge computing shift applications, data, services, intelligence, and computation away from centralized nodes to the logical extremes of the network. While the focus of edge computing is computation on edge devices while fog computing acts as a mediator between the edge and the cloud. In fog computing, typically data is transmitted from the data source to a gateway or router for processing, and then sent back to the edge for action. Furthermore, edge infrastructure targets proprietary or limited vendor solutions, that is why, radio access network in edge computing paradigm is typically a cellular network. Fog computing is more open to heterogeneity in fog nodes or servers in the fog layer, and any server, router, and gateway node can act as a fog server in the fog layer. Also, radio access

network (RAN) in fog computing can be a wireless local area network (WLAN), WiMAX, and/or cellular. Finally, edge computing is often considered a concept that brings computational resources close to data-generating sources whereas fog computing provides the architecture and repeatable structures that help push computation power away from centralized systems or clouds to the edge of the network [19].

The proposed IFCIoT-Surv is equally applicable to edge computing paradigm as it not only encompasses computations on edge devices, but also provides intermediary fog layer connecting edge devices and cloud. The fog comprises of stationary nodes, such as RAN modules, routers, etc., augmented with additional computational hardware or mobile/flying nodes, such as UAVs (see Section IV). The stationary fog nodes can be pre-deployed at locations of interest in surveillance sites, for example, along roadside infrastructure, traffic signals, route guide signs (e.g., street signs, mile markers, and exit signs), toll plazas, buildings, gas stations, power grids, etc. The fog also partially includes the radio network interface used to communicate with the edge of network sensors/devices. Each fog node manages a cluster of edge-of-network sensors/IoT devices. Fog nodes provide applications, content, services, and storage to these edge-of-network devices. For example, fog nodes can facilitate complex image processing tasks for imaging data obtained from video and imaging equipment, information fusion tasks for activity and environment sensors, such as motion and proximity detectors and provide storage for biometric databases, such as facial recognition, fingerprint scans, etc. Since the fog is in closer proximity to the edge of the network, it can perform these tasks with lower latency and communication overhead as compared to cloud.

The intermediate fog node layer of IFCIoT-Surv is connected to the top-tier centralized cloud server layer through the core network. The core network transfers locally analyzed data and other important information from the fog to the cloud for a variety of purposes including analytics and archival. Thus, fog nodes help to tremendously reduce the volume of data communicated over the core network. The cloud facilitates a broader and global contextual scope for data analytics. For example, the cloud can analyze information collected by different fog nodes and factor in data/images from satellites, national identification (ID) and biometric databases, and national criminal databases, etc., to generate city-wide, county-wide or state-wide information maps.

IV. UAVs FOR SURVEILLANCE IN FogSurv

The increasing proliferation of UAVs and the cost advantage afforded by UAVs over conventional airborne vehicles [20] provide opportunities for using UAVs both for surveillance as well as data collection platform. This section elaborates the usage of UAVs for sensing as well as data collection in FogSurv.

The IFCIoT-Surv architecture (Figure 1) in FogSurv incorporates a multitude of sensors and UAVs. UAVs are equipped with a variety of sensors that provide the flight controller with data such as telemetry, attitude, terrain, and obstacles. The most common suite of sensors for UAVs include inertial measurement unit (IMU), global positioning system, and optical sensors [21]. The combination of sensors installed on UAVs depend on the UAV type, sophistication level, and mission. For surveillance applications, UAVs are often equipped with infrared sensors in addition to optical sensors to help monitor different activities at night. UAVs can act as *reconnaissance platforms* and capture high quality images and videos in the surveillance field. The utilization of UAVs, whether disperse or in a *swarm*, in FogSurv enables efficient and accurate generation of real-time SA maps as UAVs provide better aerial view and visibility that cannot be attained by using only ground sensors. Figure 2 depicts the role of UAVs in FogSurv for enhancing surveillance where UAVs are shown performing sensing as well as data collection.

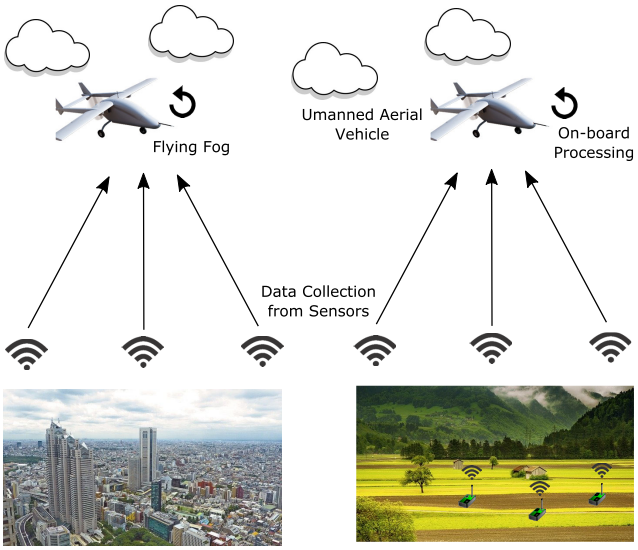


FIGURE 2. UAVs performing sensing and data collection in FogSurv.

Different types of UAVs can be deployed for surveillance and reconnaissance in FogSurv. The type of UAV deployed for surveillance depends on the area to be surveilled and the type of information to be gathered. The UAVs can be classified into (i) high altitude long endurance, (ii) medium altitude long endurance, (iii) low altitude long endurance, and (iv) low altitude short endurance. The U.S. Department of Defense (DoD) classifies UAVs into five groups based on their maximum takeoff weight, operating altitude, and speed [22]. In the DoD UAV classification, the group 1 UAVs are the lowest capability UAVs and the group 5 UAVs are the highest capability UAVs. The group 1 UAVs have the maximum takeoff weight capability of less than 20 lb, operating altitude of less than 1200 ft, and speed of less than 100 knots, whereas group 5 UAVs have the maximum takeoff weight ability of greater than 1320 lb, operating altitude of greater than 18000 ft, and can have any speed (typically greater than 200 knots, and currently can attain maximum speeds of around 14,660 knots). An example of a group 5 high altitude

long endurance surveillance UAV is the Lockheed Martin RQ-170 Sentinel [23], which is a stealth UAV for carrying out surveillance, reconnaissance, and strike missions. The RQ-170 Sentinel is capable of flying at an altitude of 50,000 ft and can capture real-time imagery (e.g., of the battlefield), and then transfer that imagery to the ground control station via a line of sight communication data link.

The use of low altitude *UAVs for data collection* in FogSurv is vital as discussed in the following. The IFCIoT-Surv architecture incorporates a multitude of sensors for data collection as depicted in Figure 1. Sensors, often being inexpensive, can be deployed in remote areas for monitoring; however, communication of this sensed data to a central server or cloud for processing and analytics is a challenge because of poor or often no network connectivity in remote areas. Furthermore, many sensors are battery-powered with limited energy reserves, and attempting to communicate over long distances can rapidly deplete their available battery reserves. Nevertheless, the data needs to be collected from these remote sensors for various tasks, such as information fusion, processing, analytics, and SA enhancement. In IFCIoT-Surv architecture, low altitude UAVs can be utilized as *flying fogs*, that is, a fog node that is aerial, and can help in data collection from other sensing nodes. In the flying fog role of UAVs, UAVs collect data from sensors/IoT devices when flying over those sensors and IoT devices as depicted in Figure 2. Since the distance between flying UAVs and ground sensors/IoT devices is relatively small, communication over this small distance does not consume large energy, and thus low altitude UAVs as data collection platforms help in conserving the battery of in-field sensors and IoT nodes. To integrate UAV as flying fogs in FogSurv, protocols need to be developed for dynamic joining and exiting of UAVs in the fog layer.

The data sensed and collected from the UAVs can be utilized for on-board processing, data fusion, analytics, or for transferring data to other fog nodes or cloud as shown in Figure 2. The increasing availability of on-board data and computing power offers new opportunities for real-time and on-board decision-making in UAVs [24]. The appropriate role of a UAV in FogSurv, such as data collection or on-board processing and decision-making, can be assigned based on the capability of the UAV. Furthermore, DDDAS methods can be leveraged in FogSurv for enabling UAVs to adapt their sensing, processing, and routing dynamically based on live sensor data to better meet mission objectives. For example, DDDAS-driven UAVs can adapt their flight path based on the movement of targets in the surveillance field.

The raw data and video streams obtained by IoT nodes and UAVs in FogSurv are also sent back to the ground controller and displayed on a smart device of a human operator. The human operator can be police or law-enforcement officials, and the smart device on which the results are displayed can be a laptop in the police car or a work station in the office of law-enforcement agencies. If any suspicious objects are observed in the received data, the objects are identified, locked, and tracked immediately along with their speed calculation. In case of identification of several suspicious objects, tracking tasks can be assigned to various fog nodes including UAVs acting as flying fogs.

In short, UAVs play a pivotal role in FogSurv as they provide several indispensable features, such as capturing aerial views of targets that cannot be obtained through ground sensors, serving as data collection platform for ground sensors, and acting as flying fog nodes. However, integration of UAVs for surveillance also pose various challenges. These challenges include high cost for surveillance and reconnaissance UAVs, communication reliability, security, privacy, simultaneous localization and mapping (SLAM) in GPS-denied environments for autonomous UAVs, and the collision avoidance between swarm of UAVs. Addressing these UAV challenges is beyond the scope of this article.

V. FRAMEWORK FOR AI PROCESSING IN FogSurv

Human-operated surveillance systems have been used for a long time for monitoring small places such as banks, office buildings, hotels, and casinos. Urban surveillance, however, requires monitoring of a town/city wide area with a large number of objects (e.g., people, vehicles, etc.) moving around in sharp angles and tight corners with non-uniform speed and acceleration. To assist with wide area monitoring, urban surveillance systems incorporate a multitude of sensors (e.g., motion detectors, proximity sensors, acoustic sensors, and cameras) and UAVs that gather an enormous amount of data. Conducting surveillance at a large scale with human operators to monitor and make sense of this data is infeasible because of the sheer volume of data that the operators have to be aware of to make informed control decisions. To obtain meaningful insights from this massive data, large-scale surveillance systems must be autonomous requiring minimal human input and supervision. Recent advances in AI and ML can help assist with large scale automated surveillance. This section proposes a framework for AI/ML processing in FogSurv and elaborates how AI/ML methods are integrated in FogSurv for enabling real-time urban surveillance.

A variety of ML methods [25], including supervised learning (e.g., decision trees, rule-based classifiers, Bayesian classification, support vector machines (SVM), etc.), unsupervised learning (e.g., k-means clustering, hidden Markov model, principal component analysis, etc.), and reinforcement learning can be integrated in FogSurv to assist with surveillance tasks. FogSurv can also assimilate deep learning based methods to aid with surveillance-related tasks. For example, convolutional neural networks (CNNs) can be utilized for object detection and recognition. Similarly, recurrent neural networks (RNNs) can be utilized for multi-target tracking. Table 1 provides a categorization of contemporary ML techniques that can be utilized in FogSurv framework and are evaluated in our experimental results (Section IX) for FogSurv. Since FogSurv is an architectural paradigm, FogSurv is indifferent to specific AI/ML algorithms utilized to aid with surveillance tasks. Hence, irrespective of any particular AI/ML method, FogSurv provides

TABLE 1. Categorization of contemporary ML techniques.

Technique	Category	Advantages	Disadvantages
Data Sort	Statistical	Aligns information based on importance	Obtains results only from available observations
Running Statistics	Statistical	Minimizes uncertainty based on sensor observations	Requires sensor calibration for uncertainty which does not capture domain knowledge
Boltzmann Machine	Stochastic	Determines latent variables efficiently through semi-supervised generative models	Involves many computations to discern unobserved relations
Bayes Network	Stochastic	Builds hierarchy of analysis based on conditional relations	Needs a prior modeling of numerous conditions
Support Vector Machine (SVM) Rank	Supervised Learning	Utilizes polynomial kernels to minimize data classification error	Controlling decision boundaries is difficult in nonlinear systems
SVM Pegasos	Supervised Learning	Solves decision boundaries with gradient descent effectively	Obtaining the dimensions challenging for streaming analysis
K-means Clustering	Unsupervised Learning	Groups data/parameters based on relations	Leads to suboptimal splits which necessitates expectation maximization (EM) methods
Cost Optimization	Unsupervised Learning	Controls uncertainty based on desired goals	Requires domain modeling for cost function
Deep Learning	Supervised/Semi-Supervised/ Unsupervised Learning	Automatically extracts higher level features from the raw input	Requires large amount of data and can have long training time

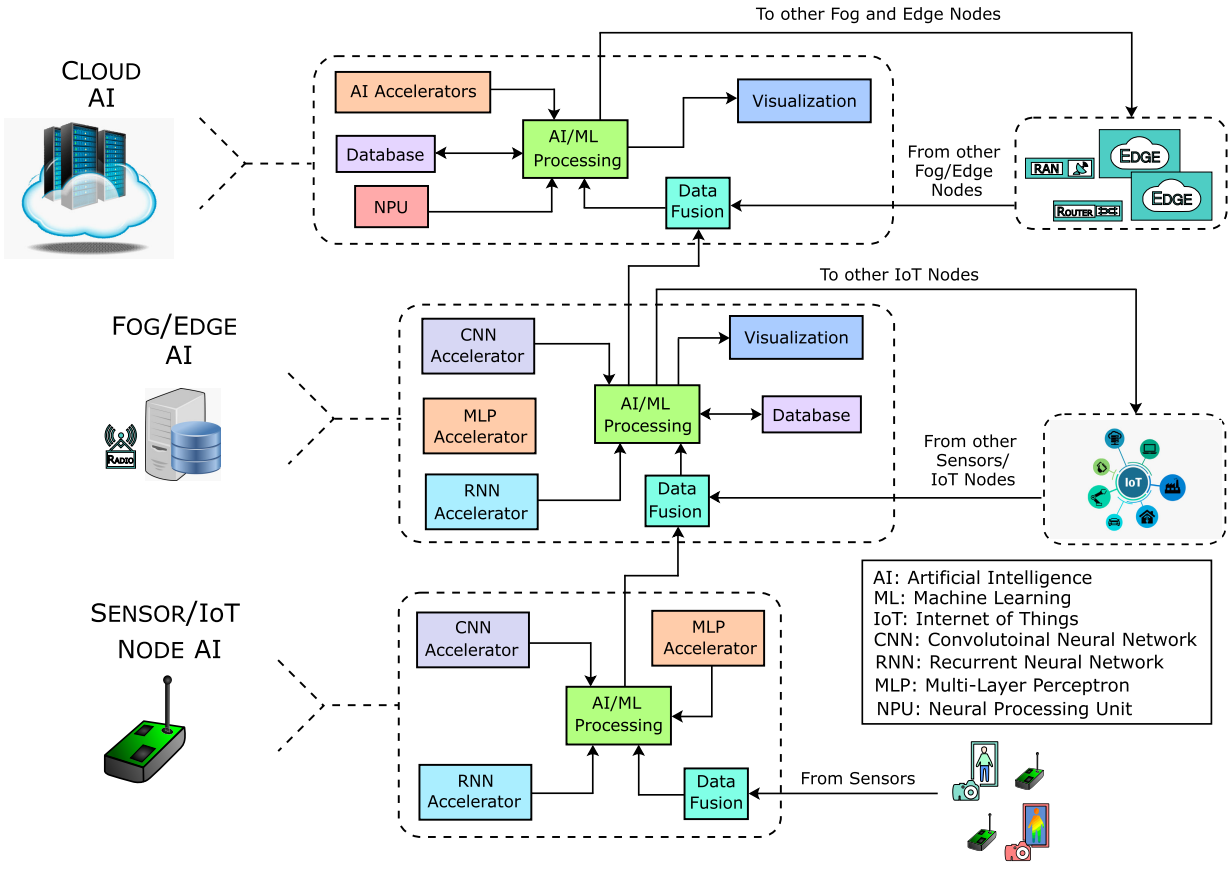


FIGURE 3. Framework for AI/ML processing in FogSurv.

a performance and energy-efficient framework for execution of AI/ML algorithms as depicted in Figure 3 and discussed below.

FogSurv alleviates the computational burden of the cloud by distributing AI/ML processing, analytics and decision-making among all three levels of the IFCIoT-Surv:

IoT, fog, and the cloud, as illustrated in Figure 3. The process of AI/ML processing, data analytics and decision-making in FogSurv is initially performed at the bottom-most tier comprising of sensor/IoT nodes. Each IoT node as well as each UAV flying over the local surveilled field generate lightweight pre-processing and analytics based on their local measurements. These bottom-most tier IoT nodes can be outfitted with some degree of computation and communication capabilities to perform in situ processing (e.g., IoT nodes with cameras can execute lightweight algorithms for object detection, classification, tracking, etc.). As Figure 3 shows, some of the IoT nodes can be equipped with different ML accelerators, such as CNN accelerator, RNN accelerator, and multi-layer perceptron (MLP) accelerator, to expedite the processing of ML algorithms in an energy-efficient manner. The design and development of deep learning accelerators for IoT and edge devices is an active area of research, and many such accelerators are being contemplated [26]–[28]. The in-situ processing of sensed data within IoT nodes, aided by various accelerators, brings about huge savings in communication bandwidth as it eliminates the overhead involved in communicating raw video data over the network. In FogSurv, features description can be forwarded between multiple sensors (e.g., cameras) tracking the same object in the scene to greatly reduce the amount of (image) processing computation that needs to be carried out at each IoT/sensor (camera) unit. Moreover, for deep learning approaches, these IoT devices can perform partial processing of deep neural network (DNN) layers. These on-field sensors and IoT nodes are connected to a distributed network of fog nodes, the middle tier of the IFCIoT-Surv architecture. The sensors and IoT nodes share their sensed data and local analytics with fog nodes. For deep learning approaches, these IoT nodes provide their processed output to the fog nodes so that the fog nodes or cloud can process the remaining/additional DNN layers to provide a labeled decision.

The fog nodes extend the computation and communication capabilities of the on-field sensors/IoT nodes. Fog nodes assist and coordinate ML tasks such as object detection, recognition, and information fusion operations between multiple IoT nodes that are part of the surveillance system. To speedup the execution of various ML algorithms, fog nodes can be equipped with various accelerators, such as CNN, RNN, and MLP accelerators as depicted in Figure 3. The fog nodes report processed, sanitized, and analyzed data back to the central platform (cloud), the top-most tier of the architecture. The cloud fuses the results generated by all fog nodes. Furthermore, cloud provides AI processing and analytics on a wide scale while supplementing the information with centrally acquired information from other sources, such as satellite and aerial imagery. To assist with AI processing, cloud is often equipped with a variety of AI accelerators and neural processing units (NPUs) as shown in Figure 3. The results of the AI/ML processing and data analytics by the cloud are shared with fogs and in turn with the sensing/IoT platforms to adjust their local insights, analytics, and decision-making. Such three-tiered distribution of computational load not only lowers the cost of bandwidth and latency in obtaining AI/ML processing, analytics and decision-making, but also provides a greater degree of resilience and autonomy by decreasing the reliance of analytics on the cloud. Fog nodes and cloud in FogSurv also store previous observations and the effect of corresponding actions in a database to help improve AI processing, analytics, and decision-making for future observations. FogSurv further leverages DDDAS in tiered AI processing. For example, DDDAS methods can be utilized in FogSurv to control the positioning of cameras based on the processed/fused data to improve multi-target tracking in surveillance applications.

VI. AI- AND DYNAMIC DATA-DRIVEN INFORMATION FUSION IN FogSurv

Section V discusses a framework for AI/ML processing in FogSurv. However, we note that AI processing on data acquired from a single source may not provide accurate results and actionable intelligence. For instance, data obtained from a single camera setup may suffer from a multitude of problems, such as occlusions, change in ambient lighting conditions, and/or chaotic elements (elements that confuse ML algorithms) in the camera's field of view. The use of different sensors (e.g., electro-optical and infrared sensors) affords robustness as anomalies or noise appear different to different sensors. For example, it is highly improbable that noise or anomalies from one image source (e.g., color camera) is the same as from another source (e.g., infrared camera). Hence, to overcome the shortcomings arising from processing on single sensor data, data from different sources need to be fused. Additionally, information fusion minimizes the information redundancy between data, such as views captured by different cameras or quantities measured by different sensors located in proximity to each other. Furthermore, to obtain meaningful insights from the surveillance data, information fusion must also be integrated with AI/ML methods. This section discusses an information fusion model that integrates AI/ML to enable real-time urban surveillance. The novelty of our proposed information fusion model lies in (i) exploiting fog computing for hierarchical information fusion; and (ii) integration of ML and DDDAS methods in information fusion models to steer the sensor mode and measurement process in accordance with environmental conditions, contextual information, and mission objectives.

Although processing of sensor data for information fusion can be centralized, there is a need for distributed information fusion for enabling real-time information fusion [29], [30]. We propose a *hierarchical information fusion* approach in FogSurv with reference to the IFCIoT-Surv architecture. In FogSurv, information fusion is performed at all levels, that is, IoT, fog, and cloud as depicted in Figure 3. The lowest tier/hierarchy comprising of sensors and IoT nodes perform information fusion at the IoT node-level to minimize the redundancy in the raw information captured by the sensors. The IoT nodes send only fused and sanitized

information to the middle tier comprising of fog nodes instead of sending the raw data thus greatly reducing the communication burden on the network and enormously conserving energy of the energy-constrained sensors/IoT nodes. The fog nodes perform information fusion on the data received from multiple IoT nodes/clusters. The fog nodes also help in determining topological relationships between the sensors (e.g., on-field cameras) and fuse the topological, contextual, and environmental information with the data. The fog nodes report processed, sanitized, fused and analyzed data back to the central platform (cloud), the top-most tier of the architecture, which performs information fusion and high-level advanced analytics functions on data received from multiple fog nodes to determine the global state of surveillance and monitoring.

Our proposed information fusion model adapts DFIG (Data Fusion Information Group) model [31] to integrate support of DDDAS and ML modules. We refer to the extended DFIG model as ML- and DDDAS-inspired DFIG (MD2FIG) model. The MD2FIG model is depicted in Figure 4 and is elaborated below.

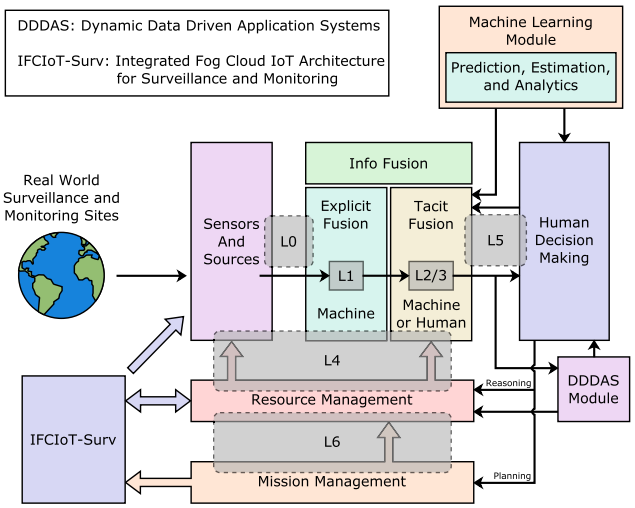


FIGURE 4. MD2FIG model for information fusion.

In the MD2FIG model (as in the DFIG model), the levels (L) determine the processing in the system, such as L0–data registration, L1–object tracking and identification assessment, L2–situation awareness, L3–impact assessment, L4–sensor management, L5–user refinement, and L6–mission management. In our model, the *low-level* information fusion processes support target classification, identification and tracking whereas *high-level* information fusion processes support situation, impact, and function process refinement. The low-level information fusion deals with numerical data, such as locations, kinematics, and attribute target types, whereas high-level information fusion deals with abstract symbolic information, such as threat, intent, and mission objectives [32].

We propose to leverage DDDAS methods in the DFIG model in order to provide adaptation to environmental conditions and steer the measurement (both ground sensors and UAVs) and information fusion process for improved recognition, identification, and target tracking. The MD2FIG model also exploits DDDAS methods to effectively allocate computational resources in the IoT and fog nodes by matching the expected entropies of localized information content as a function of multidimensional space that the observations span [31]. For example, in case of surveillance application, presence of objects of interest in a particular region would require high resolution over those regions and the regions with no activity of interest would accept coarser computations without a conspicuous degradation in surveilled artifacts.

The MD2FIG model further integrates ML approaches with information fusion as ML can be particularly useful for HLIF (level 2 to level 6). AI/ML combined with information fusion can help provide a variety of insights from the sensed data, such as SA, behavior recognition, intent recognition, threat assessment, and decision-making. The integration of DDDAS and ML helps the MD2FIG model in selecting the appropriate sensor mode (e.g., visible light, hyper-spectral, infrared, night vision, etc.) to deliver the right information in response to a query. When the user requests object information (L1) within a situation (L2), MD2FIG model considers the environmental conditions such as time of operation (e.g., day or night), weather conditions (e.g., cloudy, rainy, sunny, foggy), and geographical information (e.g., terrain type: urban, rural, mountainous) to select the appropriate sensor mode and provide the optimal response. The combination of ML and DDDAS methods in MD2FIG model enables mission coordination and cooperative sensing of IoT nodes and UAVs resulting in enhanced SA.

VII. LATENCY MODELING FOR AI AND INFORMATION FUSION IN FogSurv

This section models the latency for AI and information fusion in FogSurv. This modeling helps in understanding the constituents of latency in AI and information processing in FogSurv and also help analyze and explain the experimental results. As depicted in Figure 3, AI and information fusion in FogSurv can be done at three tiers: (i) IoT device, (ii) fog node, and (iii) cloud. In this section, we analyze the average latency for AI and information fusion in FogSurv with and without computation offloading. Average latency of execution in IoT device T^{IoT} without computation offloading to fog or cloud can be given as

$$T^{IoT} = T_p^{IoT} + T_q^{IoT}, \tag{1}$$

where T_p^{IoT} denotes average processing latency at the IoT node and T_q^{IoT} denotes average queueing delay for the tasks at the IoT node. T_p^{IoT} for FogSurv can be written as

$$T_p^{IoT} = T_{p-IF} + T_{p-AI} + T_{p-o}, \tag{2}$$

where T_{p-IF} denotes the processing latency for information fusion tasks at the IoT node, T_{p-AI} denotes the AI processing

latency, and T_{p-o} denotes the processing latency for other tasks (i.e., other than AI and information fusion) at the IoT node. T_p^{IoT} can be calculated as [33]

$$T_p^{IoT} = \frac{I_c \times CPI}{f_c}, \tag{3}$$

where I_c represents instruction count for AI, information fusion, and other tasks at the IoT node, *CPI* represents average clock cycles per instruction, and f_c represents the clock rate of the IoT processor on which the instructions are being executed. T_q^{IoT} depends on the number of jobs in the IoT device queue and can take the form of G/G/m/k queue for general task arrival rate and general service time distributions, m represents the number of processor cores in the IoT device, and k is the buffer size to hold jobs. To obtain simplified closed-form mathematical expressions, G/G/1 queues can be assumed [34] and G/G/m model can be constructed from the network of G/G/1 queues [35]. The expected packet queueing delay at the IoT node using the G/G/1 model can be given as:

$$E\left[T_q^{IoT}\right] = \frac{\lambda_{e-IoT}^2(\sigma_{u-IoT}^2 + \sigma_{v-IoT}^2) + (1-\rho_{IoT})^2}{2\lambda_{e-IoT}(1-\rho_{IoT})} - \frac{\nu_{h-IoT}^{(2)}}{2\nu_{h-IoT}} \quad : \quad \rho_{IoT} < 1 \tag{4}$$

where λ_{e-IoT} denotes the effective arrival rate at the IoT device queue, $\rho_{IoT} = \lambda_{e-IoT}/\mu_{IoT}$ denotes the IoT device queue's utilization (μ_{IoT} denotes the IoT processor service rate), σ_{u-IoT}^2 denotes the variance of tasks inter-arrival times at the IoT device queue, and ν_{h-IoT} and ν_{h-IoT}^2 denote the first and second moments of the IoT device queue *idle period* I_{IoT}, respectively. The expected value of the IoT queue idle period (the period of time when there are no tasks/jobs in the queue) can be given as

$$E[I_{IoT}] = \frac{1-\rho_{IoT}}{\lambda_{e-IoT} \times p_{0-IoT}} \quad : \quad \rho_{IoT} < 1 \tag{5}$$

where p_{0-IoT} is the probability that a task arrives when the IoT device queue is empty. The expected packet queueing delays for fog node and cloud can be written similar to Eq. (4).

Average latency of execution at fog node T^{Fog} can be given as

$$T^{Fog} = T_t^{IoT-Fog} + T_p^{Fog} + T_q^{Fog}, \tag{6}$$

where $T_t^{IoT-Fog}$ denotes the transmission latency from an IoT device to fog node. T_p^{Fog} represents the average processing latency at the fog node, which comprises of latency for information fusion processing, AI processing, and other tasks' processing, and can be written similar to Eq. (2). T_q^{Fog} denotes average queueing delay for the tasks at the fog node. Since a fog node receives computation offloading requests from multiple IoT nodes in its vicinity, T_q^{Fog} captures the queueing delay due to offloading requests/tasks from other IoT devices in the vicinity. T_q^{Fog} can be expressed similar to Eq. (4) and Eq. (5). $T_t^{IoT-Fog}$ depends on the data shared between the IoT device and the fog node (where the IoT device offloading computations to), and the communication bandwidth between the IoT device and the fog node, and can be expressed as

$$T_t^{IoT-Fog} = \frac{D^{IoT-Fog}}{BW^{IoT-Fog}}, \tag{7}$$

where $D^{IoT-Fog}$ symbolizes the data shared (sent and received) between the IoT device and the fog node for computation offloading from the IoT device to the fog node as well as sending of results data back to the IoT node from the fog node, and $BW^{IoT-Fog}$ stands for the available communication bandwidth between the IoT device and the fog node. We note that T^{Fog} should be less than $T^{IoT} + T_t^{IoT-Fog}$ to justify computation offloading from an IoT device to a fog node, that is, $T^{Fog} < T^{IoT} + T_t^{IoT-Fog}$.

Average latency of execution at cloud T^{Cloud} can be given as

$$T^{Cloud} = T_t^{Cloud} + T_p^{Cloud} + T_q^{Cloud}, \tag{8}$$

where T_t^{Cloud} denotes the transmission latency for offloading computations to cloud, T_p^{Cloud} represents the average processing latency at the cloud, which comprises of latency for information fusion processing, AI processing, and other tasks' processing, and can be written similar to Eq. (2). T_q^{Cloud} denotes average queueing delay for the tasks at the cloud and can be expressed similar to Eq. (4) and Eq. (5). There are two cases for T_t^{Cloud}, which are given below

Case 1: The computation is directly offloaded from an IoT device to cloud and thus

$$T_t^{Cloud} = T_t^{IoT-Cloud} = \frac{D^{IoT-Cloud}}{BW^{IoT-Cloud}}, \tag{9}$$

where $T_t^{IoT-Cloud}$ stands for the transmission latency for offloading computations from the IoT device to the cloud and receiving the results back from the cloud, $D^{IoT-Cloud}$ denotes the data shared (sent and received) between the IoT device and the cloud for computation offloading, and $BW^{IoT-Cloud}$ stands for the available communication bandwidth between the IoT device and the cloud.

Case 2: The computations are offloaded hierarchically, that is, from IoT device to fog, and from fog to cloud, as depicted in Figure 3, and thus

$$T_t^{Cloud} = T_t^{IoT-Fog} + T_t^{Fog-Cloud}, \tag{10}$$

where $T_t^{IoT-Fog}$ stands for the transmission latency for offloading computations from the IoT device to the fog node, and $T_t^{Fog-Cloud}$ denotes the transmission latency for computation offloading from fog to cloud. T_t^{Cloud} for this case can be given as

$$T_t^{Cloud} = \frac{D^{IoT-Fog}}{BW^{IoT-Fog}} + \frac{D^{Fog-Cloud}}{BW^{Fog-Cloud}}, \tag{11}$$

where $D^{Fog-Cloud}$ denotes the data shared (sent and received) between the fog node and the cloud for computation offloading, and $BW^{Fog-Cloud}$ stands for the available communication bandwidth between the fog node and the cloud.

TABLE 2. Summary of FogSurv use cases.

Use Case	Significance	Surveillance	Surveilled Data
Battlefield applications	(i) Supporting soldiers (ii) Battlefield control and command (iii) Enhanced situational awareness (iv) Support in asymmetric warfare	(i) Soldier's psychophysical and emotional conditions (ii) Battlefield status (iii) Enemy personnel, equipment, and sites	(i) Biometrics (e.g., face, iris, heart rate, gait, gestures, and facial expressions) (ii) Environmental conditions/parameters (iii) Visual information from cameras
Assistance in gray zone warfare	(i) Assistance to soldiers (ii) Mitigating negative effects of gray zone warfare (iii) Support long-term analysis	(i) Gray zone actors (ii) Communication channels (iii) Critical infrastructures (iii) Public places and tourist attractions	(i) Biometrics (e.g., face, gestures) (ii) Social media (iii) Environmental conditions/parameters (iii) Visual information from cameras
Homeland security and defense	(i) Damage control from natural disasters (ii) Counter-terrorism (iii) Preventing smuggling (iii) Support long-term analysis	(i) Environment (ii) Critical infrastructures (iii) Airports, sea ports, national borders (iii) Public places and tourist attractions	(i) Biometrics (e.g., face, finger prints) (ii) X-ray scanning of transported goods (iii) Environmental conditions/parameters (iii) Visual information from cameras
Monitoring and early warning systems	(i) Damage control from natural disasters (ii) Loss control from man-made disasters (iii) Fast emergency response (iii) Safety enhancement by early alerts	(i) Environment (air, soil, water) (ii) Weather (e.g., tornadoes, hurricanes) (iii) Critical infrastructures (iii) Critical sites (e.g., volcanoes, forests)	(i) Environment sensors (ii) Satellite imagery (iii) Radiation monitoring instruments (iii) Visual information from cameras

In case of hierarchical computation, overall latency of execution T can be given as

$$T = T^{IoT'} + T^{Fog'} + T^{Cloud'}, \tag{12}$$

where $T^{IoT'}$ represents the computation execution time at IoT device and the execution time to process any data/results received from the fog node. $T^{Fog'}$ stands for the computation execution time at the fog node and the execution time to process any data/results received from the cloud. $T^{Fog'}$ includes transmission, processing, and queuing latency, and can be written similar to Eq. (6). $T^{Cloud'}$ signifies the computation execution time at cloud for the data offloaded from the fog node. $T^{Cloud'}$ includes transmission, processing, and queuing latency, and can be written similar to Eq. (8). We clarify that different notations, that is, $T^{IoT'}$, $T^{Fog'}$, and $T^{Cloud'}$, are used in Eq. (12) instead of T^{IoT}, T^{Fog}, and T^{Cloud} to indicate that in addition to in-node computations, computations are also offloaded to the next level in the hierarchy, that is from an IoT device to a fog node, and from the IoT device/fog node to the cloud. For compute-intensive tasks, hierarchical execution latency should be less than the execution time on the IoT device alone, that is, $T \ll T^{IoT}$, or at least $T < T^{IoT}$ to justify computation offloading from an IoT device to fog and cloud.

VIII. FogSurv USE CASES

The proposed FogSurv can have a huge impact on multifarious fronts of national significance. FogSurv can provide technological and strategic advantages for armed forces, safeguard national security, bolster homeland security and defense, apprehend fugitives, improve counter terrorism capabilities, assist anti-smuggling efforts, and monitor critical infrastructures and early warning systems. This section discusses some of the use cases for FogSurv. Table 2 summarizes the discussed use cases for FogSurv along with their significance and data that needs to be collected for enabling surveillance.

A. BATTLEFIELD APPLICATIONS

FogSurv can provide significant advantages for battlefield applications as discussed below.

1) ENHANCED BATTLEFIELD SITUATIONAL AWARENESS

An increasing number of ubiquitous sensing and computing devices are worn by military personnel and are embedded within military equipment (e.g., combat suits, instrumented helmets, weapon systems, etc.), which in FogSurv framework can be referred as Internet of military things (IoMT) or Internet of battlefield things (IoBT) [36]. The IoMT/IoBT are capable of acquiring a variety of static and dynamic biometrics (e.g., face, iris, periocular, fingerprints, heart-rate, gait, gestures, and facial expressions), which can be used to carry out context-adaptive continuous monitoring of soldier's psychophysical and emotional conditions on the field with the help of IFCIoT-Surv architecture. Information fusion and big data analytics in FogSurv framework can then be applied to the collected data for SA, situation evaluation, decisional activity, and providing real-time support to soldiers in the battlefield.

2) BATTLEFIELD COMMAND AND CONTROL SYSTEM

FogSurv can be used to achieve more informed and reliable battlefield command and control system. FogSurv can enable the armed forces to fully exploit the information gathered by a wide set of heterogeneous IoBT/IoMT devices deployed in the battlefield and can provide the armed forces a strategical

advantage over adversaries. FogSurv can help soldiers identify the enemy, and access devices and weapons systems under low-latency edge/fog computing. FogSurv can also help the armed forces to provide close air support to soldiers. Using biometrics, environmental sensors, and other connected IoMT/IoBT devices to send and receive data quickly as enabled by FogSurv will permit military personnel to timely respond to potentially dangerous situations on the battlefield and improve command and control operations.

3) ASYMMETRIC WARFARE

FogSurv can be particularly useful in asymmetric warfare and battlefields where it is not always straightforward to identify enemy combatants (e.g., the enemies can appear as civilians or access restricted military bases with a stolen badge). In the FogSurv framework, sensors can scan irises, fingerprints, and other biometric data to identify individuals who might pose a danger. The use of DDDAS methods in FogSurv can then help in targeted collection of data on identified individuals and can also mobilize the responding units (e.g., snipers) to diffuse the threat.

B. ASSISTANCE IN GRAY ZONE WARFARE

FogSurv can provide assistance to armed forces in *gray zone* warfare and conflicts [37] where the gray zone actors use techniques such as misinformation and disruptions of services and critical infrastructures to destabilize nations and thus create favorable conditions for military engagements. FogSurv can provide real-time monitoring of sites and potential gray zone actors to help prevent or mitigate negative effects of gray zone actors, such as misinformation and disruptions of services and critical infrastructures. Specifically, FogSurv can serve as a practical framework for long-term analysis and mitigation of adversarial strategies in gray zone conflicts by providing event detection, adversarial intent estimation, policy optimization, and model adjustment by using the network of sensors and IoT devices including UAVs, MD2FIG information fusion model, and hierarchical AI/ML processing.

C. HOMELAND SECURITY AND DEFENSE

FogSurv architecture and framework can be very advantageous for enhancing homeland security and defense. Real-time surveillance and monitoring enabled by FogSurv can help in providing timely detection and response against both natural disasters (e.g., hurricanes, floods) and man-made events (e.g., terrorism). Furthermore, FogSurv can assist the law enforcement officials in apprehending fugitives, counter terrorism, and anti-smuggling efforts.

D. MONITORING AND EARLY WARNING SYSTEMS

FogSurv can help enable efficient and real-time monitoring of critical infrastructures, such as bridges, electricity generation and distribution network, water supply network, telecommunication network, transportation network, etc. For example, traffic monitoring and accurate detection of suspicious vehicles can result in reduced crimes, accidents, and fatalities. FogSurv can also serve as an integral part of early warning systems and can help in safety enhancement and sustainability of cities by facilitating quick detection and response to emergency situations, such as fire, flood, earth quakes, volcanic eruption, and tornadoes, etc.

IX. RESULTS AND ANALYSIS

In this section, we provide results and analysis for our FogSurv framework. We have conducted three sets of experiments targeting the following objectives:

1) Evaluation of latency or timeliness for FogSurv versus a cloud-based system.
2) Evaluation of latency and accuracy for combined data fusion and AI in FogSurv.
3) Evaluation of average precision and latency for combined multimodal data fusion and AI for UAVs in FogSurv.

In the following, we describe the experimental setups, results and analysis for the three set of experiments.

A. AVERAGE LATENCY FOR FogSurv VERSUS CLOUD

Our first set of experimental results focus on providing a comparison of FogSurv versus a cloud-based system following a similar approach as presented in our earlier work [12]. The comparison focuses on average latency metric which is the time taken by FogSurv or a cloud-based system to complete and return the results of the tasks offloaded by an IoT device. These tasks could include image analysis by FogSurv or cloud for an image captured by an IoT device for object detection and recognition or could include information fusion of sensor data recorded by IoT devices according to our proposed MD2FIG model (Figure 4). Through these results, we intend to validate that FogSurv provides lower latency for computation offloaded tasks as compared to only cloud based surveillance systems. Our experiments primarily focus on latency metric as timeliness or latency is of foremost significance for real-time surveillance.

A.1 EXPERIMENTAL SETUP

For our experiments, we run ML benchmarks summarized in Table 1 of varying complexity and sizes from the *dlib* library [38]. These benchmarks loosely represent the type of data and computations that are performed by IoT and/or fog nodes. Figure 5 depicts the normalized execution time results (i.e., normalized to execution time of a benchmark on the IoT device) that have been derived from our earlier experiments on computation offloading [12]. Figure 5 depicts (in brackets) the size of data transfer, which is the sum of the sent and received data, between client (IoT node) and the server (fog node or cloud) for each benchmark. The data transfer size from a client to a server is proportional to the workload size.

We conduct our experiments for two scenarios. In the first scenario, an IoT node offloads tasks to a nearby fog node in FogSurv. In the second scenario, the IoT node offloads the

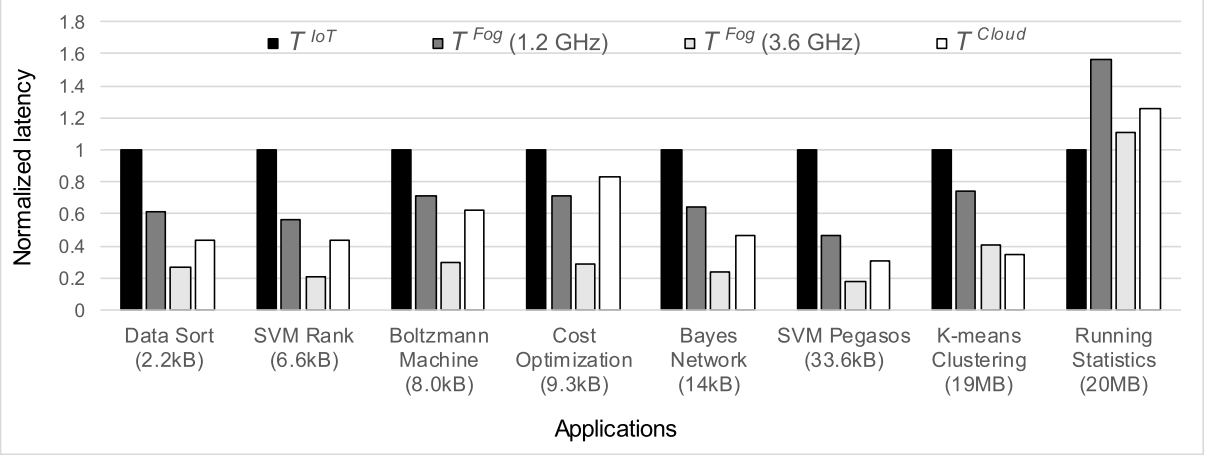


FIGURE 5. Comparison of benchmarks' normalized execution times on an IoT device versus the normalized execution times on a fog node and a cloud server including the computation offloading overhead from the IoT device to fog/cloud.

same tasks to a remote cloud server. For client IoT device, we use a Linux virtual machine configured to a single core running at 800 MHz and having 2 GB of random access memory (RAM). For a fog node in FogSurv, we obtain results for an eight (8) core Intel processor running at frequencies ranging from 0.8 GHz to 3.6 GHz. For a cloud server, we use Amazon Elastic Compute Cloud (EC2) located in western US (Northern California) region and running at 2.4 GHz [12], [39]. The client IoT device communicates with the fog node in FogSurv through a wireless ad-hoc network. The client IoT device communicates with the cloud over the Internet. We use mean execution time and mean round trip time values (averaged over ten (10) runs) to determine the execution time (without offloading) and the average latency resulting from computation offloading to fog/cloud, respectively. In our experiments, since we have dedicated machines for IoT and fog nodes, and reserved resources for cloud (Amazon EC2), queueing delay is negligible (the only queueing delay that our benchmarks can experience is due to some background system tasks that are run by the nodes periodically), however, processing and transmission times are non-negligible and dominate the normalized execution time presented in Figure 5. We note that in real-world urban surveillance where fog and cloud need to process data from multiple IoT devices, queueing delay will be non-negligible as captured by our latency model for FogSurv in Section VII.

A.2 AVERAGE APPLICATION LATENCY FOR FOG AND CLOUD

Figure 5 depicts the comparison of normalized execution time of the benchmarks when the benchmarks are run on the IoT device locally (i.e., without offloading) T^{IoT} versus the normalized average latency T^{Fog} to execute the tasks on the fog node including send and receive times when the IoT device offloads computation to the fog node in FogSurv. Figure 5 also indicates the normalized average latency T^{Cloud} when the IoT device offloads the tasks to the cloud. Results in Figure 5 reveal that the normalized average latency of applications offloaded to the fog node in FogSurv by an IoT device is much lesser than the normalized average latency when the tasks are offloaded to the cloud by the IoT device. This observation applies to most of the benchmarks in our test suite. For example, Bayes network benchmark exhibits an improvement in average latency of 49.6% when the benchmark is offloaded to a fog node in FogSurv by the IoT device as compared to offloading the benchmark to the cloud. Experiments show that offloading tasks to the fog nodes in FogSurv for all the benchmarks listed in Figure 5 results in an overall improvement of 37% over offloading tasks to the cloud. These results can be intuitively explained as the fog nodes in FogSurv are in close proximity to IoT devices as compared to the cloud and often high bandwidth is available for communication between IoT devices and fog nodes as compared to the cloud. We note that there are some aberrations for running statistics benchmark where offloading to fog and/or cloud results in increase in execution time as indicated by higher bar lines for normalized execution times on fog and cloud in Figure 5. This aberration can be explained by our latency model (Eq. (6), Eq. (7), Eq. (8), and Eq. (9)), where latency to transmit data to fog and/or cloud overshadows the improvement in processing time at fog/cloud.

A.3 EFFECT OF OPERATING FREQUENCY OF FOG NODES ON AVERAGE LATENCY

Since FogSurv comprises of heterogeneous fog nodes, which can be running on different processor frequencies, we obtain results for different operating frequencies of the fog node. In our experiments, the frequency of fog nodes/servers is varied between 1.20 GHz to 3.60 GHz in steps of 0.4 GHz. Results reveal that offloaded task's latency decreases as the operating frequency of the fog node increases. For example, average latency for Boltzmann machine benchmark decreases by 58% as the operating frequency of the fog node processor is varied from 1.20 GHz to 3.60 GHz. We also observe that average latency of the offloaded applications from IoT

devices does not decrease linearly with the increasing processor frequency of fog nodes. This observation can be explained as the average latency not only depends on the fog node processor's operating frequency but also on several other parameters including number of active processor cores, network bandwidth, cache and memory size.

B. AVERAGE LATENCY AND ACCURACY FOR COMBINED DATA FUSION AND AI IN FogSurv

Our second set of experimental results focus on evaluating the effectiveness of combining data fusion with AI. More specifically, the experimental results in this section present latency and accuracy comparisons between CNN processing with and without data fusion.

B.1 EXPERIMENTAL SETUP

We use MNIST handwritten digit dataset [40] for our experiments. Depending on specific use case, urban surveillance systems may need to recognize handwritten digits from various input sources, such as paper documents, touch screens, or handwritten digits in images captured by surveillance cameras. We clarify that we choose MNIST dataset for evaluating the effectiveness of combined data fusion and AI; nevertheless, experimental results for other datasets can be obtained by following a similar procedure. For our experiments, we randomly select 100 handwritten digit images for each digit from the dataset. We make 10 sets of images with each set containing 10 images for each digit class. For no fusion case, the inference is run on the original images in the set, and the final result is obtained by averaging the results of the CNN inferences of the images in the sets. For fusion case, we generate one fused image from each set, generating total 10 fused images for each digit. For data fusion, we introduce two data fusion techniques: average (AVG) fusion and sum of square differences (SSD) fusion.

For AVG fusion [41], we take average of the pixel values in the same location (i.e., per-pixel averaging) of 10 images in the same set. Figure 6 shows examples of 10 source images and their fused images for digits zero (0), two (2), and three (3). We also introduce an SSD-based average fusion technique. Although the AVG fusion approach can be beneficial for latency reduction, it is susceptible to outlier data, which can deteriorate classification accuracy. The SSD fusion help mitigate the adverse impact from the outliers. In SSD fusion, we exclude the outlier images based on the SSD when generating the fused image. Figure 7 shows how we fuse

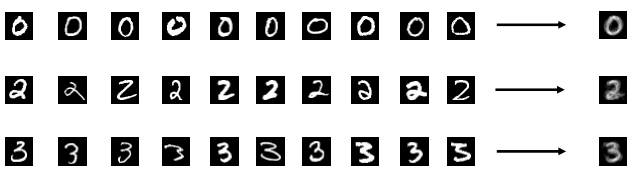


FIGURE 6. Examples of fused images of handwritten digits 0, 2, and 3 from MNIST dataset.

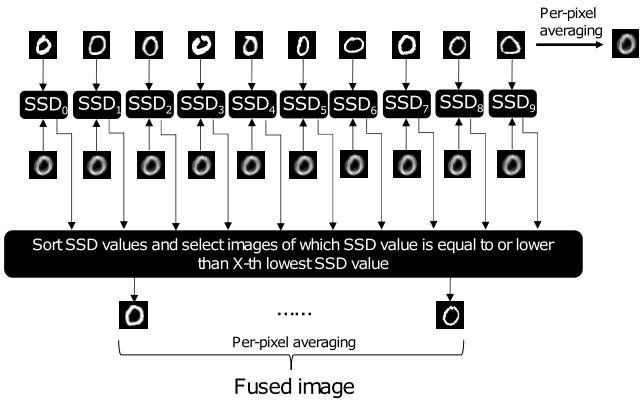


FIGURE 7. A sum of square differences (SSD-X) fusion technique.

images by using the SSD fusion technique. Assuming that we have ten images for fusion, we first generate a fused image from the ten images by using the AVG fusion (per-pixel averaging). We then calculate the SSD value between each image and averaged image. The normalized SSD value between two images (I and J of size $N \times M$) can be calculated as follows:

$$NormalizedSSD = \frac{\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}(J[n,m]-I[n,m])^2}{\sqrt{\sum J[n,m]^2 \times \sum I[n,m]^2}} \tag{13}$$

It is to be noted that we use normalized SSD values for better generalization of measuring the image differences. After calculating the ten normalized SSD values (SSD_0–SSD_9), we sort these values and select images of which normalized SSD value is equal to or lower than the X-th lowest normalized SSD value, which we call as SSD-X fusion. For example, SSD-5 chooses the images of which the normalized SSD value is equal to or lower than the 5th lowest normalized SSD value, thus, eventually choosing five images for fusion. By using the selected images while excluding the other non-selected images, we perform per-pixel averaging again, generating the final fused image. The SSD-X fusion technique effectively removes the outlier images so that we can expect an accuracy improvement by expending more computation steps (e.g., calculating and sorting normalized SSD values).

We consider three feasible fog/edge computing environments: (1) data fusion and CNN in Intel Xeon CPU [42] (Xeon_CPU), (2) data fusion and CNN in Nvidia Jetson TX2 (JTX2 [43]) CPU (JTX2_CPU), and (3) data fusion in JTX2 CPU and CNN in JTX2 GPU (JTX2_CPU+JTX2_GPU). We report the execution time of data fusion and CNN inference, which is averaged over 10 sets of 10 digits. We use Ubuntu 18.04, CUDA 10.1, and Python 3.6.9 for operating system, GPGPU (general purpose GPU) framework, and data fusion code implementation, respectively. For the case of AI with data fusion, 10 images are fused with our python code (accepting 10 images as inputs and generating one fused image by either AVG fusion or SSD-X fusion), and the resulting fused image is sent to

TABLE 3. Average latency (in milliseconds) comparison between the LeNet CNN execution among five cases (w/o fusion, AVG fusion, SSD-3 fusion, SSD-5 fusion, and SSD-7 fusion).

Fog Computing Platform	LeNet				
	w/o Fusion	AVG Fusion	SSD-3 Fusion	SSD-5 Fusion	SSD-7 Fusion
Xeon_CPU	49.77 ms	19.53 ms	26.95 ms	28.95 ms	30.27 ms
JTX2_CPU	162.07 ms	42.45 ms	44.91 ms	45.96 ms	48.16 ms
JTX2_CPU+JTX2_GPU	4920.08 ms	508.58 ms	514.61 ms	517.06 ms	524.88 ms

TABLE 4. Average latency (in milliseconds) comparison between the AlexNet CNN execution among five cases (w/o fusion, AVG fusion, SSD-3 fusion, SSD-5 fusion, and SSD-7 fusion).

Fog Computing Platform	AlexNet				
	w/o Fusion	AVG Fusion	SSD-3 Fusion	SSD-5 Fusion	SSD-7 Fusion
Xeon_CPU	3289.89 ms	338.59 ms	339.65 ms	343.17 ms	348.65 ms
JTX2_CPU	13666.13 ms	1393.09 ms	1396.28 ms	1400.76 ms	1408.63 ms
JTX2_CPU+JTX2_GPU	4668.74 ms	494.71 ms	496.08 ms	497.32 ms	501.89 ms

Darknet framework [44] as an input for CNN inference. For data fusion techniques, we consider AVG fusion, SSD-3 fusion, SSD-5 fusion, and SSD-7 fusion. For the case of AI without data fusion, each of the 10 images is sequentially processed with Darknet framework for CNN inferences. For CNN models, we use LeNet [45] and AlexNet [46]. For weights of CNN models, we use trained weights provided by [47] for LeNet, while we ourselves train the weights of AlexNet with MNIST training dataset (60,000 images). To smooth any discrepancies in latency due to operating system overhead or variations in environmental parameters, we calculate and present the average latency results from 10 independent measurements.

B.2 LATENCY COMPARISON BETWEEN AI WITHOUT DATA FUSION AND WITH DATA FUSION

Table 3 and Table 4 show the average latency comparison between the two CNN models, LeNet and AlexNet, respectively, with and without data fusion for the three fog/edge computing platforms (Xeon_CPU, JTX2_CPU, and JTX2_CPU+JTX2_GPU) described in experimental setup. For data fusion, results are shown for AVG fusion, SSD-3 fusion, SSD-5 fusion, and SSD-7 fusion. Results in Table 3 and Table 4 indicate that CNN with data fusion leads to huge latency reductions for both LeNet and AlexNet as compared to the CNN models' execution without information fusion.

Figure 8 depicts the speedup of AI ((a) LeNet and (b) AlexNet CNN models in our experiments) with data fusion over AI without data fusion. Results indicate that AI with data fusion provides substantial speedups over AI without data fusion. For example, when using Xeon CPU for data fusion and CNN models' execution, data fusion (averaged over AVG fusion, SSD-3 fusion, SSD-5 fusion, and SSD-7 fusion) leads to speedups of 1.9×–9.5× and 9.4×–9.8× for LeNet and AlexNet, respectively. In the case of using JTX2 (which can be used as an on-board processing platform for UAVs) CPU, data fusion results in 3.6× and 9.8× speedup for LeNet and AlexNet, respectively. When using GPUs for CNN inferences, data fusion also engenders huge speedups of 9.5× and 9.4× for LeNet and AlexNet, respectively. Since the SSD fusion requires more computations than the AVG fusion, SSD fusion leads to higher latency than the AVG fusion by 19.4% and 1.0%, on average, for LeNet and AlexNet, respectively.

As the CNN workloads can be accelerated with GPUs, offloading the CNN inference tasks to the GPU leads to 2.8× to 2.9× speedup for AlexNet compared to the case of only using the JTX2 CPU. However, results in Table 3 indicate that LeNet execution on GPUs leads to worse performance (i.e., higher execution time) as compared to execution on CPU. This is because LeNet is a very small CNN model with only ∼60 thousand parameters, which can be efficiently executed in the CPU without offloading. When using GPU for CNN inferences for LeNet, the data transfer among CPU, memory, and GPU incurs non-negligible latency overhead that results in higher overall execution time on GPU as compared to the execution on CPU. On the contrary, in the case of AlexNet, which is a very large CNN model with ∼60 million parameters, offloading CNN execution to the GPU actually leads to better performance (as shown in Table 4) because the computation time for large data dominates the data transfer time.

B.3 ACCURACY COMPARISON BETWEEN AI WITHOUT DATA FUSION AND WITH DATA FUSION

Since data fusion transforms input images for CNN inferences, data fusion can impact CNN classification accuracy.

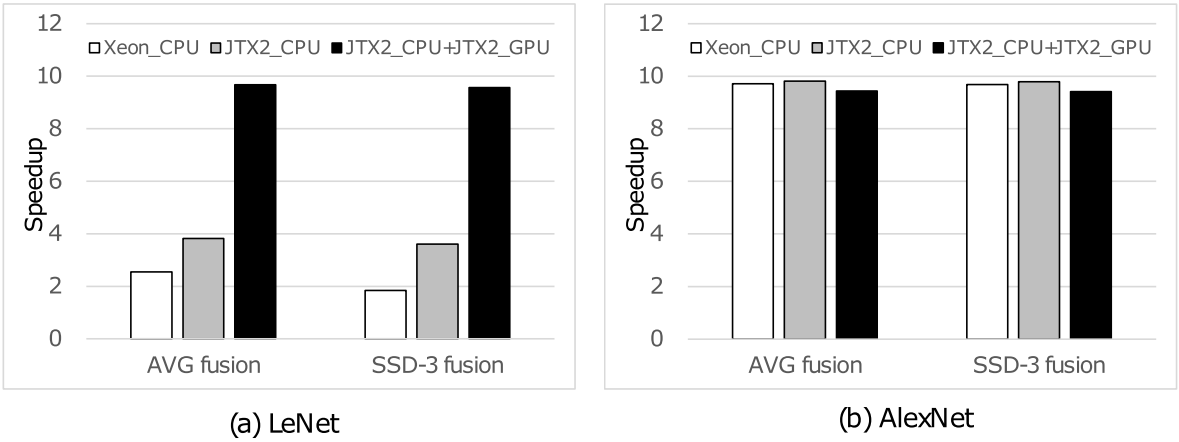


FIGURE 8. Speedup of AI with data fusion (AVG fusion and SSD-3 fusion) as compared to AI without data fusion.

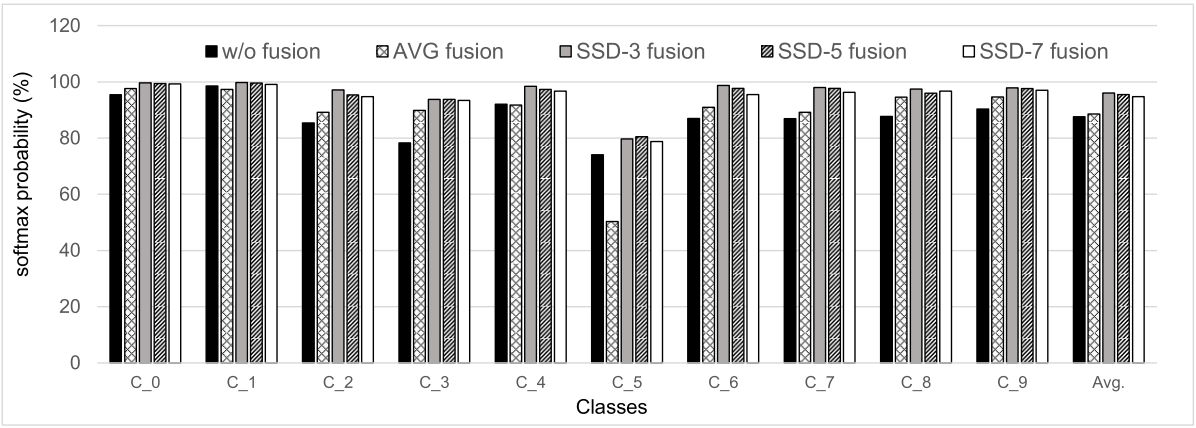


FIGURE 9. Per digit class (C_0 – C_9) accuracy comparison for five cases: w/o fusion, AVG fusion, SSD-3 fusion, SSD-5 fusion, and SSD-7 fusion.

We compare the accuracy results with LeNet for five cases: without fusion, AVG fusion, SSD-3 fusion, SSD-5 fusion, and SSD-7 fusion. Figure 9 summarizes the classification accuracy results across class 0 (C_0) – class 9 (C_9) for MNIST dataset, meaning that we separately show the accuracy results for each digit classification.

Results reveal that without fusion, the baseline classification accuracy, on average, over all digit classes is 87.6%. When employing the AVG fusion, we can obtain an average accuracy of 88.5%, which is slightly higher than the baseline accuracy (i.e., without fusion). This implies that the AVG fusion does not adversely affect the CNN inference accuracy; on the contrary it alleviates the adverse impact of outliers by averaging the pixel values from various images. However, for class C_5, the AVG fusion leads to a non-negligible accuracy loss of 23.7% as compared to the baseline. This is because for class C_5 in MNIST dataset, a non-negligible number of outliers are fused in the AVG fusion, resulting in an accuracy loss. On the other hand, the SSD fusion can help mitigate the accuracy losses from outliers. As summarized in Figure 9, SSD-3, SSD-5, and SSD-7 fusion techniques help improve LeNet classification accuracy, on average, by 7.5%, 7.0%, and 6.2%, respectively, over the AVG fusion. For class C_5, in particular, the SSD-3, SSD-5, and SSD-7 fusion techniques help improve LeNet classification accuracy by 29.3%, 30.2%, and 28.5%, respectively, over the AVG fusion. Furthermore, for class C_5, the SSD-3, SSD-5, and SSD-7 fusion techniques provide better classification accuracy than the baseline (i.e., no fusion) by 5.6%, 6.5%, and 4.8%, respectively. Results also indicate that SSD-3 fusion, on average over all classes, provides 8.5% improvement in accuracy over the baseline (i.e., no fusion). This implies that our proposed SSD fusion can effectively compensate for the accuracy losses by CNNs for the classification classes for which there is non-negligible outlier data present within the dataset. Results also show that the SSD-3 fusion shows the best accuracy among other SSD-X fusion techniques (i.e., SSD-5 and SSD-7 fusions for our experiments). This implies that more aggressive outlier removal can prove helpful for attaining high classification accuracy.

C. AVERAGE PRECISION AND LATENCY OF COMBINED MULTIMODAL DATA FUSION AND AI FOR UAVs IN FogSurv

Our third set of experimental results focus on evaluating the effectiveness of combining AI with data fusion from different types of sensors outfitted on UAVs in FogSurv. In some surveillance scenarios, it is desirable to obtain multiperspective and multimodal observations of a target. UAVs in FogSurv are befitting for this task as surveillance UAVs (Section IV) are outfitted with different types of sensors, such as visual (VI) and medium wavelength infrared (MWIR). Accordingly, a swarm of UAVs can self-organize based on the mission needs such that UAVs observe a target from different perspectives and distances. The data obtained from a swarm of UAVs can have *overlapping* or *orthogonal* viewpoints. The same viewpoints (0 degrees) maximize data registration whereas orthogonal viewpoints (90 degrees) present different perspective of a target. This set of experiments demonstrate *multimodal complementary data fusion* [41]. It is to be noted that there is a distinction between *multimodal complementary data fusion* and *multimodal competitive data fusion*. In competitive sensor fusion, either data from the same sensor modality is fused together or data from different sensor modalities is first transformed to the same baseline, and then fused. In complementary sensor fusion, sensors observe distinct parts of the same event and fusion of these sensors provide a complete characterization of the event. Theoretical models relate object distance to the visual electro-optical and infrared image resolution as well as to the multimodal image fusion for object detection and classification. Once theoretical models indicate benefits of multimodal image fusion, AI methods for contextual analysis are then employed to determine which configuration of multimodal sensors would give favorable results, that is, whether to utilize visual sensor only, or infrared sensor only, or both visual and infrared sensors, etc. We note that context from the scenario includes lighting conditions and position of sensor as a function of range.

We have evaluated this set of experiments on automatic target recognition (ATR) dataset from the Military Sensing Information Analysis Center (SENSIAC) [48]. This dataset contains 207 GB of MWIR imagery and 106 GB of VI imagery along with ground truth data. The dataset was collected during both the daytime and nighttime with different observation distances ranging from 500 m to 5000 m. For multimodal fusion, we consider three types of images: MWIR, VI, and motion image (MI). The MI can be calculated from VI as follows:

$$M = |I_t(x, y) - I_{t-\delta}(x, y)|, \tag{14}$$

where M represents MI, I represents original VI, (x, y) indicates the 2D coordinates in the image array, t denotes the tth frame in a continuous image sequence, and δ denotes the frame interval between two consecutive frames which is affected by the frame sample rate. For target detection, we use faster region-based CNN (R-CNN) coupled with ResNet-101 [49]. We use *average precision* as the evaluation metric, which can be calculated by finding the area under the precision-recall curve.

There are many factors that deteriorate the performance of target detection from UAVs among which is the *target scale*. There exists an inverse relation between the target scale and observation distance, that is, a larger observation distance leads to a smaller target scale and vice versa. In this set of experiments, we use the *observation distance* to represent the relative *target scale*. We have selected the set of data from SENSIAC dataset across different observation distances ranging from 1000 m to 5000 m. The observation distance range of [1000 m, 2000 m] can be classified as corresponding to large target scale whereas the observation distance range of [2500 m, 5000 m] can be classified as corresponding to small target scale, where within these ranges, larger the distance, smaller is the target scale. To verify the effectiveness of multimodal fusion, we have implemented four detectors of incremental image modality, that is, from single image modality (MWIR and VI) to multimodal image fusion (MWIR + VI and MWIR + VI + MI).

Figure 10 shows average precision results against observation distances for different modalities. Results indicate that for relatively larger target scale (i.e., the observation distance range of [1000 m, 1500 m]), the VI, MWIR + VI and MWIR + VI + MI results in pretty close average precision whereas MWIR results in slightly lesser average precision. Results further reveal that at a observation distance of 2000 m, MWIR + VI results in best average precision (average precision of 99.1%) followed by MWIR + VI + MI (average precision of 97.9%), which is trailed by VI (average precision of 97.5%) followed by MWIR (average precision of 93.6%). Results indicate that MI modality results in least average precision for target detection, and in many instances also deteriorate the performance of MWIR + VI + MI as compared to MWIR + VI. For relatively smaller target scale (i.e., the observation distance range of [2500 m, 5000 m]), the detection performance based on all types of image modalities decreases significantly with the increasing observation distances. Results show that at the observation distance of 2500 m, MWIR + VI, MWIR + VI + MI, VI, and MWIR results in average precision of 99%, 98%, 97%, and 79%, respectively.

Results also indicate that MWIR + VI and MWIR + VI + MI perform better than other image modalities, in particular, for extremely small targets corresponding to observation distances of 4500 m and 5000 m. In particular, for the observation distance of 5000 m, target detection based on MWIR + VI + MI provides 2.8× and 7× improvement of average precision over VI and MWIR, respectively. Results also indicate that average precision of target detection from almost all image modalities perform poorly at the observation distance of 3000 m as compared to the observation distance of 3500 m. This indicates that other than target scale, there exists other critical factors affecting the performance of target detection. In this case, the complexity of scene such as presence of buildings, trees, and variation in terrain obscure target

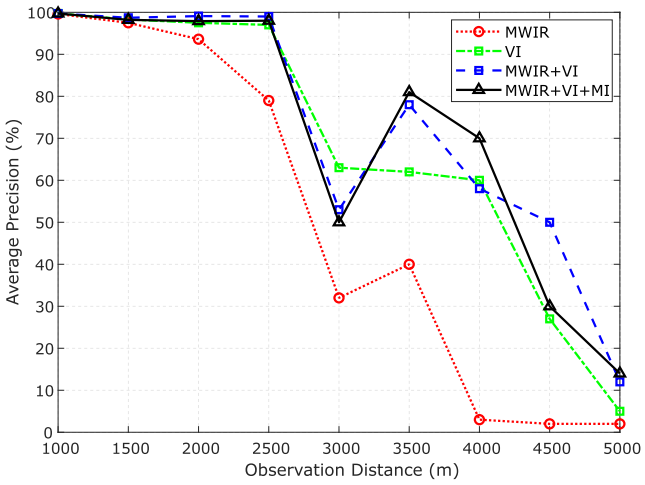


FIGURE 10. Precision versus observation distance for multimodal fusion and AI for UAVs.

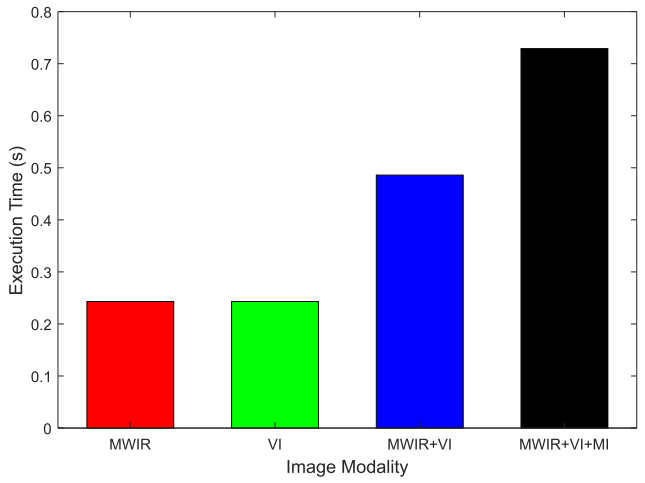


FIGURE 11. Execution time for target detection for different image modalities.

objects, which makes target detection challenging. We refer to this factor impacting performance of target detection as *environment complexity*. This environment complexity can be quantified by calculating the signal-to-noise ratio (SNR) at different observation distances [49]. Our results indicate that SNR of different image modalities, on average, is lower at the observation distance of 3000 m as compared to the observation distance of 3500 m, where a lower SNR indicates higher environmental complexity and a higher SNR represents relatively lower environmental complexity.

Figure 11 presents the execution time results for target detection on Intel Core i7 CPU and NVIDIA GeForce GTX 1080 GPU for different image modalities obtained by UAVs in FogSurv. Results indicate that execution time increases for target detection as the number of image modalities to be fused together increases. Results indicate that MWIR + VI + MI results in 1.5× higher execution time than MWIR + VI, which in turn results in 2× higher execution time than target detection through single image modalities (VI and MWIR). The higher computation time when combining MI to MWIR + VI is due to the computation of MI from VI (see Eq. (14)) and then performing image fusion. We note that the execution time results in Figure 11 correspond to pixel-by-pixel image fusion and searching for targets over the whole image. When employing more sophisticated fusion techniques, such as deep multimodal image fusion (DIF) technique [49] that fuses the complementary information from multimodal images using a CNN, execution time for multimodal fusion can be reduced as shown in [49]. By looking at both the average precision and runtime results, we can see that MWIR + VI presents a good tradeoff between average precision and latency, and is feasible for real-time target detection as compared to MWIR + VI + MI and single image modalities (VI and MWIR).

X. CONCLUSION

In this article, we have proposed FogSurv—a fog-assisted architecture and framework leveraging artificial intelligence (AI), information fusion, and dynamic data-driven methods for enabling real-time urban surveillance. We have elaborated data-driven usage of unmanned aerial vehicles (UAVs) in FogSurv for airborne urban surveillance and enhancing situational awareness (SA). Afterwards, we have proposed a framework for AI processing in FogSurv to assist with surveillance tasks. We have further proposed an AI- and data-driven model for information fusion in FogSurv. We have also modeled latency of AI and information fusion processing in FogSurv considering transmission, processing, and queueing delays at Internet of things (IoT) devices, fog/edge, and cloud. We have discussed various use cases of FogSurv that include battlefield applications, gray zone warfare, homeland security and defense, and monitoring and early warning systems. To illustrate the superiority of FogSurv over cloud-based platforms for real-time urban surveillance, we have conducted experiments to compare the average latency of benchmarks executed locally on IoT devices versus offloading those benchmarks' execution to fog nodes as well as to cloud. Results verify that for the selected benchmarks, offloading IoT tasks to fog nodes in FogSurv provide latency improvement of 37%, on average, as compared to offloading the tasks to the cloud. We have further evaluated the effectiveness of combining data fusion with AI for surveillance by providing latency and accuracy comparison between convolutional neural networks (CNN) processing with and without data fusion. Results indicate that combining AI with data fusion can provide a speedup of up to 9.8× over AI without data fusion while maintaining or improving the inference accuracy. An accuracy improvement of up to 8.5% has been observed in our experiments by using AI combined with data fusion as compared to AI without data fusion. We have also evaluated the average precision and latency of combined multimodal data fusion and AI for target detection by UAVs in FogSurv for different target scales and environment complexity. Results indicate that

multimodal data fusion results in improved average precision of target detection over single image modalities for different target scales and environment complexity. In particular, target detection using multimodal data fusion can provide up to 7× improvement of average precision over single modalities for extremely small target scales.

In the future work, we plan to conduct scalability analysis of the proposed FogSafe framework. One of the main objectives of the scalability analysis will be to determine the number of fog servers required in a geographical location to handle a given number of sensors and IoT devices to ensure that latency, real-time, energy, and throughput constraints of the surveillance application are satisfied for different communication technologies, such as WiFi and 5G. Another objective of the scalability analysis will be to determine how many UAVs will be required in a given geographical area for data collection from sensors and acting as fog nodes for different communication technologies to ensure that surveillance is conducted in a timely and energy-efficient manner.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the AFRL and AFOSR. In particular, the U.S. Department of the Air Force notes that the U.S. Department of Defense mission does not include surveillance of cities in the United States.

REFERENCES

[1] Wikipedia. (Sep. 2018). *2017 Las Vegas Shooting*. [Online]. Available: https://en.wikipedia.org/wiki/2017_Las_Vegas_shooting

[2] VOA. (Sep. 2018). *List of us Mass Shootings*. [Online]. Available: https://www.voanews.com/a/list-of-us-mass-shootings/4255503.html

[3] M. Carminati, O. Kanoun, S. L. Ullo, and S. Marcuccio, "Prospects of distributed wireless sensor networks for urban environmental monitoring," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 6, pp. 44–52, Jun. 2019.

[4] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.

[5] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, and R. Zimmermann, "Dynamic urban surveillance video stream processing using fog computing," in *Proc. IEEE 2nd Int. Conf. Multimedia Big Data (BigMM)*, Taipei, Taiwan, Apr. 2016, pp. 105–112.

[6] J. Wang, J. Pan, and F. Esposito, "Elastic urban video surveillance system using edge computing," in *Proc. Workshop Smart Internet Things*, San Jose, CA, USA, Oct. 2017, pp. 1–6.

[7] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp, "Urban surveillance systems: From the laboratory to the commercial world," *Proc. IEEE*, vol. 89, no. 10, pp. 1478–1497, Oct. 2001.

[8] L. Hu and Q. Ni, "IoT-driven automated object detection algorithm for urban surveillance systems in smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 747–754, Apr. 2018.

[9] G. Liu, S. Liu, K. Muhammad, A. K. Sangaiah, and F. Doctor, "Object tracking in vary lighting conditions for fog based intelligent surveillance of public spaces," *IEEE Access*, vol. 6, pp. 29283–29296, 2018.

[10] B. Liu, Y. Chen, A. Hadiks, E. Blasch, A. Aved, D. Shen, and G. Chen, "Information fusion in a cloud computing era: A systems-level perspective," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 29, no. 10, pp. 16–24, Oct. 2014.

[11] Z. Wang, E. Blasch, G. Chen, D. Shen, X. Lin, and K. Pham, "A low-cost, near-real-time two-UAS-based UWB emitter monitoring system," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 30, no. 11, pp. 4–11, Nov. 2015.

[12] R. M. Shukla and A. Munir, "An efficient computation offloading architecture for the Internet of Things (IoT) devices," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2017, pp. 728–731.

[13] A. E. Eshratifar and M. Pedram, "Energy and performance efficient computation offloading for deep neural networks in a mobile cloud computing environment," in *Proc. Great Lakes Symp. VLSI*, Chicago, IL, USA, May 2018, pp. 111–116.

[14] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Latency-aware application module management for fog computing environments," *ACM Trans. Internet Technol.*, vol. 19, no. 1, pp. 1–21, Mar. 2019.

[15] R. K. Naha, S. Garg, A. Chan, and S. K. Battula, "Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment," *Future Gener. Comput. Syst.*, vol. 104, pp. 131–141, Mar. 2020.

[16] R. K. Naha and S. Garg, "Multi-criteria-based dynamic user behaviour-aware resource allocation in fog computing," *ACM Trans. Internet Things*, vol. 2, no. 1, pp. 1–31, Feb. 2021.

[17] A. Bose, A. Munir, and N. Shabani, "A quantitative analysis of big data clustering algorithms for market segmentation in hospitality industry," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2020, pp. 1–6.

[18] A. Munir, P. Kansakar, and S. U. Khan, "IFCIoT: Integrated fog cloud IoT: A novel architectural paradigm for the future Internet of Things," *IEEE Consum. Electron. Mag.*, vol. 6, no. 3, pp. 74–82, Jul. 2017.

[19] Cisco. (2021). *What is Edge Computing?*. [Online]. Available: https://www.cisco.com/c/en/us/solutions/computing/what-is-edge-computing.html# revenue-opportunities

[20] A. Boyle. (Jul. 2012). *The US and its UAVs: A Cost-Benefit Analysis*. [Online]. Available: https://www.americansecurityproject.org/the-us-and-its-uavs-a-cost-benefit-analysis/

[21] Y. Alghamdi, A. Munir, and H. M. La, "Architecture, classification, and applications of contemporary unmanned aerial vehicles," *IEEE Consum. Electron. Mag.*, early access, Mar. 4, 2021, doi: 10.1109/MCE.2021.3063945.

[22] U. S. Army. (2010). *Eyes of The Army: U. S. Army Roadmap for Unmanned Aircraft Systems 2010-2035*. U.S. Army, United States. [Online]. Available: https://fas.org/irp/program/collect/uas-army.pdf

[23] AirForce Technology. (2021). *RQ-170 Sentinel Unmanned Aerial Vehicle*. [Online]. Available: https://www.airforce-technology.com/projects/rq-170-sentinel/

[24] D. Allaire, D. Kordonowy, M. Lecerf, L. Mainini, and K. Willcox, "Multifidelity DDDAS methods with application to a self-aware aerospace vehicle," *Procedia Comput. Sci.*, vol. 29, pp. 1182–1192, Jan. 2014.

[25] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine Learning: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2017.

[26] K. Lee, J. Kong, and A. Munir, "HW/SW co-design of cost-efficient CNN inference for cognitive IoT," in *Proc. Int. Conf. Intell. Comput. Data Sci. (ICDS)*, Fez, Morocco, Oct. 2020, pp. 1–6.

[27] M. A. Qureshi and A. Munir, "NeuroMAX: A high throughput, multi-threaded, log-based accelerator for convolutional neural networks," in *Proc. 39th Int. Conf. Comput.-Aided Design*, Nov. 2020, pp. 1–9.

[28] Y. Kim, J. Kong, and A. Munir, "CPU-accelerator co-scheduling for CNN acceleration at the edge," *IEEE Access*, vol. 8, pp. 211422–2114332, Nov. 2020.

[29] E. Blasch, I. Kadar, L. L. Grewe, R. Brooks, W. Yu, A. Kwasinski, S. Thomopoulos, J. Salerno, and H. Qi, "Panel summary of cyber-physical systems (CPS) and Internet of Things (IoT) opportunities with information fusion," *Proc. SPIE*, vol. 10200, May 2017, Art. no. 102000O.

[30] A. Munir, A. Gordon-Ross, and S. Ranka, "Multi-core embedded wireless sensor networks: Architecture and applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1553–1562, Jun. 2014.

[31] E. Blasch, G. Seetharaman, and K. Reinhardt, "Dynamic data driven applications system concept for information fusion," *Procedia Comput. Sci.*, vol. 18, pp. 1999–2007, Jan. 2013.

[32] E. P. Blasch, D. A. Lambert, P. Valin, M. M. Kokar, J. Llinas, S. Das, C. Chong, and E. Shahbazian, "High level information fusion (HLIF): Survey of models, issues, and grand challenges," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 27, no. 9, pp. 4–20, Sep. 2012.

[33] D. Patterson and J. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*. San Mateo, CA, USA: Morgan Kaufmann, 2014.

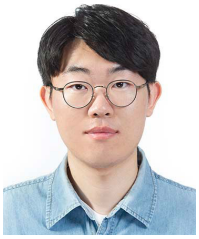
[34] A. Munir and A. Gordon-Ross, "SIP-based IMS signaling analysis for WiMax-3G interworking architectures," *IEEE Trans. Mobile Comput.*, vol. 9, no. 5, pp. 733–750, May 2010.

[35] A. Munir, A. Gordon-Ross, S. Ranka, and F. Koushanfar, "A queueing theoretic approach for performance evaluation of low-power multicore embedded systems," *J. Parallel Distrib. Comput.*, vol. 74, no. 1, pp. 1872–1890, Jan. 2014.
[36] A. Castiglione, K.-K. R. Choo, M. Nappi, and S. Ricciardi, "Context aware ubiquitous biometrics in edge of military things," *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 16–20, Nov./Dec. 2017.
[37] J. L. Votel, C. T. Cleveland, C. T. Connett, and W. Irwin, "Unconventional warfare in the gray zone," *Joint Force Quart.*, vol. 80, pp. 101–109, Jan. 2016.
[38] D. King, "A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, Dec. 2009.
[39] P. Kansakar, A. Munir, and N. Shabani, "A fog-assisted architecture to support an evolving hospitality industry in smart cities," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Islamabad, Pakistan, Dec. 2018, pp. 59–64.
[40] Y. LeCun, C. Cortes, and C. J. C. Burges. *The MNIST DATABASE of Handwritten Digits*. Accessed: Apr. 30, 2021. [Online]. Available: https://yann.lecun.com/exdb/mnist
[41] A. Munir, E. Blasch, J. Kwon, J. Kong, and A. Aved, "Artificial intelligence and data fusion at the edge," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 36, no. 7, pp. 62–78, Jul. 2021.
[42] Intel. *Xeon Processor E3-1230 v5*. Accessed: Jul. 20, 2021. [Online]. Available: https://ark.intel.com/content/www/kr/ko/ark/products/88182/intel-xeon-processor-e3-1230-v5-8m-cache-3-40-ghz.html
[43] Nvidia. *Jetson TX2*. Accessed: Jul. 20, 2021. [Online]. Available: https://developer.nvidia.com/embedded/buy/jetson-tx2
[44] J. Redmon. (2013–2016). *DarkNet: Open Source Neural Networks in C*. [Online]. Available: https://pjreddie.com/darknet/
[45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[47] T. Ashitani. (2020). *Darknet_Mnist*. [Online]. Available: https://github.com/ashitani/darknet_mnist
[48] D. Shumaker. (Nov. 2008). *SENSIAC (Military Sensing Information Analysis Center) for PEOs/PMs*. Military Sensing Information Analysis Center Atlanta, GA. (Jul. 13, 2021). [Online]. Available: https://apps.dtic.mil/sti/citations/ADA535651
[49] S. Liu, H. Liu, V. John, Z. Liu, and E. Blasch, "Enhanced situation awareness through CNN-based deep multimodal image fusion," *Opt. Eng.*, vol. 59, no. 5, p. 1, May 2020.

ARSLAN MUNIR (Senior Member, IEEE) received the M.A.Sc. degree in ECE from The University of British Columbia (UBC), Vancouver, Canada, in 2007, and the Ph.D. degree in ECE from the University of Florida (UF), Gainesville, FL, USA, in 2012.

From 2007 to 2008, he worked as a Software Development Engineer at Embedded Systems Division, Mentor Graphics Corporation. He was a Postdoctoral Research Associate with the Electrical and Computer Engineering (ECE) Department, Rice University, Houston, TX, USA, from May 2012 to June 2014. He is currently an Associate Professor with the Department of Computer Science (CS), Kansas State University (KSU). His current research interests include embedded and cyber-physical systems, secure and trustworthy systems, parallel computing, reconfigurable computing, and computer vision. He received many academic awards, including the Doctoral Fellowship from Natural Sciences and Engineering Research Council (NSERC) of Canada. He earned the gold medals for best performance in electrical engineering, the gold medals, and an Academic Roll of Honor for securing rank one in Pre-Engineering Provincial Examinations (out of approximately 300000 candidates).

JISU KWON (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, in 2018 and 2020, respectively. He is currently with Samsung Electronics. His research interests include convolutional neural network acceleration, mobile system-on-chip design, and FPGA-based design.



JONG HUN LEE (Student Member, IEEE) received the B.S. degree in electronics engineering from Kyungpook National University, in 2020, where he is currently pursuing the M.S. degree in electronic and electrical engineering. His research interests include convolutional neural network acceleration, data compression, and FPGA-based design.

JOONHO KONG (Member, IEEE) received the B.S. degree in computer science and the M.S. and Ph.D. degrees in computer science and engineering from Korea University, in 2007, 2009, and 2011, respectively.

He worked as a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, Rice University, from 2012 to 2014. Before joining Kyungpook National University, he worked as a Senior Engineer at Samsung Electronics, from 2014 to 2015. He is currently an Associate Professor with the School of Electronics Engineering, Kyungpook National University. His research interests include computer architecture, heterogeneous computing, embedded systems, and hardware/software co-design.

ERIK BLASCH (Fellow, IEEE) received the B.S. degree in mechanical engineering from Massachusetts Institute of Technology, in 1992, the master's degree in mechanical engineering from Georgia Tech, in 1994, the master's degree in health science from Georgia Tech, in 1995, the master's degree in industrial engineering (Human Factors) from Georgia Tech, in 1995, the M.D./Ph.D. degree in neuroscience/mechanical engineering from the University of Wisconsin, in 1996, the M.B.A. degree in electrical engineering from Wright State University, in 1998, the M.S.Econ. degree in electrical engineering from Wright State University, in 1999, the Ph.D. degree in electrical engineering from Wright State University, in 1999, and graduated from the Air War College, in 2008. He is currently a Program Officer at United States (U.S.) Air Force Research Laboratory (AFRL)—Air Force Office of Scientific Research (AFOSR), Arlington, VA, USA. Previously, he was the Information Fusion Evaluation Tech Lead at the AFRL Sensors Directorate—COMprehensive Performance Assessment of Sensor Exploitation (COMPASE) Center, Dayton, OH, USA, from 2000 to 2009, an Exchange Scientist at the Defence Research and Development Canada (DRDC), Valcartier, QC, from 2010 to 2012, and a Principal Scientist at the AFRL Information Directorate, Rome, NY, USA, from 2012 to 2017.

Additional assignments include United States Air Force (USAF) Reserve Officer Colonel supporting intelligence, acquisition, and space technology with 18 medals. Academically, he was an Adjunct Associate Professor in electrical and biomedical engineering at Wright State University, from 2000 to 2010, the Air Force Institute of Technology (AFIT) teaching classes in signal processing, electronics, and information fusion, and a Research Adjunct appointments at the University of Dayton, Binghamton University, and Rochester Institute of Technology. He attended the University of Wisconsin for a M.D./Ph.D. degree in neuroscience/mechanical engineering until being call to military service, in 1996, to USAF. He is an American Institute of Aeronautics and Astronautics (AIAA) Associate Fellow and the Society of Photonics and Industrial Engineers (SPIE) Fellow. He was a recipient of the IEEE Bioengineering Award (Russ-2008), the IEEE AESS Magazine Best Paper Award (Mimno-2012), the Military Sensing Symposium Leadership in Data Fusion Award (Mignogna-2014), the Fulbright Scholar Selection, in 2017, and the 15 Research/Technical and Team Awards from AFRL.

ALEXANDER J. AVED (Senior Member, IEEE) received the B.A. degree in computer science and mathematics from Anderson University, Anderson, IN, USA, in 1999, the M.S. degree in computer science from Ball State University, and the Ph.D. degree in computer science from the University of Central Florida, in 2013, with a focus on real-time multimedia databases. He is currently a Technical Advisor at the Air Force Research Laboratory Information Directorate, Rome, NY, USA. His research interests include multimedia databases, stream processing (via CPU, GPU, or coprocessor), dynamically executing models with feedback loops incorporating measurement, and error data to improve the accuracy of the model.

KHAN MUHAMMAD (Member, IEEE) received the Ph.D. degree in digital contents from Sejong University, Republic of Korea, in February 2019. He is currently the Director of the Visual Analytics for Knowledge Laboratory (VIS2KNOW Lab), Sungkyunkwan University, Seoul, South Korea. He has registered eight patents in South Korea (seven)/Australia (one) and has contributed more than 150 articles in peer-reviewed journals and conference proceedings in his areas of research. He was recently selected among top 100 000 scientists around the globe by Stanford Researchers List. His research interests include intelligent video surveillance (fire/smoke scene analysis, transportation systems, and disaster management), medical image analysis, (brain MRI, diagnostic hysteroscopy, and wireless capsule endoscopy), information security (steganography, encryption, watermarking, and image hashing), video summarization, multimedia data analysis, computer vision, the IoT/IoMT, and smart cities.