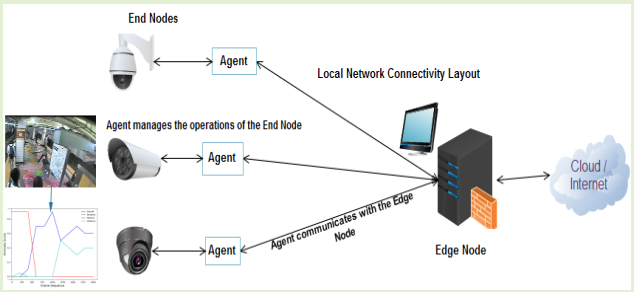
SurveilNet: A Lightweight Anomaly Detection System for Cooperative IoT Surveillance Networks

Martins O. Osifeko, *Student Member, IEEE*, Gerhard P. Hancke, *Life Fellow, IEEE*, and Adnan M. Abu-Mahfouz, *Senior Member, IEEE*

***Abstract*—The boring and repetitive task of monitoring video feeds makes real-time anomaly detection tasks difficult for humans. Hence, crimes are usually detected hours or days after the occurrence. To mitigate this, the research community proposes the use of a deep learning-based anomaly detection model (ADM) for automating the monitoring process. However, the isolated setup of existing surveillance systems makes ADM inefficient and susceptible to staleness due to the lack of resource sharing and continuous learning (CL). CL is the incremental development of models that adapts continuously to the external world. Thus, for efficient CL in surveillance systems, devices must share resources and cooperate with neighbor sites. Yet, solutions from the literature focus on the isolated environment thereby neglecting the need for resource sharing and CL. To address this gap, this paper proposes a cooperative surveillance system called SurveilNet that allows for resource sharing between surveillance sites under the control of a cooperator node. We further propose a lightweight subscription scheme that allows for a joint specialized model development process that continually adapts to the dynamics of the secured environment. Our proposed scheme offers the ability to learn from the neighboring site's data without compromising data privacy. The performance of our scheme is evaluated using a reclassified UCF-Crime dataset with the result showing the efficiency of our proposed scheme when compared to the state-of-the-art.**



***Index Terms*— Anomaly detection system, federated deep learning, Internet of Things (IoT), surveillance systems.**

I. Introduction

ONE of the growing application areas of IoT is surveillance systems where ubiquitous devices such as sensors, cameras, drones, etc. are used for monitoring purposes [1], [2]. Generally, during surveillance, an object, person, or location of interest is continually monitored for crime prevention or other related purposes [3], [4]. In such systems, an operator monitors the feeds from various installations to detect anomalies. However, the boring and monotonous nature of this task makes it difficult for the operators to prevent crime, or violations before occurrence. Hence, surveillance systems have been mostly used as a tool for collecting crime evidence.

For surveillance systems, the need to have smarter nodes is further underscored by the sheer volume of raw data generated by these devices and the resultant burden of transmitting the data across the network [5]–[7]. Apart from this, sometimes, deployments can take place in locations with limited energy and unstable network connectivity. As a result, intelligent on-device processing and energy-efficient operations become a necessary feature for optimal performance [8]–[12].

The recent advancement in deep learning (DL) technology is providing new milestones in the activity and scene recognition research domain [13]. The major task in this area is the use of extracted image features to obtain structural information that can identify the face or scene category. When integrated with surveillance systems, DL possesses the ability to automatically prevent or detect crimes via high-speed inference generation with pre-trained DL models [14]–[16]. Succinctly stated, using DL, a surveillance system can be made to predict illegal

Manuscript received June 24, 2021; revised July 26, 2021; accepted July 26, 2021. Date of publication August 6, 2021; date of current version November 12, 2021. The associate editor coordinating the review of this article and approving it for publication was Prof. Subhas C. Mukhopadhyay. *(Corresponding author: Gerhard P. Hancke.)*

Martins O. Osifeko is with the Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa (e-mail: u18379428@tuks.co.za).

Gerhard P. Hancke is with the College for Automation and Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing 210049, China, and also with the Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa (e-mail: g.hancke@ieee.org).

Adnan M. Abu-Mahfouz is with the Council for Scientific and Industrial Research, Pretoria 0184, South Africa, and also with the Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa (e-mail: a.abumahfouz@ieee.org).

Digital Object Identifier 10.1109/JSEN.2021.3103016

activities, detect crimes, trigger alarms or notify relevant authorities when a set rule is violated.

Nevertheless, despite these potentials, such surveillance systems are still largely unavailable due to reasons such as inadequate training data, data privacy problems, computational capability, reliability, and latency [17]. Furthermore, anomaly detection in surveillance videos poses several challenges such as inappropriate feature extraction, the difficulty of defining normal and anomalous activities and their imbalanced distribution in training data, occlusion, environmental variations, camera quality, etc [18]. In addition to these challenges, existing surveillance networks are usually isolated, non-cooperative, and under the private control of an organization, which makes data sharing difficult. A drawback of this scenario is that a monitored location A can only learn from its local data. For efficiency, location A should learn from security violations in location B and prevent a repeat occurrence in its premises. Furthermore, a continuous learning (CL) process becomes difficult in an isolated setup due to the limited availability of training data. CL provides for an incremental development of models that adapts to the external world [19]. Thus, for effective CL, devices need to cooperate and share resources.

In addressing some of these problems, research efforts focus on the use of DL techniques to automate the monitoring process in an isolated environment [20], [21]. However, the earlier identified drawbacks make these solutions inefficient for a robust surveillance system capable of handling modern security challenges. Thus, to address these problems, this paper proposes the use of a cooperative network setup that leverages interconnectivity among nodes to improve system capacity and quality of service requirements. Our proposed system provides for data sharing, specialized model development, and a continuous learning process for various devices in the network.

Recently, the use of the federated learning (FL) technique that cooperatively trains a machine learning (ML) model across multiple decentralized edge nodes to support IoT operations is gaining traction. Unlike the conventional ML approach where a central server uses aggregated datasets to train a single model, FL trains multiple models on multiple nodes after which the various model updates (weights) are aggregated to form a single robust model without exchanging the underlying training dataset. These benefits have made FL more attractive for use in various IoT applications such as Industrial IoT [22], [23], healthcare [24], [25], data processing [26], etc. Thus, to solve the identified drawbacks in existing surveillance systems, this paper leverages FL techniques to provide solutions needed for a cooperative surveillance system. To be precise, the contributions of this paper are as follows:

i. Firstly, to solve the problem of data sharing and inadequacy of training data, we propose a cooperative network setup for surveillance systems. The setup allows isolated sites to cooperate and provide specialized data without compromising privacy.
ii. Secondly, we present a lightweight subscription scheme that allows nodes to continuously learn from new anomalies. The scheme provides for the cooperative training of specialized anomaly detection models (ADM) that is suitable for use in the proposed network.
iii. Lastly, to evaluate the system, we reclassify the UCF crime dataset [27] and make it available to the research community [28]. Then, based on our presented scheme, we develop specialized anomaly detection models that are suitable for use in five application areas.

The rest of the paper is organized as follows: Section II provides a review of related works whereas Section III presents the system model and the problem formulation. Section IV presents the proposed SurveilNet system where the architecture and operating algorithm are discussed. In section V, we evaluate the performance of the system and present the results in Section VI. Finally, the conclusion and future research direction are presented in Section VII.

II. Literature Review

A. Related Works

Recently, anomaly detection in video streams and the deployment strategy for surveillance systems have been the focus of many types of research. For instance, Ding, *et al.* [1] posited that apart from connectivity needs, surveillance devices must be able to learn, think, and understand their physical and social worlds. Thus, to facilitate this vision, the authors proposed a Dragnet system, that empowers a drone surveillance system to carry out cognitive tasks such as sensing, data analytics, semantic derivation, and intelligent decision-making. The proposed system is then evaluated for its proof of concept using a drone detection and classification problem.

Shreyas, *et al.* [29] proposed the use of an adaptive video compression technique and the convolutional 3D network with contextual multiple scales based on the temporal features proposed by Xu, *et al.* [30]. The paper adaptively compresses the video feed before passing it through an activity recognition system. The adaptive nature of the compression technique operates by encoding the insignificant portion of the video feed with low-level precision whereas the significant and semantically meaningful portions of the video are encoded with higher precision. The encoded video frames are then fed into an anomaly detection system that classifies the frames into their respective anomaly class. The scheme, however, like the work done in [20], [27], [31] provides no support for continuous improvement of the recognition system which is needed for a robust surveillance system. Also, Sun, *et al.* [32] proposed an end-to-end model that uses a one-class Support Vector Machine (SVM) into Convolutional Neural Network (CNN), named Deep One-Class (DOC) model optimized using a loss function. The proposed work uses CNN to extract high-level features from image data, which is then fed into the OC-SVM for classification into an abnormal or normal event. Amraee, *et al.* [33] proposed a method for detecting abnormal events in surveillance systems that extracts candidate regions from feeds and eliminates redundant information. The histogram of oriented gradients, local binary pattern, a histogram of optical flow are then calculated from the candidate regions. The values are then fed into a one-class support vector machine that detects anomalies. Gu, *et al.* [34] propose a

TABLE I
EXISTING DETECTION METHODS AND THEIR SHORTCOMING FOR A ROBUST SURVEILLANCE SYSTEM

Existing Detection Methods	Lightweight	Data Privacy	Scalability	Continuous learning	Resource Sharing	Special Usage
Hybrid Autoencoders [20]	√	×	√	×	×	×
C3D Feature Extraction [27]	√	×	√	×	×	×
Adaptive Compression and C3D network with contextual multiple scales [29]	×	√	√	×	×	√
CNN and One-class Support Vector Machine [32, 33]	√	×	√	×	×	×
Two Stream CNN [36]	×	√	√	×	×	×
Recurrent Neural Network-based Feature Extraction [37]	×	√	√	×	×	×
This Paper	√	√	√	√	√	√

cooperative network architecture for multiple unmanned aerial vehicles surveillance aimed at the recognition and localization of moving small targets whereas authors in [35] discussed the main issues related to the design of an architecture for a smart cooperative video surveillance system. The work further proposed a use case to demonstrate the feasibility and the interoperability of the system. Table I provides a summary of existing methods and their shortcomings for a robust surveillance system. Despite the impressive results obtained from these works, their application areas remain limited. Firstly, the solutions are developed for an isolated setup which makes them inefficient in a cooperative environment where there is a need for resource sharing and continuous learning. Apparently, for a robust surveillance network, sites should be inter-connected to provide for data, knowledge, and model sharing. This allows a site to learn from an anomalous event that occurred in a neighbor site. Secondly, the solutions are general applications which makes them less effective for specialized usage. Generally, the challenge of anomaly detection in a cooperative surveillance environment differs from an isolated environment. These peculiarities and how this paper addresses them are summarized in the following subsection.

B. Potential Challenges in a Cooperative Surveillance System and Our Contributions

The following challenges differentiate a cooperative surveillance environment from an isolated one;

1) Network Congestion: In the case of network congestion or failure, communication between sites becomes impossible or degraded. To avoid this, our approach uses a local model broadcast to avoid communication outages on the cooperator node.

2) Heterogenous Requirements Support: Most times, the security, privacy, and task requirements for various monitored sites differ. Our proposed system uses a layered architecture that caters to the requirement of individual sites.

3) Heterogeneous Device Support: Operations in a cooperative network must cater to various capabilities of participating nodes. Our system uses an asynchronous approach to the model development process that accommodates the capabilities of various devices.

4) Privacy/Trust Challenges: Data owners at the various sites may be unwilling to release their data to the control center due to privacy or trust problems. Our federated training approach ensures the privacy of the cooperating sites is maintained.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

We consider the case of a cooperative surveillance network as shown in Fig. 1 which comprises five groups of sites Ψ namely homes Ψ_h, offices Ψ_o, shopping malls Ψ_m, roads Ψ_r, and stations Ψ_s, i.e. $\Psi_h, \Psi_o, \Psi_m, \Psi_r, \Psi_s \in \Psi$. Each group comprises of sites with similar requirements and environments i.e. $\Psi_{h1}, \Psi_{h2}, \Psi_{h3}, \ldots, \Psi_{hn} \in \Psi_h$ where Ψ_{h1} is the first home site among n homes. Each site in a group has a set C of C heterogeneous devices (HDs) such as drones, cameras, sensors, etc., and one edge node (EN) connected to a control center via a communication network. For instance, devices in the first home site Ψ_{h1} are denoted as $C = (c_{1h1}, c_{2h1}, c_{3h1}, \ldots, c_{|nh1|})$ whereas the EN through which these devices communicate is denoted as E_{h1}. The number and types of devices in each site vary according to the size and security requirements of the location. Regularly, or when an anomaly is reported, onsite security personnel check and review the feeds for incidents of interest. In existing systems, detected feeds are tagged and kept for record or investigation purposes. However, in our proposed system, detected feeds are tagged and used to train a lightweight anomaly detection model (ADM) that is shared across all sites in a group. The trained ADM is then sent to the edge node for aggregation and transmission to other sites. This approach allows member sites to learn about the incident and prevent the occurrence of such incidents on their premises.

Furthermore, the system ensures the privacy of cooperating sites because no data is uploaded apart from the trained model. The described network has three data processing domains which include, end nodes (via the use of an agent), edge nodes, and the cooperator node (CN). The HDs have limited storage/computational capability and are used majorly to collect data, train local ADM, and detect anomalies in real-time.

Due to their remote installations, some HDs are assumed to be energy and bandwidth-constrained. Furthermore, depending on the state of their resources (memory, energy, and network), HDs can perform computations or offload to the ENs. Inferences and alerts must be generated from the collected feeds in real-time with zero tolerance for latency and failure. This ensures prompt and preemptive actions are taken by security personnel to prevent/stop unwanted incidents.

Fig. 1. The layout of the cooperative surveillance network.

B. Computation and Transmission Model

The system model consists of a cooperator node (CN) and multiple edge nodes. An edge node E_{yz} serve as a gateway for devices on the site Ψ_{yz} where y denotes the group and z is the site's position in the group. A group y is a collection of sites in a similar location or with similar requirements. Thus, device C_{xyz} denotes the x^{th} node in the z^{th} site of group y. Each end node has a local training data sample D_{xyz} and uses a size s_{xyz} of the samples to train its local ADM.

The weight of the local data samples D_{xyz} as compared to the total data sample is calculated as

$$\varsigma_{total}^{xyz} = \frac{s_{xyz}}{s_{total}} \tag{1}$$

Each data sample consists of an input-output pair, in which the input is a captured video, and the output is the label value indicating the incident in the video. The total computation resources for local model training, i.e., CPU cycle frequency, for each node C_{xyz} is denoted as f_{xyz}. The number of CPU cycles required for a node C_{xyz} to process one sample of data during the local model training is denoted as h_{xyz}. Hence, for any node C_{xyz}, the computation time of a local iteration is [38]

$$T_{c_{xyz}}^{comp} = \frac{h_{xyz}.s_{xyz}}{f_{xyz}} \tag{2}$$

The CPU energy consumption for the node for one local iteration is expressed as:

$$P_{C_{xyz}}^{cmp}\left(f_{xyz}\right) = \eth\, h_{xyz} s_x f_{xyz}^2 \tag{3}$$

where $\eth$ is the effective capacitance parameter of the computing chipset for each node [39]. This paper uses a layered approach that allows model aggregation to take place on the edge nodes and at the cooperator node (CN). During the edge update, denoted as $U_{E_{yz}}$, a node $C_{xyz} \in E_{yz}$ sends an update to the edge node E_{yz} via a communication channel.

We note that the quality of each local model update from node C_{xyz} is affected by its local data quality, which is denoted as q_c, with a value normalized to a range, where a high value signifies a better data quality. Let $\log(1/q_c)$ represent the number of iterations of a local model update when the global accuracy is fixed. The transmission rate for the node C_{xyz} and E_{yz} is denoted as [38]

$$r_{c_{xyz}} = B\ln(1 + \frac{\phi_c \Phi_{cxyz-Eyz}}{N_0}) \tag{4}$$

$$r_{E_{xyz}} = B\ln(1 + \frac{\phi_E \Phi_{Eyz-F}}{N_0}) \tag{5}$$

where B is the transmission bandwidth and ϕ_c and ϕ_E is the transmission power of the node C_{xyz} and E_{yz} respectively. $\Phi_{cxyz-Eyz}$ and Φ_{Eyz-F} is the channel gain of the link between the node C_{xyz}, edge node E_{yz}, and cooperator node F. N_0 is the background noise. If the data size of a local model update is σ, then the transmission time and energy consumed during a local model update with the data size of σ is expressed as

$$T_{c_{xyz}}^{trans} = \frac{\sigma}{B\ln(1 + \frac{\phi_c \phi_{cxyz-Eyz}}{N_0})} \tag{6}$$

$$P_{c_{xyz}}^{trans} = \frac{\sigma \phi_c}{B\ln(1 + \frac{\phi_c \phi_{cxyz-Eyz}}{N_0})} \tag{7}$$

The total time $T_{C_{xyz}}^{total}$ of one local iteration for a node c_{xyz} is the sum of the computation time $T_{c_{xyz}}^{comp}$ and the transmission time $T_{c_{xyz}}^{trans}$.

$$T_{c_{xyz}}^{total} = \log\left(\frac{1}{q_c}\right)[\frac{h_{xyz}.s_i}{f_{xyz}} + \frac{\sigma}{B\ln(1 + \frac{\phi_c \phi_{cxyz-Eyz}}{N_0})}] \tag{8}$$

For a global iteration, the total energy consumed by a node c_{xyz} is denoted as

$$P_{c_{xyz}}^{total} = \log(\frac{1}{q_c})(P_{c_{xyz}}^{cmp} + P_{c_{xyz}}^{trans}) \tag{9}$$

C. Problem Formulation

One major problem to be addressed by the system is how to make surveillance devices detect anomalies without a prior experience of the incident and how the privacy of cooperating nodes is preserved. The cooperating sites must collaboratively train a model in an approach that accommodates the resource limitations of participating devices while ensuring model convergence is achieved. Thus, we formulate this problem as an asynchronous federated learning task in a cooperative

surveillance network with heterogeneous requirements. The goal of each end node C_{xyz} is to use its dataset D_{xyz} to train a local anomaly detection model (ADM) w_{xyz}. Each node is to find parameters that minimize the loss function defined in (10) and (11). i.e.

$$w_{xyz} = \underset{w_{xyz}}{\operatorname{argmin}} L\left(w_{xyz}, u_k, \hat{v}_k\right) \tag{10}$$

where,

$$L\left(w_{xyz}, u_k, \hat{v}_k\right) = -\sum_{k=1}^{T} \left[u_k \log(\hat{v}_k) + (1-u_k)\log(1-\hat{v}_k)\right] \tag{11}$$

and each dataset D_{xyz} contains multiple training data samples u_k and $\hat{v}_k$. u_k is the input data sample while $\hat{v}_k$ is the corresponding output label. To reduce the computational burden on the resource-constrained devices, and further provide a means for measuring the privacy guarantees provided by the system. we use differentially private stochastic gradient descent (DP-SGD) to optimize the loss function [40]. DP-SGD uses a twin technique of gradient clipping and noising to mitigate the risk of exposing sensitive training data. In the t^{th} epoch, the model w_{xyz} is sent to the edge node E_{yz} for weighted averaging computation as

$$w_{yz}^{t} = (1-\theta)w_{yz}^{t-1} + \theta w_{xyz} \quad \forall x \in z \tag{12}$$

whereas the averaging at the CN is computed as

$$w_{y}^{t} = (1-\theta)w_{y}^{t-1} + \theta w_{yz} \quad \forall z \in y \tag{13}$$

where $\theta \in (0, 1)$ is the mixing hyperparameter. Thus, the goal of the edge node E_{yz} is to aggregate a local model w_{yz} $\forall$ x $\in z$ whereas the CN aggregates a global model for $\forall$ z $\in y$.

Further, the end nodes must be capable of real-time anomaly detection. This second problem is formulated as a multiclass classification task that produces a vector $\hat{v}_k$ with $\hat{v}_k = P(\hat{v}_k = i|u_k)$. It is required each element of $\hat{v}_k$ be between 0 and 1, and that the vector sums to 1 i.e. $\sum_{k=1}^{i} p(\hat{v}_k = i|u_k) = 1$ and i is the output value for each class. To achieve this, a linear layer predicts unnormalized log probabilities as

$$z = W^T h + b \tag{14}$$

where

$$z_i = \log P((v_k = i|u_k) \tag{15}$$

The softmax function can then exponentiate and normalize z to obtain the desired $\hat{v}_k$. Formally, the softmax function is given as

$$softmax(z_i) = \frac{exp(z_i)}{\sum_i exp(z_i)} \tag{16}$$

IV. The SurveilNet System

A. Architecture

A simplified architecture of the system is shown in Fig. 2. In this architecture, a cooperator node coordinates the system operations at the cloud layer while the edge node coordinates the operations at the local network layer. The operation of each device on a local network is managed by an agent running on a separate host attached to the device. Such operations include device configuration, video preprocessing, feature extraction, model training, inference, alert generation, and interfacing with the edge node. Apart from managing the operation of the camera, the agent also interfaces with the edge device connected to all cameras on the network as shown in Fig. 2.

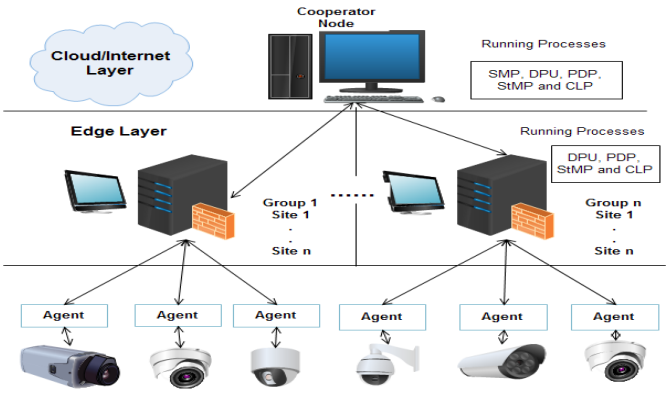


Fig. 2. SurveilNet system architectural framework.

The CN coordinates the communication flow between participating edge devices i.e. inter edge communication. It also manages the subscription of participating sites and ensures updates are received, aggregated, and forwarded promptly. The edge nodes, on the other hand, manages the model updates and forwarding process in the local edge network i.e. inter-device communication. The communication approach adopted in the system ensures nodes are only aware of the edge node and not the CN. This mode eliminates the need for constant authentication and security key exchanges which makes the communication process extremely lightweight.

Thus, to maintain the integrity of the communication process, a one-time authentication is carried out and maintained throughout the scheme. Five major processes run on the CN and edge nodes. They include subscription management, dynamic parameters update (DPU), parameter distribution, status monitor, and continuous learning. The subscription management process (SMP) executes on the CN and it maintains the database shown in Table III. When necessary, it adds, deletes, or updates the data stores in the database. The dynamic parameters update process (DPU) on the other hand, calculates and updates the global parameter weights according to the information received from the status monitor. The parameter distribution process (PDP) queries the database for a list of sites subscribed to a model and subsequently forwards the updated model parameters to the subscribed sites after aggregation. The status monitors (StMP) process monitors the performance of various devices in the architecture whereas the continuous learning process prevents model staleness via a periodic model update or after a false positive or negative result.

B. A Lightweight Subscription Scheme With Continuous Learning Capability

In this scheme, sites can only subscribe to the model updates of their group. This results in a fast aggregation and update

TABLE II
NOTATIONS USED WITHIN THE SCHEME

Notation	Description
l	Anomaly list
$\mathcal{L}$	Learning rate
$\hat{J}$	Loss Function
λ	Optimizer
e	Event
w^t_{xyz}, w^t_{yz}, w^t_y	Model weights at the local, edge, and cooperator node at time t
ss	Subscription status
Ack_1, Ack_2	Acknowledgments received by the CN and EN from the edge and end nodes respectively
w^{e*}_y, w^{t*}_y .	Global model weights after anomaly e or time t respectively

TABLE III
DATABASE MAINTAINED AT THE COOPERATOR NODE

Model	Subscribed Sites	Anomaly list, l	$\mathcal{L}$	$\hat{J}$	λ
w_{ψ_h}	Home 1-5	Ab, Ars, Bg, Nml, Vlc	0.001	SCL	SGD
w_{ψ_o}	Office 1-5	Ab, Ars, Bg, Nml, Vlc	0.001	SCL	SGD
w_{ψ_m}	Shopping mall 1-5	Bg, Nml, Plt, Slt, Vlc	0.001	SCL	SGD
w_{ψ_s}	Station 1-4	Ast, Rob, Nml, Vlc	0.01	SCL	SGD
w_{ψ_r}	Roads 1-2	Ars, Ast, Nml, Acd	0.001	SCL	SGD

Ab-Abuse, Ars-Arson, Bg-Burglary, Nml-Normal, Vlc-Violence, Plt-Parking lot, Slt-Shoplifting, Ast-Assault, Rob-Robbery, Acd-Accident

process at the CN because there are no queries for external sites subscribed to the update. However, a major challenge with this mode of operation is the time difference between the download and upload of model updates. This difference is a function of the processing capability of the devices as defined in (2) [41]. Thus, to account for the difference, the DPU evaluates a staleness value which is used to adjust the weight of the uploaded model. The staleness value is defined as

$$\mathfrak{F}_{staleness} = \mathfrak{f}_{downloaded} - \mathfrak{f}_{uploaded} \tag{17}$$

Thus, using the staleness value, the mixing hyperparameter is calculated as

$$\theta_t = \theta \times \mathfrak{F}_{staleness} \tag{18}$$

where, $\theta \in (0, 1)$. Fig. 3 presents the communication flow within the architecture whereas algorithms 1 and 2 provide detailed steps of the subscription scheme and the continuous learning process.

To commence operation, the DPU (CN version) initializes several models with a timestamp (w^0_y, t) that corresponds to the number of groups managed by the CN. Subsequently, the PDP queries the database and forwards the initialized models to the edge nodes (EN) along with a set of corresponding parameters shown in Table III.

These parameters are needed for the consistency of the training process across the various devices. When received at the EN, the PDP (EN version) acknowledges the CN and

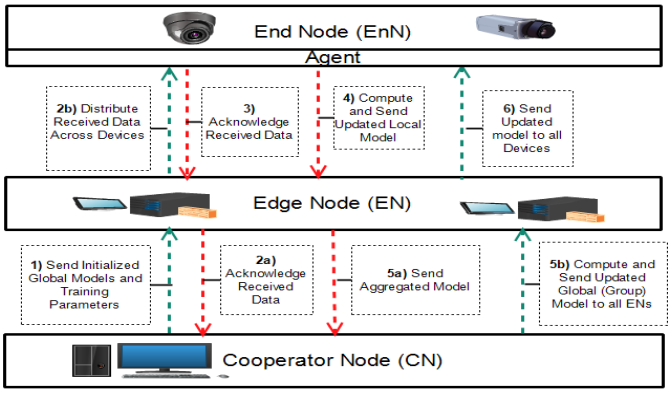


Fig. 3. Communication flow within the architecture.

forwards the received data to all devices in its local network. If an acknowledgment is not received from the EN after three resend attempts, the SMP removes the EN from the list of subscribed sites. The node C_{xyz} updates the received model using its local dataset D_{xyz} and sends the updated model with the completion timestamp (w^{t+1}_{xyz}, t_{comp}) to the EN E_{yz}. The goal of the node C_{xyz} during the update is to find optimal parameters that minimize the loss function defined in (10) and (11).

Afterward, the DPU (EN version) aggregates the models received from the various nodes as defined in (12). When completed, the PDP (EN version) broadcast the aggregated model (w^{t+1}_{yz}, t_{comp}) to all devices on the local network (temporary) and the CN. The DPU (CN version) receives and aggregates the models as defined in (13). Subsequently, the PDP (CN version) forwards the aggregated models (w^{t+1}_{y}, t_{comp}) to all ENs that are subscribed to the W_y update. The described steps are repeated until the global loss function converges or a desirable training accuracy is achieved. After convergence, the continuous learning process (CLP) is activated. The CLP updates the model when a false classification is reported by the StMP or after a predefined time interval. A major challenge with CL is the tendency to lose previously learned knowledge during the update process. Thus, to prevent this, we adopt an approach that selectively slows down learning on the previously learned weights [42]. We initialize a temporary weight w^{temp}_{xyz} from the existing weight w^T_{xyz}. The new dataset D^{new}_{xyz} is then used to update the temporary weight by computing (10) and (11).

Afterward, a weighted sum is used to update the weight of each class $w^T_{xyz(i)}$ [43] as

$$w^{T+1}_{\mathrm{xyz}(i)} = \frac{(w^T_{\mathrm{xyz}(i)} \times w^i_{\mathrm{samples}}) + (w^{temp}_{xyz(i)} - avg(w^{temp}_{\mathrm{xyz}})}{w^i_{\mathrm{samples}} + 1} \tag{19}$$

where $w^i_{\mathrm{samples}} = \sqrt{\frac{samples^T_i}{samples^{T+1}_i}}$, $samples^T_i$ is the total sample of class i encountered in past batches whereas $samples^{T+1}_i$ is the currently encountered samples.

V. SYSTEM EVALUATION

Based on the proposed system, we develop five specialized anomaly detection models (ADM) that classify recorded

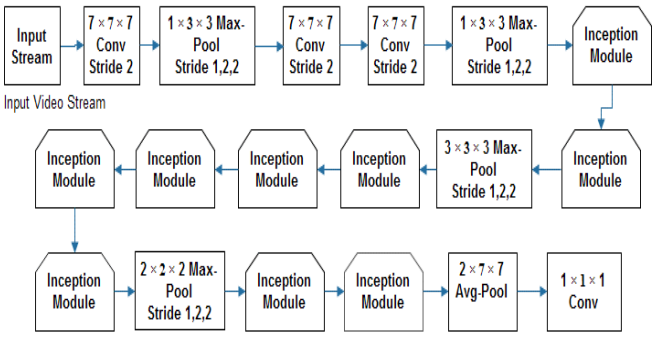


Fig. 4. The architecture of the I3D video feature extractor.

anomalies in real-time. The experiments were conducted on a computer with an Intel Core i5-7500 CPU Processor and a speed of 3.40 GHz. It has a 16GB DDR4 and runs Ubuntu 20.04 as the operating system. Software used include Keras, NumPy Python, and TensorFlow. Each model consists of two parts, a CNN-based extraction layer, and a detection layer. The first layer extracts feature from the video feed whereas the second layer classifies the features into various anomaly categories.

In the following subsections, we briefly describe the model architecture, dataset parameters, and training process.

A. Model Architecture (Feature Extractor)

For each ADM, we adopt the Inflated 3D (I3D) ConvNets [44] that uses 2D ConvNet inflations of filters and pooling kernels as a feature extractor. The architecture shown in Fig. 4 uses a stride of 2 in the first convolutional layer. Also, there are four max-pooling layers with a stride of 2 and a 7×7 average-pooling layer preceding the last linear classification layer, besides the max-pooling layers in parallel inception branches. The input videos are processed at 25 frames per second; The final average pooling layer uses a $2 \times 7 \times 7$ kernel. The convolutional layer uses a rectified linear unit (ReLU) as activation functions followed by a max-pooling layer.

B. Dataset

For this paper, we reclassify the UCF-Crime dataset [27] into five groups according to the location and action that is carried out in the video. The reclassified dataset is available for download on IEEE Dataport [28]. The groups include home, office, shopping mall, road, and station. The dataset groups are used to train five specialized ADMs. The Home model is useful for old people's homes, personal homes, whereas the office model is useful for offices and the general business environment. Table IV presents the number of training videos for each group and their distribution across the participating nodes. The mall model is useful for detecting anomalies in shopping malls and stores. The road model detects anomalies on public roads whereas the station model detects anomalies in public stations such as bus and train stations. The dataset consists of 752 training video sequences of anomaly videos and 539 normal video sequences. Each sequence has a minimum of 9000 frames (30 seconds), with a 320×240 resolution at a frame rate of 30 frames per second.

Algorithm 1 A Lightweight Subscription Scheme With Continual Learning Capability

function SurveilNet//*To be executed on the cooperator node (CN)*
//Five Processes running on the CN: SMP, DPU, PDP, StMP & CLP
Input: Define P, where l, $\mathcal{L}$, $\hat{J}$, λ, $\theta \in P$ (see Table III)
Output: Global model parameters, w_{y}^{T}, and w_{y}^{e} *//e-event, T-time*
DPU initializes several models (w_{G}^{0}, t) $\forall$ y $\in \Psi$
PDP forwards (w_{y}^{0}, t), *P to* $E_{yz}, \forall$ z $\in$ y
If Ack_1 is not received from any E_{yz} after three attempts, *SMP* updates Table III (suspend/remove the edge node)
Else await w_{yz}^{t+1}, $\forall$ z $\in y$
for each received w_{yz}^{t+1}, at |t $\in$ T| **do**
DPU computes $w_{y}^{t+1} = (1-\theta_t) w_{yz}^{t-1} + \theta_t w_{\mathrm{yz}}^{t}, \forall$ z $\in y$
// θ_t *is defined in (18)*
PDP forwards (w_{y}^{t+1}, t) *to* $E_{yz}, \forall$ z $\in$ y for onward transmission to all nodes on the local network
end for each
Repeat until time T *or when* a desirable training accuracy is achieved.
Activate *CLP, StMP*
//
function SurveilEdge *//To be executed on the edge nodes* E_{yz}
//Four Processes running on the EN: DPU, PDP, StMP & CLP
Input: l, $l\mathcal{L}$, $\hat{J}$, λ, θ, w_{y}^{0}
Output: w_{yz}^{t}, w_{yz}^{e} *//e-event, T-time*
PDP sends $\mathbf{Ack_1}$ to the CN after receiving **inputs**
PDP sends received data to all connected devices $\forall$ x $\in z$
for each received w_{xyz}^{t}, at |t $\in$ T| **do**
DPU computes $w_{yz}^{t+1} = (1-\theta_t) w_{xyz}^{t-1} + \theta_t w_{\mathrm{xyz}}^{t}, \forall$ x $\in z$
PDP forwards (w_{yz}^{t+1}, t) *to CN*
Await aggregated model w_{y}^{t+1} from CN
Repeat until time T *or when* a desirable training accuracy is achieved.
Activate *CLP, StMP*
//
function SurveilNode *//To be executed by the agent for each node*
Input: l, $l\mathcal{L}$, $\hat{J}$, λ, w_{yz}^{0}
Output: w_{xyz}^{t}. or w_{y}^{e} *//e-event, t-time*
Send $\mathbf{Ack_2}$ after receiving **inputs** from the EN
at |t $\in$ T| do
compute (9)
if $P_{c_{xyz}}^{total}$ < Node remaining energy, offload training to EN compute $w_{\mathrm{xyz}}^{t} = \underset{w_{\mathrm{xyz}}}{\operatorname{argmin}} L\left(w_{xyz}, u_k, \hat{v}_k\right)$ using local dataset D_{xyz} send the w_{xyz}^{t} to EN E_{yz}.
Wait for the updated model w_{yz}^{t}
Repeat until time T *or when* notified by the EN
Activate CLP, *StMP*

TABLE IV
TRAINING DATASET PARAMETERS

Model	No of Videos	Training	
		Nodes	Sites
Home	185	15	5
Office	296	15	5
Mall	343	15	5
Station	146	15	5
Road	321	15	5

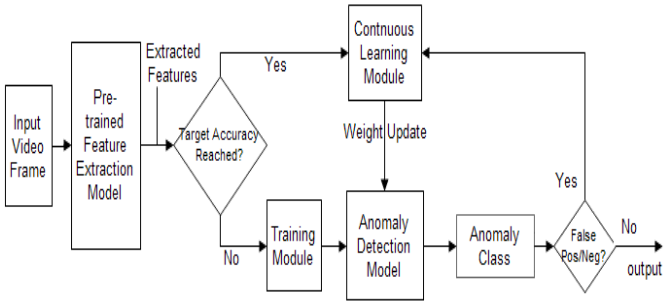


Fig. 5. The anomaly detection training process.

C. Training and Deployment

Due to the resource-constrained nature of the surveillance devices, it is usually difficult to train a CNN model from scratch especially models with large datasets. Thus, we leverage a Transfer Learning (TL) technique. With TL, a smaller node can adapt a pre-trained model to a new or similar task. We use the pre-trained I3D model to extract features from the surveillance videos and further freeze the convolution layers of the feature extractors to ensure the weights of the pre-trained convolution layers are not destroyed. Otherwise, the backpropagated error via the randomly initialized classifier layers will be too large, which makes the model more difficult to train. The weights of the classifier are initialized using the HE technique [45] as

$$W^{[l]} = n \times \sqrt{\frac{2}{size^{[l-1]}}} \tag{20}$$

where size is the network size, l is the network layer, and n is a random value between the size of the current layer and its previous layer. This allows each ADM to converge faster and avoid the chances of exploding or vanishing gradients [46]. The classifier uses the SoftMax activation plus Cross-Entropy loss (SCEL) which trains a CNN to output a probability over a set of classes for each input feature (multiclass classification). The output vector represents the predicted probabilities of all classes which sum up to 1. Precisely, the objective of the training is to classify the extracted features into a set of possible classes by minimizing the loss function defined in (10) and (11) using a SoftMax activation function $\Phi(z)$ defined in (16). The output of the classifier is fed through a monitor process that checks for its accuracy. Fig. 5 illustrates the ADM training process. After training, a continuous learning process updates the weights of the ADM periodically or when a false output is detected by the monitor process.

Algorithm 2 Continual Learning Process

Input: l, $l\mathcal{L}$, $\hat{J}$, λ, w_{yz}^{T}
Output: w_{xyz}^{T+1}
//Two modes of operation: periodic update and false output
if mode == periodic update,
For each new training data D_{xyz}^{new}
Initialize temporary weight $w_{\text{xyz}}^{\text{temp}}$ from existing model w_{xyz}^{T}
Compute $w_{\text{xyz}}^{temp} = \underset{w_{\text{xyz}}}{\text{argmin}}\, L\left(w_{\text{xyz}}^{temp}, u_k, \hat{v}_k\right)$ using D_{xyz}^{new}
For each class i in D_{xyz}^{new};
Compute (19)
Send the w_{xyz}^{t} to EN E_{yz} for distribution across the network
else // false output triggered update,
fetch the correctly labeled data from the status monitor $D_{xyz}^{correct}$
Initialize temporary weight $w_{\text{xyz}}^{\text{temp}}$ from existing model w_{xyz}^{T}
Compute $w_{\text{xyz}}^{temp} = \underset{w_{\text{xyz}}}{\text{argmin}}\, L\left(w_{\text{xyz}}^{temp}, u_k, \hat{v}_k\right)$ using $D_{xyz}^{correct}$
For the wrongly predicted class i in $D_{xyz}^{correct}$;
Compute (19)
Send the w_{xyz}^{t} to EN E_{yz} for distribution across the network

VI. RESULT AND DISCUSSION

To evaluate the performance of our proposed system, we conducted experiments on our system model. Each group has a default of 15 end nodes and 5 edge devices i.e. 5 sites in a group with 3 devices per site. All nodes have local datasets on which model training is conducted. The first set of experiments investigate the effectiveness of the system for large-scale deployment i.e. the effect of an increased number of nodes on the system's performance and the effect of insufficient data on the scheme's performance. The second set of experiments focuses on the operational performance of the system while the third set of experiments investigates the continuous learning capability of the system.

A. Scalability Result

By default, the dataset was distributed across 15 devices. Thus, it becomes necessary to investigate the consistency of the scheme's performance across an increased number of nodes and datasets. Based on the initial assessment of the five model's performance, the office model was selected as a reference model for the evaluation. Firstly, the nodes are increased from 15 to 120 with a step of 15. The effect of the variation is observed on the training loss and accuracy.

From Fig. 6, it is observed that the convergence period is directly proportional to the number of nodes participating in the training process up to the 75$^{\text{th}}$ node. For instance, the shortest convergence time was observed when the nodes were 15 and 30.

However, the convergence period slowly increased as the number of nodes also increased. The gradual increase however peaked at the 90$^{\text{th}}$ node. It is observed that after this period, the convergence period was the same for the 105$^{\text{th}}$ and 120$^{\text{th}}$ node variation. Intuitively, this is due to the increased data diversity between nodes as the node increases and the time

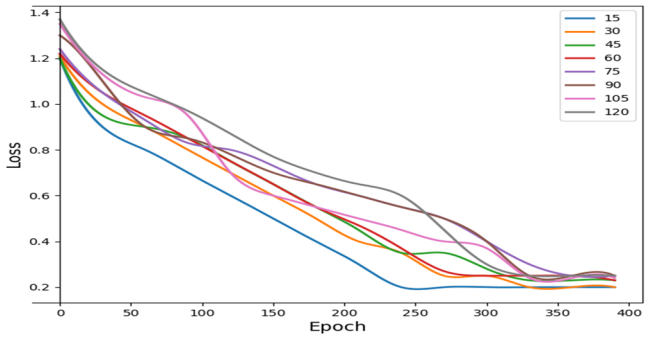


Fig. 6. Training loss at different node variation.

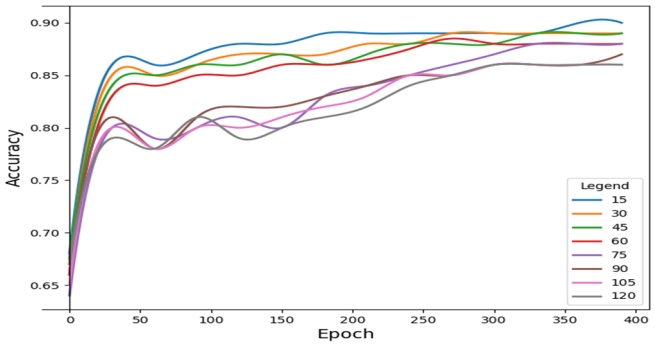


Fig. 7. Training accuracy at different node variation.

taken for the model updates to be sent across nodes. The consistent convergence period observed after the 90^{th} node is due to the layered computation mode that was triggered at the 75^{th} node. This mode ensures models are aggregated on the edge nodes before transmission to the cooperator node (CN) to reduce computational burden. From this result, we draw conclusion that the system is stable and can support an increasing number of nodes. A similar trend is observed in Fig. 7 which shows the accuracy obtained during training. It is observed that a faster accuracy is achieved between the 10^{th} and 60^{th} nodes variation. A wide accuracy gap is however observed between the nodes below the 60^{th} variation and the node above the 75^{th} variation. A better training accuracy of $\approx$0.9 is also observed for nodes below the 60^{th} variation. The nodes above the 75^{th} variation had a training accuracy of $\approx$0.85.

Secondly, we investigate the effect of dataset variation on the system's performance. To do this, the dataset is partitioned into three $(1/3, 2/3, 3/3)$ while keeping the nodes constant at 15. This investigation aims to study the effect of limited data on the scheme's performance. The effect of the three data partitions on the training loss and accuracy is observed and shown in Fig. 8 and 9.

When using a third of the dataset, it is observed that the model failed to converge. This challenge can be mitigated via the use of bootstrapped training data at the commencement of the system's operation. The system however maintained a consistent performance at a $2/3, 3/3$ dataset partition. This result showed the effectiveness of the system when faced with a relatively insufficient data sample.

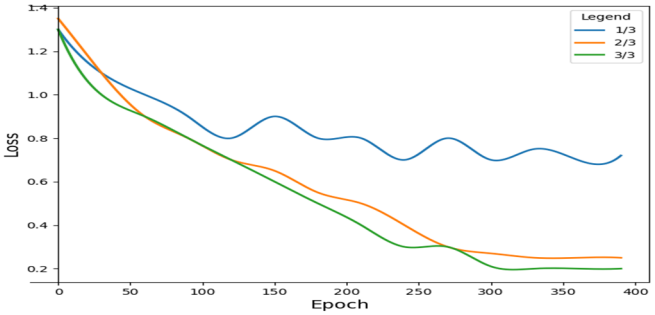


Fig. 8. Training loss at different dataset variation.

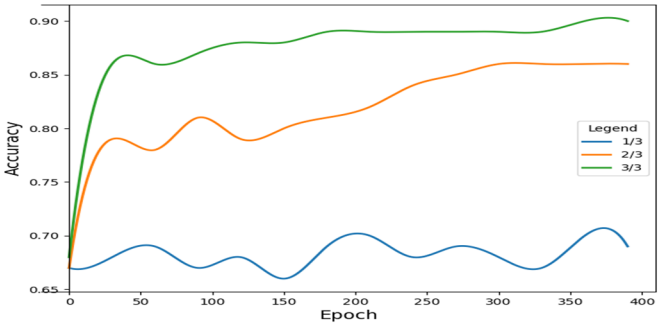


Fig. 9. Training accuracy at different dataset variation.

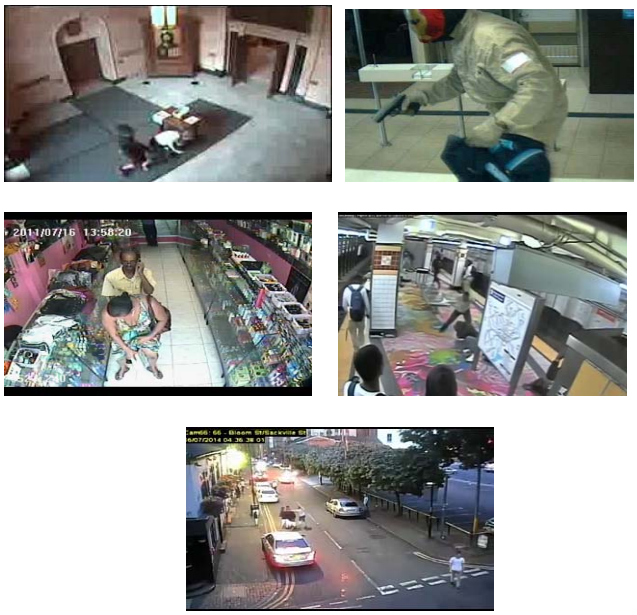


Fig. 10. Extracted anomaly frames from test videos in the five dataset groups.

B. Performance Result

We use a confusion matrix to show the combination of the actual and predicted classes. Each row represents the instances in a predicted class, while each column represents the instances in an actual class. It is a good measure of whether models can account for the overlap in class properties and understand which classes are most easily confused. A confusion matrix is a tabular way of visualizing the performance

TABLE V
CONFUSION MATRIX FOR THE HOME MODEL w_{Ψ_h}

Predicted Class \ True Class	Abuse	Arson	Burglary	Normal	Violence
Abuse	**3**	1	0	0	0
Arson	0	**4**	1	0	1
Burglary	1	1	**3**	0	1
Normal	1	0	0	**3**	0
Violence	0	1	0	0	**1**

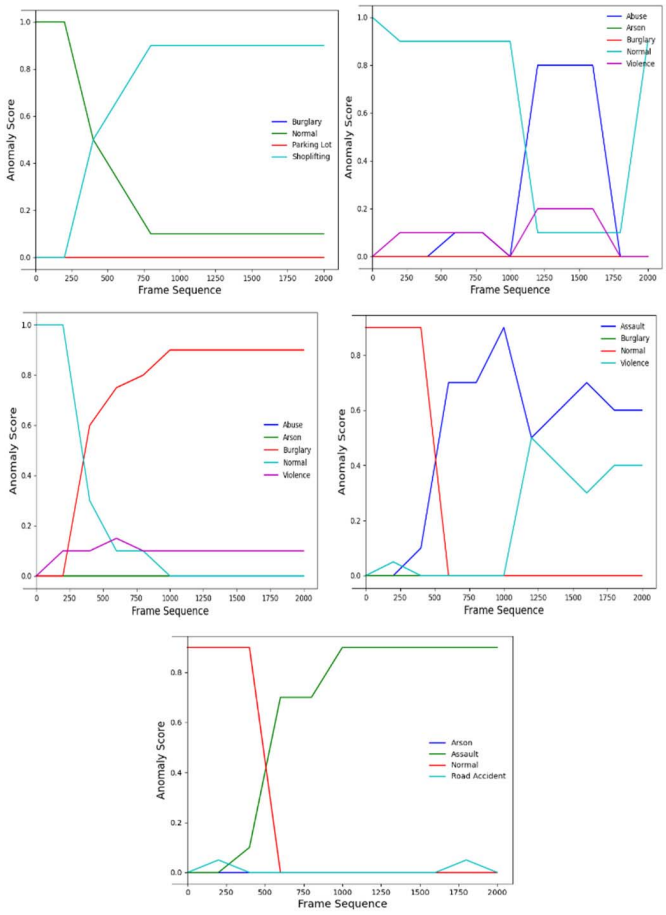


Fig. 11. Anomaly score for the test frames in Fig. 10.

of a prediction model. Each entry in a confusion matrix denotes the number of predictions made by the model where it classified the classes correctly or incorrectly. Fig. 10 shows five extracted anomaly frames from test videos in the five dataset groups while Fig. 11 shows the corresponding anomaly scores for each frame. Table V-IX presents the confusion matrix for each model at a threshold score of 0.6 i.e. a score within the range of 0.0 - 0.59 is recorded as a negative outcome (0) whereas a score within the range of 0.6 – 1.0 is recorded as a positive outcome (1).

The confusion matrices are used to generate a classification report (CR). A CR has three main metrics of precision, recall, F1. These metrics are calculated using true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) values.

TABLE VI
CONFUSION MATRIX FOR THE OFFICE MODEL w_{Ψ_O}

Predicted Class \ True Class	Abuse	Arson	Burglary	Normal	Violence
Abuse	**2**	0	0	0	0
Arson	0	**2**	0	0	0
Burglary	0	0	**1**	0	1
Normal	0	0	0	**3**	0
Violence	1	0	0	0	**1**

TABLE VII
CONFUSION MATRIX FOR THE SHOPPING MALL MODEL w_{Ψ_m}

Predicted Class \ True Class	Burglary	Normal	Parking	Shoplifting	Violence
Burglary	**1**	0	0	0	1
Normal	0	**1**	0	1	0
Parking Lot	0	0	**1**	0	0
Shoplifting	0	0	0	**1**	0
Violence	1	0	0	0	**1**

TABLE VIII
CONFUSION MATRIX FOR THE STATION MODEL w_{Ψ_s}

Predicted Class \ True Class	Assault	Normal	Robbery	Violence
Assault	**2**	1	0	0
Normal	0	**2**	0	0
Robbery	1	0	**2**	1
Violence	0	0	0	**1**

TABLE IX
CONFUSION MATRIX FOR THE ROAD MODEL w_{Ψ_r}

Predicted Class \ True Class	Arson	Assault	Normal	Road Accident
Arson	**1**	0	0	0
Assault	0	**2**	0	0
Normal	0	1	**1**	0
Accident	0	0	0	**1**

The precision metric measures the ability of a model to correctly predict an outcome whereas the recall metric evaluates what fraction of all positive samples were correctly predicted as positive by the classifier. The F1-score is the harmonic mean of precision and recall.

The three metrics can be calculated using (21), (22), and (23) while the CR for each model is presented in Table X-XIV.

$$Precision = \frac{TP}{(TP + FP)} \tag{21}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{22}$$

$$F1 = \frac{2 * TP}{(2TP + FP + FN)} \tag{23}$$

For the Home model, the precision was 0.64 which indicated that 64% of the prediction made by the model was correctly identified as a positive outcome. The recall value of the model has a slightly better value of 0.66 which indicated that 66% of all positive samples were correctly predicted as positive by the model. The office model, however, had a better performance

TABLE X
CLASSIFICATION REPORT FOR THE HOME MODEL w_{Ψ_h}

Metrics	Precision	Recall	F1
Abuse	0.75	0.60	0.67
Arson	0.67	0.60	0.63
Burglary	0.50	0.75	0.60
Normal	0.75	1.00	0.86
Violence	0.50	0.33	0.40
Average	0.64	0.66	0.63

TABLE XI
CLASSIFICATION REPORT FOR THE OFFICE MODEL w_{Ψ_o}

Metrics	Precision	Recall	F1
Abuse	1	0.67	0.80
Arson	1	1	1.0
Burglary	0.50	1	0.67
Normal	1	1	1.0
Violence	0.50	0.5	0.5
Average	0.80	0.83	0.80

TABLE XII
CLASSIFICATION REPORT FOR THE MALL MODEL w_{Ψ_m}

Metrics	Precision	Recall	F1
Burglary	0.5	1.0	0.67
Normal	0.5	1.0	0.67
Parking Lot	1.0	1.0	1.0
Shoplifting	1.0	1.0	1.0
Violence	0.5	0.5	0.5
Average	0.7	0.9	0.77

TABLE XIII
CLASSIFICATION REPORT FOR THE STATION MODEL w_{Ψ_s}

Metrics	Precision	Recall	F1
Assault	0.67	0.67	0.67
Normal	1	0.67	0.80
Robbery	0.5	1	0.67
Violence	1	0.5	0.67
Average	0.79	0.71	0.70

TABLE XIV
CLASSIFICATION REPORT FOR THE ROAD MODEL w_{Ψ_r}

Metrics	Precision	Recall	F1
Arson	1	1	1.0
Assault	1	0.67	0.8
Normal	0.5	1	0.67
Accident	1	1	1.0
Average	0.88	0.92	0.87

with a precision value of 0.8 and a recall value of 0.83. The corresponding precision values for the shopping mall, station, and road models are 0.7, 0.79, and 0.88 whereas their recall values are 0.9, 0.71, and 0.92.

The results at this threshold score showed that the road model had the best performance among the evaluated models. Amongst these models, the home model's performance was the least. A possible reason for this outcome lies in the data used in the subscription scheme. We observed the video feeds were mostly dark as a result of the indoor environment. This can be confirmed with the better performance of the road model wherein the training videos were recorded in an outdoor environment. To enhance the home model's performance, the cooperator node can specify to subscribers that an appropriate lightning environment is one of the requirements for subscription to the home model or require the use of infrared cameras. The enforcement of high-quality training is another approach that can be used to enhance the model's quality. We further note that the threshold score can be varied according to the application requirement.

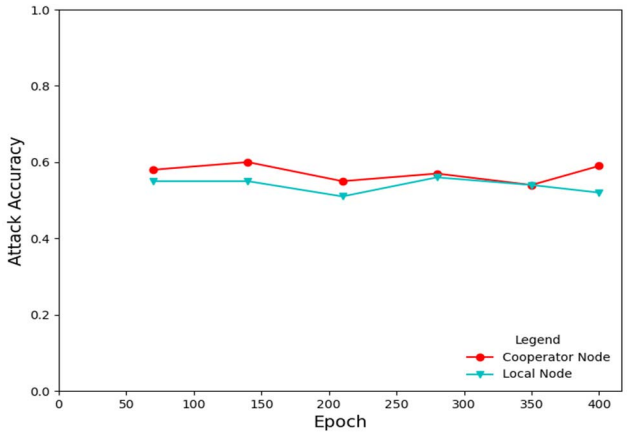


Fig. 12. Membership inference attack accuracy for the cooperator and local nodes at various training epochs.

TABLE XV
COMPUTATION PROFILE OF SOME TYPICAL LIGHTWEIGHT IOT APPLICATIONS

Metrics	CPU Flops (Million)	Parameters (Million)	Memory (MB)
IoTNet [48]	11	-	192
MCUNet [49]	168	1.20	1.0
EffNet [50]	43.7	-	-
SurveilNet	37.9	24.6	28

C. Computation Cost and Privacy Results

The system's ability to run on resource-constrained devices is evaluated by measuring the computation and communication cost as defined in equations (9). We note that the total cost for each device is obtained as the sum of the energy consumed during data computation $P_{c_{xyz}}^{comp}$ and transmission $P_{c_{xyz}}^{trans}$. Thus, during the training, evaluation, and inference process, we obtain [47] this information from the reference model as a function of the CPU floating-point operations, i.e., FLOPs, memory, and the number of model parameters transmitted in each iteration. We then compare the obtained result with the profile of three applications [48]–[50] developed for constrained IoT devices. The comparison shown in Table XV validates the low overhead requirements of the system and its ability to run on resource-constrained devices.

Further, in this system, we observe the potential for privacy leakage through parameters update that is shared across the network to potential adversaries (cooperator node and neighbor sites) during training and usage. Thus, we perform a privacy analysis of the system. To do this, we carry out a white-box membership inference attack on the local anomaly detection model at different training epochs. This attack

TABLE XVI
CLASSIFICATION REPORT FOR THE OFFICE MODEL w_{ψ_o} AFTER CL

Metrics	Precision	Recall	F1
Abuse	1	1	1.0
Arson	1	1	1.0
Burglary	0.67	1	0.8
Normal	1	1	1.0
Violence	0.67	0.67	0.67
Average	0.87	0.93	0.89

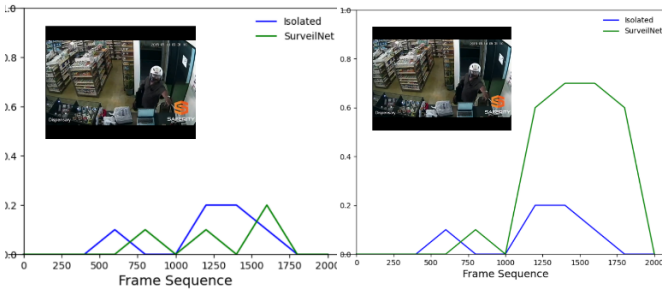


Fig. 13. Anomaly score for a burglary event in an isolated and SurveilNet system before and after continuous learning.

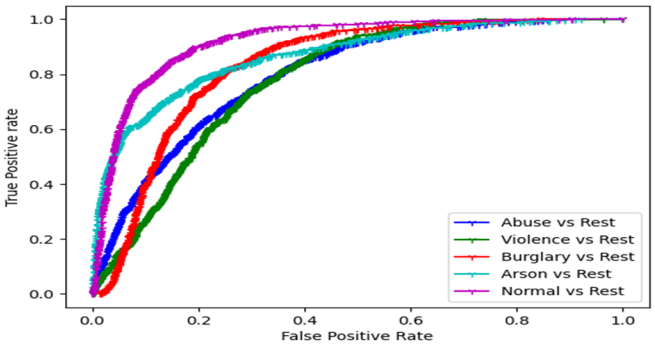


Fig. 14. ROC curve for our reference model.

TABLE XVII
COMPARISON OF AUC WITH THE STATE-OF-THE-ART

Papers	AUC (%)
Sultani, et al. [27]	75.4
Zhong, et al. [51]	82.12
Liu and Ma [52]	82
Zhu and Newsam [53]	79
This Paper	82.1

aims to infer if a particular surveillance feed was included in the training dataset. For this attack, we develop an attack model that is trained on the labeled inputs and outputs of a shadow model similar to the ADM at selected epochs. The experimental result is shown in Fig 12 and it reveals the attack accuracies carried out by the cooperator and local nodes at various epochs. We observe an average attack accuracy of 57.1% and 53.8% for the cooperator and local nodes respectively. Since a 50% accuracy indicates a random membership guess, we conclude that the system can preserve the data privacy of participating nodes.

D. Continuous Learning Result and Comparison With State-of-the-Art Isolated System

For the continuous learning capability test, the weight of the reference model (office) was updated using the test data that was falsely classified in the previous evaluation. As shown in Table XVI and when compared with the results earlier obtained in Table XI, the updated reference model performed better by accurately classifying the anomalies previously missed. The improved precision and recall values confirmed the efficiency of the CL process. We further compare SurveilNet with an isolated system. To do this, we isolate a site from our system and compare its detection accuracy with a connected site. Firstly, we observe that both sites failed to detect a previously unseen anomaly as shown in Fig. 13a. However, after the CL process of the system, the connected site detected the anomaly on the second attempt as shown in Fig. 13b. Secondly, using the same dataset and without CL, we observe no improved accuracy performance over the isolated system. Thus, we conclude that our system may provide no extra benefit in a restricted environment where anomaly situations rarely change.

We further use the frame-level area under the receiver operating characteristics (AUC) curve metric to compare the performance of our reference model with four state-of-the-art approaches i.e. Sultani, *et al.* [27], Zhong, *et al.* [51], Liu and Ma [52], and Zhu and Newsam [53]. The AUC measures the area underneath the ROC curve which provides an aggregate measure of performance across all possible classification thresholds. The ROC curve of our reference model is shown in Fig. 14 with an average AUC of 82.1%. This value competes favorably with the state-of-the-art as presented in Table XVII. The competitive score confirms the ability of the SurveilNet system to accurately detect anomalies. However, in addition to this, the proposed system extends the state-of-the-art anomaly detection approaches by providing the extra features of continuous learning and the ability to learn from the experience of other devices in a cooperative approach.

VII. CONCLUSION AND FUTURE WORK

The use of deep learning models for real-time analysis of surveillance feeds poses several challenges such as data insufficiency, privacy violations, outdated models, and the inability to learn from other sites. To address these issues, this paper presents a lightweight anomaly detection system (ADS) called SurveilNet. The solution provided is two-pronged. Firstly, we proposed a cooperative network setup that allows for resource sharing among neighbor sites. Unlike the existing isolated systems where sites are unable to learn from neighbor sites or update their model parameters, SurveilNet creates a network of related sites that allows sites to share resources without compromising privacy. Secondly, we propose a lightweight subscription scheme that allows sites to jointly train specialized anomaly detection models continuously. This approach ensures a member site learns from the anomaly experience in other sites thereby preventing a reoccurrence on its premises. Furthermore, our scheme uses a continuous learning technique to ensure the ADM used in the various sites do not become stale after a period.

To test the performance of the proposed system, we developed and evaluated five specialized ADM models and further compare their performance with the state-of-the-art approach. Analysis reveals that the performance of our

proposed system favorably competes with the existing approach in the literature. However, our system provides the extra advantage of specialization, periodic updates, and the ability to learn from other sites. Apart from its performance, another key potential of our system lies in its commercial applicability as it allows the development of specialized model targeted for a specific purpose such as schools, hospitals, roads, shopping malls, etc., hence, users interested in monitoring a location can subscribe to the relevant model and its updates.

For future work, we will explore the need to integrate solutions that addresses occlusion, blurring and scene switching challenges in surveillance videos into the system. Secondly, to further improve privacy, we aim to provide data encryption for the communication process between nodes.

REFERENCES

[1] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, and Y.-D. Yao, "An amateur drone surveillance system based on the cognitive Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 29–35, Jan. 2018.

[2] D. D. Olatinwo, A. Abu-Mahfouz, and G. Hancke, "A survey on LPWAN technologies in WBAN for remote health-care monitoring," *Sensors*, vol. 19, no. 23, p. 5268, 2019.

[3] S. Zhang, C. Wang, S. C. Chan, X. Wei, and C. H. Ho, "New object detection, tracking, and recognition approaches for video surveillance over camera network," *IEEE Sensors J.*, vol. 15, no. 5, pp. 2679–2691, May 2015.

[4] N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination of elderly," *IEEE Sensors J.*, vol. 12, no. 6, pp. 1965–1972, Jun. 2012.

[5] M. O. Osifeko, G. P. Hancke, and A. M. Abu-Mahfouz, "Artificial intelligence techniques for cognitive sensing in future IoT: State-of-the-art, potentials, and challenges," *J. Sensor Actuat. Netw.*, vol. 9, no. 2, p. 21, Apr. 2020.

[6] A. H. Sanoob, J. Roselin, and P. Latha, "Smartphone enabled intelligent surveillance system," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1361–1367, Mar. 2016.

[7] L. Zhou, K.-H. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and privacy for the industrial Internet of Things: An overview of approaches to safeguarding endpoints," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 76–87, Sep. 2018.

[8] V. Jelicic, M. Magno, D. Brunelli, V. Bilas, and L. Benini, "Benefits of wake-up radio in energy-efficient multimodal surveillance wireless sensor network," *IEEE Sensors J.*, vol. 14, no. 9, pp. 3210–3220, Sep. 2014.

[9] D. T. Ramotsoela, G. P. Hancke, and A. M. Abu-Mahfouz, "Attack detection in water distribution systems using machine learning," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, pp. 1–22, Dec. 2019.

[10] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study," *Sensors*, vol. 18, no. 8, p. 2491, 2018.

[11] B. Silva and G. P. Hancke, "Ranging error mitigation for through-the-wall non-line-of-sight conditions," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 6903–6911, Nov. 2020.

[12] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3846–3853, Oct. 2013.

[13] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[14] W. G. Hatcher and W. Yu, "A survey of deep learning: Platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018.

[15] N. Dawar and N. Kehtarnavaz, "Action detection and recognition in continuous action streams by deep learning-based sensing fusion," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9660–9668, Dec. 2018.

[16] U. A. B. U. A. Bakar, H. Ghayvat, S. F. Hasanm, and S. C. Mukhopadhyay, "Activity and anomaly detection in smart home: A survey," in *Next Generation Sensors and Systems*, S. C. Mukhopadhyay Ed. Cham, Switzerland: Springer, 2016, pp. 191–220.

[17] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: http://arxiv.org/abs/1901.03407

[18] W. Liu *et al.*, "Argus: Efficient activity detection system for extended video analysis," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW)*, Mar. 2020, pp. 126–133.

[19] B. LIU, "Lifelong machine learning: A paradigm for continuous learning," *Frontiers Comput. Sci.*, vol. 11, no. 3, pp. 359–361, 2017.

[20] F. Zhou, L. Wang, Z. Li, W. Zuo, and H. Tan, "Unsupervised learning approach for abnormal event detection in surveillance video by hybrid autoencoder," *Neural Process. Lett.*, vol. 52, no. 2, pp. 961–975, Oct. 2020.

[21] F. Landi, C. G. M. Snoek, and R. Cucchiara, "Anomaly locality in video surveillance," 2019, *arXiv:1901.10364*. [Online]. Available: http://arxiv.org/abs/1901.10364

[22] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[23] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "A trustworthy privacy preserving framework for machine learning in industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6092–6102, Sep. 2020.

[24] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *Int. J. Med. Informat.*, vol. 112, pp. 59–67, Apr. 2018.

[25] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," 2019, *arXiv:1911.06270*. [Online]. Available: http://arxiv.org/abs/1911.06270

[26] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2019.

[27] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6479–6488.

[28] M. Osifeko, G. Hancke, and A. Abu-Mahfouz, *SurveilNet*. USA: IEEE Dataport, 2021, doi: 10.21227/4gnf-a488.

[29] D. G. Shreyas, S. Raksha, and B. G. Prasad, "Implementation of an anomalous human activity recognition system," *Social Netw. Comput. Sci.*, vol. 1, no. 3, p. 168, May 2020, doi: 10.1007/s42979-020-00169-0.

[30] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," 2015, *arXiv:1510.01553*. [Online]. Available: http://arxiv.org/abs/1510.01553

[31] A. Li, Z. Miao, Y. Cen, X.-P. Zhang, L. Zhang, and S. Chen, "Abnormal event detection in surveillance videos based on low-rank and compact coefficient dictionary learning," *Pattern Recognit.*, vol. 108, Mar. 2020, Art. no. 107355.

[32] J. Sun, J. Shao, and C. He, "Abnormal event detection for video surveillance using deep one-class learning," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3633–3647, 2019.

[33] S. Amraee, A. Vafaei, K. Jamshidi, and P. Adibi, "Abnormal event detection in crowded scenes using one-class SVM," *Signal Process.-Image*, vol. 12, no. 6, pp. 1115–1123, Mar. 2018.

[34] J. Gu, T. Su, Q. Wang, X. Du, and M. Guizani, "Multiple moving targets surveillance based on a cooperative network for multi-UAV," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 82–89, Apr. 2018.

[35] A. F. Santamaria, P. Raimondo, N. Palmieri, M. Tropea, and F. De Rango, "Cooperative video-surveillance framework in Internet of Things (IoT) domain," in *The Internet Things for Smart Urban Ecosystems*. USA: Springer, 2019, pp. 305–331.

[36] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014, *arXiv:1406.2199*. [Online]. Available: http://arxiv.org/abs/1406.2199

[37] W. Ullah, A. Ullah, I. U. Haq, K. Muhammad, M. Sajjad, and S. W. Baik, "CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks," *Multimedia Tools Appl.*, vol. 80, no. 11, pp. 16979–16995, May 2021.

[38] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[39] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symp. (APWCS)*, Aug. 2019, pp. 1–5.

[40] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2020.
[41] X. Lu, Y. Liao, P. Lio, and P. Hui, "Privacy-preserving asynchronous federated learning mechanism for edge network computing," *IEEE Access*, vol. 8, pp. 48970–48981, 2020.
[42] K. James *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
[43] V. Lomonaco, D. Maltoni, and L. Pellegrini, "Rehearsal-free continual learning over small non-IID batches," in *Proc. CVPR*, 2020, pp. 989–998.
[44] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[46] R. Grosse, "Lecture 15: Exploding and vanishing gradients," Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2017.
[47] T. Ilyas. *Model-Profiler*. Accessed: Jun. 8, 2021. [Online]. Available: https://pypi.org/project/model-profiler/
[48] T. Lawrence and L. Zhang, "IoTNet: An efficient and accurate convolutional neural network for IoT devices," *Sensors*, vol. 19, no. 24, p. 5541, Dec. 2019.
[49] J. Lin, W.-M. Chen, Y. Lin, J. Cohn, C. Gan, and S. Han, "MCUNet: Tiny deep learning on IoT devices," 2020, *arXiv:2007.10319*. [Online]. Available: http://arxiv.org/abs/2007.10319
[50] I. Freeman, L. Roese-Koerner, and A. Kummert, "Effnet: An efficient structure for convolutional neural networks," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 6–10.
[51] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1237–1246.
[52] K. Liu and H. Ma, "Exploring background-bias for anomaly detection in surveillance videos," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1490–1499.
[53] Y. Zhu and S. Newsam, "Motion-aware feature for improved video anomaly detection," 2019, *arXiv:1907.10211*. [Online]. Available: http://arxiv.org/abs/1907.10211

Martins O. Osifeko (Student Member, IEEE) received the B.Sc. (Hons.) degree in computer engineering from Olabisi Onabanjo University in 2009, the M.Sc. degree in electrical engineering from the University of Lagos in 2015, on communications engineering, and the master's degree in computer science from Ladoke Akintola University of Technology in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa. His research interests include sensor networks, wireless/mobile communications, artificial intelligence, and mathematical modeling.

Gerhard P. Hancke (Life Fellow, IEEE) received the B.Sc. and B.Eng. degrees from Stellenbosch University, South Africa, in 1970, the M.Eng. degree in electronic engineering from Stellenbosch University in 1973, and the D.Eng. degree from the University of Pretoria, South Africa, in 1983. He is a Professor with Nanjing University of Posts and Telecommunications, China, and the University of Pretoria. He is recognized internationally as a pioneer and a leading scholar in industrial wireless sensor networks research. He has initiated and co-edited the first Special Section on *Industrial Wireless Sensor Networks* in the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS in 2009 and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS in 2013. He has co-edited a textbook *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards* (2013), the first on the topic. Prof. Hancke has been serving as an Associate Editor and a Guest Editor for the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE ACCESS, and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS. He is currently the Co-Editor-in-Chief of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and a Senior Editor of IEEE ACCESS.

Adnan M. Abu-Mahfouz (Senior Member, IEEE) received the M.Eng. and Ph.D. degrees in computer engineering from the University of Pretoria. He is currently the Centre Manager of the Emerging Digital Technologies for 4IR (EDT4IR) Research Centre, Council for Scientific and Industrial Research (CSIR), an Extraordinary Professor at the University of Pretoria, a Professor Extraordinaire at Tshwane University of Technology, and a Visiting Professor at the University of Johannesburg. He has participated in the formulation of many large and multidisciplinary research and development successful proposals (as a principal investigator or main author/contributor). He is the founder of the smart networks collaboration initiative that aims to develop efficient and secure networks for the future smart systems, such as smart cities, smart grid, and smart water grid. His research interests include wireless sensor and actuator networks, low power wide area networks, software defined wireless sensor networks, cognitive radio, network security, network management, and sensor/actuator node development. He is a member of many IEEE Technical Communities. He is an Associate Editor of IEEE ACCESS, IEEE INTERNET OF THINGS, and IEEE TRANSACTION ON INDUSTRIAL INFORMATICS.