Computing and Informatics, Vol. 43, 2024, 633–666, doi: 10.31577/cai_2024_3_633

# TRIPLE DES WITH RADIAL BASIS FUNCTION NETWORKS FOR SECURING IN PROCESS OF CCTV FOOTAGE IMAGES

Prasanna Deshpande*

*Indian Institute of Information Technology*
*Nagpur, Maharashtra, India*
*e-mail:* deshpandeprasanna44@gmail.com

Ashwin G. Kothari

*Department of Electronics and Communication Engineering*
*Visvesvaraya National Institute of Technology*
*Nagpur, 440010, Maharashtra, India*

**Abstract.** The present smart city concept drives academics and urban planners to give residents modern, secure, and sustainable infrastructure as well as a fair level of living. CCTV footage has been employed to improve public safety and wellbeing to fill this demand recognising unusual occurrences. Despite scientific advances, it is challenging and time-consuming to monitor video systems inside of buildings. While cloud computing is unable to support a variety of dispersed IoT setups like wireless sensor networks, local services fall far short of meeting storage requirements. New strategies based on AI and deep learning are gradually replacing conventional computer vision techniques. Modern hardware and software are installed in large data centres, enabling them to process enormous amounts of data. To give cloudlets, fog computing, and multi-access edge computing with intelligent privacy protection for video data while it is being kept, we combine three separate edge computing techniques into a hierarchical edge computing architecture. Then, we offer a straightforward but secure solution. In this study, we look at the creation of a CCTV encryption method that combines RBFN with the Triple Data Encryption Standard (DES) to identify anomalies in intelligent video surveillance. The purpose of this study was to encourage the usage of an algorithm based

* Corresponding author

on RBFN-TDES that provides authenticated encryption rather than the original Data Encryption Standard (DES) method. Because the goal of this investigation was to figure out how the original DES algorithm was cracked, it was unavoidable that hackers would succeed. The suggested technique the security requirements for different encryption properties are also examined and analysed by RBFN-TDES. Finally, the comparison findings between the proposed system and existing systems are presented, along with an examination of each's security characteristics and performance metrics. Our proposed RBFN-TDES model is efficient enough to be used for securing CCTV pictures in a distributed computing environment. To evaluate the effectiveness of our approach, we run large experiments on benchmarks constructed on top of the EPFL dataset. Comparing our strategy to state-of-the-art techniques tested over the datasets, we find a 97.21 % increase in accuracy.

**Keywords:** Triple Data Encryption Standard (TDES), Internet of Things, artificial intelligence, edge computing, deep learning, Radial Basis Function Networks (RBFN)

## 1 INTRODUCTION

Video surveillance systems are common on city streets and roads, in businesses, neighbourhoods, bank branches, train stations, airports, and other public places, and they are becoming more and more important for public safety. Through the video surveillance system, suspicious people, places, and things can be found in time and closely watched, making it much less likely that criminal damage will happen. The police can learn about criminals from surveillance videos and find out where vehicles and people they think are involved are. The surveillance video can be used as objective court evidence during the interrogation phase of a case investigation. After criminal science and technology, action technology, and network investigation technology, video surveillance is now the fourth most important area of investigation technology. It is one of the most important parts of building a safe and smart city. In 2015, "several opinions on strengthening the networking application of public security video monitoring construction" were put forward by the Ministry of Public Security, the Ministry of Science and Technology, and nine other ministries and commissions. And they said that building a network for public security video surveillance helps keep the country safe and keep society stable. It also helps stop and punish violent terrorist crimes in the new situation. It is great significance to improve how cities and towns are run and to come up with new ways to run the social governance system.

Digital images are used in many ways, such as medical imaging, remote sensing, and private conferencing, because technology has improved. These pictures may have private or sensitive information on them [1]. When these images are sent over public networks, they could be changed or accessed without permission. Leaking sensitive information could cause problems with the military, national security, and

personal freedom. Also, when people want to share images over a public network, they need their privacy to be protected. So, images need to be protected from different security threats [2].

When large crowds gather, a camera system with early warning capabilities could detect potentially dangerous situations before they occur. Many crimes have been prevented by surveillance cameras, and they will continue to do so. Security cameras should always be installed in our personal areas. A surveillance camera system discourages vandalism and theft. It is extremely difficult to steal when cameras are always recording [3]. As a result, the thief is apprehended quite frequently. The thief will be recorded on video before or during the heist. Intruders who are caught on camera can be positively identified by police. The police can use surveillance cameras to both prevent criminal activity and quickly gather evidence to solve crimes. The installation of a CCTV system has the potential to reduce people's perceptions of the level of safety in public spaces while increasing their time spent there. CCTV camera installation will not only reduce crime rates but will also help with data collection, facility management, medical aid, and police investigations [4].

Modern centralised video processing systems store and process video data collected from the camera network [5]. Because the operator or central console only sees a few alerts or video clips, it is not necessary for them to pay close attention to how the surveillance system is working. Numerous research studies [6, 7] have proposed distributed video-surveillance systems based on this concept, which has recently gained popularity due to advancements in IoT and compute power at the edge nodes. As a result, the system's brains are dispersed across a large number of nodes, each of which may be equipped with a camera and some sort of processing system. These processing systems perform repetitive tasks and provide the results to the operator, relieving the latter of some of their responsibilities.

Because of the recent rapid advancement and widespread use of electronic technology, major urban areas now have far more modern technology. Many cities around the world are currently using ICTs to make their cities safer for their citizens and enhance the quality of the services they provide by leveraging the Internet of Things (IoT) [8, 9]. Mobile cameras mounted on manned or unmanned aerial and ground vehicles are frequently used for surveillance, in addition to stationary closed-circuit television (CCTV) cameras [10, 11]. Aeroplanes, satellites, unmanned aerial vehicles, human-driven ground patrol cars, and unmanned ground vehicles are all included in this broad category. First responders, government organizations, and private security service providers can gather a lot of audio-visual data about a lot of people without their knowledge or consent due to the pervasive and flexible use of CCTV cameras in public spaces such as streets, city corners, stores, and marketplaces [12, 13].

By 2021, there will most likely be more than a billion fixed security cameras in use worldwide, both in urban and suburban areas [14]. These closed-circuit television cameras serve as the system's "eyes and ears", and they frequently use a public network to transmit all of the data they generate and collect on a large number of people to remote video analytics and surveillance operation centers. These

nodes could be located in outlying areas. Because of the built-in vulnerabilities in the network design, the possibility of someone's right to privacy being infringed is increased, and this is one of the most essential concerns. The TCP/IP network architecture, which is widely used on today's Internet, is vulnerable to a multitude of threats since it was designed without proper security considerations. That is why the architecture was designed. Regardless, it is regarded as one of the most creative breakthroughs of the twentieth century [15]. Any monitoring system implementation that fails to address the aforementioned issues risks infringing on the privacy of users. Most modern virtual security systems are built on cloud or fog computing infrastructure. Video analytics can be performed in data centres with massive amounts of processing capacity located in the cloud, using cloud computing principles. Regardless, the network through which the raw video feeds are transmitted can be attacked and stolen. There are several scenarios in which this can occur. In general, using the fog computing or cloud computing paradigm raises the risk of privacy infringement.

Without the assistance of a human operator, modern intelligent video surveillance systems are capable of detecting and responding to potentially dangerous anomalies. Data is collected at this stage using sight sensors installed in the monitored area. The raw visual data must be pre-processed before features can be extracted [16]. The data gathered in this manner is fed into a modelling system that employs a learning technique to mimic the behaviours of prospective suspects while they are being observed and determine whether any of those behaviours are abnormal. A variety of machine learning techniques make use of cloud computing to analyse and store data for anomaly detection. Because network delays are unavoidable, cloud computing consumes a lot of bandwidth and slows response times [17, 18]. The nature of video surveillance, which is a time-sensitive application, necessitates reduced latency. Cloud computing and computation at the edge of the network are both parts of the best real-time intelligent video surveillance solution that is currently available [19].

The Triple Data Encryption Method (TDES) is a block cypher that employs symmetric keys and encrypts each data block three times with the DES cypher. Because of recent advances in cryptanalysis and computer capacity, the Data Encryption Standard's (DES) 56-bits key is no longer considered secure. In 2016, the DES and 3DES algorithms were discovered to have a severe security issue, which was made public via CVE-2016-2183. Because of the insufficient key size that these algorithms' CVE, DES, and 3DES display, NIST has deprecated CVE, DES, and 3DES for new applications as of 2017 and will do so for all applications by the end of 2023. It was replaced by the more reliable and secure AES algorithm. Although government and industry standards use the abbreviations TDES (Triple DES) and TDEA (Triple Data Encryption Algorithm), the majority of suppliers, customers, and cryptographers have used the word 3DES since RFC 1851 first introduced the notion. RBFN can yield universal approximations. RBFN is commonly used in the areas of regression, classification, pattern recognition, and time series forecasting [20, 21]. RBFNs have a small impact on the environment, can approximate any

continuous network, and can handle a lot of background noise. They are also better at approximating the world as a whole.

We look at how integrating the Triple Data Encryption Standard (DES) with radial basis function networks can result in a safe CCTV solution for intelligent video surveillance anomaly detection. In response to the original Data Encryption Standard (DES) algorithm being readily cracked by hackers, this study sought to advocate upgrading to an RBFN-TDES-based approach that permits authenticated encryption. In order for an encryption system to be secure, the suggested RBFN-TDES technique investigates and analyses the need for specific encryption features. Finally, we give the findings of a comparison between the proposed system and current systems, assessing each's security features and performance metrics. Our RBFN-TDES model is good enough to be used to protect CCTV images in a cloud-based network.

Images contain very sensitive and confidential information. Because images play an important role in many applications such as military communication, remote sensing, and medical imaging, it is critical to protect sensitive and proprietary information from unauthorised use and modification. Encryption is one of the greatest approaches for accomplishing this goal among information hiding methods. Many picture encryption techniques have been developed in recent years by researchers. To improve security, they employ several picture encryptions concepts. The main contributions of this paper are summarized as follows:

1. This paper created a revolutionary complete analysis framework for surveillance footage. It boosts the efficiency and accuracy of video analysis by combining object detection, keyframe selection, and super-resolution algorithms.
2. This research presented a Triple Data Encryption Standard (DES) with Radial Basis Function Networks (RBFN) for smart video surveillance anomaly detection to distinguish video objects in real time.
3. This paper suggested an RBFN-DES strategy that extensively incorporates the advantages of pixel space and feature space to improve the resolution of surveillance video identification objects.

The rest of the paper follows this format, and then we will look at some data. Section 2 examines the literature survey. Section 3 then delves deeper into the proposed architecture and present algorithms. Following that, the key outcomes of the tests are reported in Section 4 of the work, and Section 5 summarises the conclusion of the paper and potential future directions.

## 2 LITERATURE SURVEY

Barman et al. [22] proposed DNA-based ECC-based IoT encryption. The plain text characters are mapped to DNA genome sequences. The plain text is encrypted by selecting genome sequences at random from a large pool of publicly

available ones. Decryption requires the same DNA genome sequences as encryption. Thus, the sender and recipient must use the same sequences to encrypt and decrypt plain text. Encrypted delivery employs the same cypher text as plain text encryption.

A computer vision-based crowd catastrophe avoidance system explores crowd scene analysis [23]. These include the system's goal and the number of cameras. Other crowd analysis approaches, in addition to crowd disaster analysis, include calculations for forensic crowd analysis, people counting, crowd density estimation, person re-identification, and crowd evacuation. The benchmarked datasets are summarised below.

Yogameena and Nagananthini [24] introduced machine learning and ECC for IoT security. The authors screened the data gathering to optimise transmission and communicate just accurate data to minimise processing ($d$). $P$ is an elliptical curve point, and d is a secret random number. The plan's clean data transmission restriction increased IoT performance.

When IntegerDigits $[n, b]$ is less than 1, the encoding algorithms of Das and Giri [25] produce numerical values by accumulating weight $n$ with base $b$. The first technique encodes dynamic integer values in base $b$. The maximum number permitted is 65 536. IntegerDigits [192 bits, 65 bytes, and 536 bytes] = 11 are obtained using their method. There are more than 11 categories that are based on $b$ when it falls below 65 536. The authors did not provide a risk-free bit reduction approach for reducing ASCII table mapping. When $b$ is not dynamic, the combing group of the second algorithm is the number of $p$ digits in the variable IntegerDigits $[n, b]$. The authors' method yields $1.581 + 11 = 6$, and neither method yields a large number of categories. Computing will get increasingly challenging. The method was open to CPA because the system's encoding and mapping sections did not change plain text characters after applying ASCII table values.

The design of Arceda et al.'s three-phase system [25] includes face detection, normalisation, and violent situation recognition. Horn-Schunck and ViF are used to detect violent scenes. The optical flow algorithm is the technical term for this type of technology that examines area of the skin first, then examine each region separately.

Wang et al.'s [26] solution to the drawbacks of traditional, centrally managed data collection is to provide privacy protection to all users by utilising an algorithm-based method for achieving differential privacy. Wang and colleagues created a system to strike To strike a balance between user privacy, data integrity in Internet of Things devices, and computational cost, a distributed ledger and edge computing are combined in a proposed enhanced strategy with a balanced truth-finding approach. The design of the scheme made this possible.

Ajay and Rao [27] created a more effective and quick emotion recognition system, a Binary Neural Network (BNN) fed by Local Binary Pattern (LBP) output. To extract facial features for the BNN layer to successfully infer, LBP is set up as a pre-processing step. The pre-processing technique uses the Viola-Jones (VJ) algorithm to extract facial data from the image while obliterating extraneous background

elements. Facial Expression 2013 (FER-2013) data collection is used to train the LBP-BNN network. For the inference, the designed IP is synthesised as a bespoke hardware accelerator or overlay, and an FPGA is used to implement it.

Parate et al. [28] proposed employing CPU-only edge devices for anomaly detection in intelligent surveillance. The development of an object-level inference and tracking modular framework. We used feature encoding and trajectory association controlled by two complementary metrics to handle partial occlusions, posture deformations, and complicated sceneries. The components of an anomaly detection framework have been made to function as efficiently as possible on edge devices that just have CPUs (FPS).

Ullah et al. [29] looked at the problem of spotting unusual things in AIoT video surveillance settings. They have come up with a two-stream deep neural network for real-time analysis of surveillance data that can both spot oddities quickly and in detail. In the first step, surveillance video is fed into a fine-tuned CNN model to figure out if it is a normal or unusual event. This model is then used on IoT devices with limited resources.

Zhao et al. [30] proposed an intelligent edge surveillance techniques (INES) technique for a certain IIoT application that is based on deep learning. First, a depth wise separable convolutional strategy is used to build a lightweight deep neural network, which reduces the amount of work that needs to be done on it. Second, we use edge computing and cloud computing together to cut down on network traffic. Wan et al. [31] suggest using edge computing for video pre-processing to get rid of duplicate frames. This would let us move some or all the video processing tasks to the edge, reducing the need for computing, storage, and network bandwidth in the cloud centre and making video analyses more accurate. We present the magnitude of motion detection based on spatio-temporal interest points (STIP) and the multimodal linear features combination, which splits a video into super frame segments of interest to get rid of the redundancy in traffic video.

Chen et al. [32] propose a Distributed Intelligent Video Surveillance (DIVS) system that uses Deep Learning (DL) algorithms and deploy it in an edge computing environment. They build a multi-layer edge computing architecture and a distributed DL training model for the DIVS system. The DIVS system can move computing workloads from the network centre to the network edges to reduce the huge amount of network communication overhead and provide accurate and lowlatency video analysis solutions.

In foggy surveillance environments, smoke detection is crucial for disaster management in industrial systems. Existing methods fall short when applied to foggy videos due to clutter and unclear content challenges. Addressing this, the paper proposes an energy-efficient smoke detection method using edge intelligence and deep convolutional neural networks [33]. The method features a light-weight architecture, meeting accuracy, running time, and deployment feasibility requirements for industrial settings.

According to Hu et al. [34], MEC enables unique service scenarios that improve user experience, network performance, meet stringent latency requirements, and fos-

ter innovation. The researchers' own studies supported these findings the significant resource constraints that currently plague mobile devices. To do this, it is required to enable resource-intensive apps to utilise cloud computing without experiencing jitter, congestion, or outages. The increased time it takes for messages to arrive is a significant barrier. When working with the restricted resources of a mobile device, edge computing is preferable to the cloud, which is more remote and has fewer resources, because it is closer and has more of what is required.

Xie et al. [35] developed a novel, effective compressive data gathering method. This approach employs these strategies to avoid traffic analysis and flow tracing, to have untraceable message flows, and to have both secure message contents and Homomorphic encryption is the most effective for collecting compressive data. Gu et al. [36] developed a differential privacy mechanism to protect personal data during location data mining to safeguard very often accessed location data or user preferences of location through the distortion of accessing frequencies. The developed method for mining anonymous location data prioritizes privacy in location data mining. Emphasizing the major contributions of these studies is crucial to advancing the field of cloud computing-based privacy protection.

Hamdi et al. [37] deal with continuous deep one-class learning to detect anomalies in UAV video streams. Deep CNN networks can extract more features, which improves prediction accuracy. Encoders and decoders are employed in deep learning. Even though the exact methods for encoding and decoding vary from model to model, the main benefit of these processes is that they allow you to learn how typical input data is spread out and create a measure of anomaly. The comparative analysis of various methods and the datasets are discussed in Table 1.

## 3 PROPOSED SYSTEM

In this paper, we look at the development of a secure CCTV approach that uses the Triple Data Encryption Standard (DES) with Radial Basis Function Networks (RBFN) for smart video surveillance anomaly detection. The goal of this research was to propose that an RBFN-TDES-based approach that permits authenticated encryption (AE) replace the Data Encryption Standard (DES) algorithm that has failed. Figure 1 shows a block diagram of the RBFN-TDES.

### 3.1 Pre-Processing

In image processing, a feature is, in the simplest of terms, a meaningful piece of information for determining the computational work associated with a particular application. Features can be image-specific structures such as edges, points, objects, texture, etc. In complex situations, a single type of feature may not provide adequate information about the image data, necessitating the extraction of two or more features. Most colour object tracking systems utilise the HSI colour of the item and are resistant to variations in lighting. Using RGB features improves the

| Reference | Author | Method | Features | Learning | Anomaly criteria | Dataset | Year |
|---|---|---|---|---|---|---|---|
| [27] | Ajay and Rao | Handcrafted Features | Accuracy | FPGA system architecture using a binary neural network | Emotions on the face can be identified | JAFFE | 2021 |
| [28] | Parate et al. | CNN | Real-time applications | Connection of trajectory and extraction of spatial-temporal features | Identifying pedestrians | UCSD Ped1 UMN | 2021 |
| [29] | Ullah et al. | GAN | Efficient resource administration | L-CNN and BD-LSTM in videos | Crime scene | Crime scene | 2021 |
| [30] | Zhao et al. | $DNN$ | Reduce network occupancy and system response delay | Intelligent Edge Surveillance | Cloud, Deep Learning, Edge | Self deployed | 2020 |
| [31] | Wan et al. | CNN | Reduced bandwidth, security, real-time | Motion magnitude for Intelligent Edge Surveillance videos | Vehicle anomaly | UA-DETRAC, crossroad in Beijing city | 2022 |
| [32] | Chen et al. | DNN | Balances computational power, workload | DIVS | Vehicle classification | Self deployed | 2019 |
| [33] | Muhammad et al. | CNN | Processing requires less memory | LCNN | Detecting smoke during hazy surveillance | ImageNet | 2019 |

Table 1.

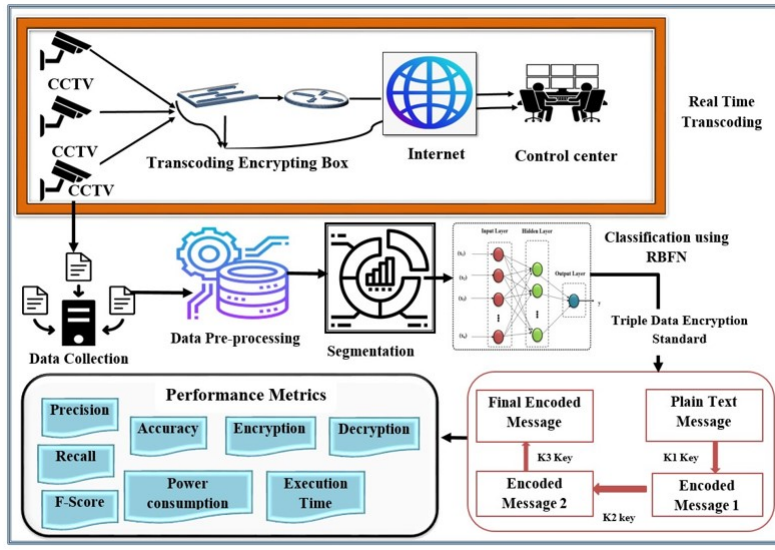


Figure 1. Proposed method of RBFN-TDES

classification accuracy of colour feature vectors, although these features are sensitive to variations in lighting. Scale Invariant Feature Transform (SIFT) is employed to retrieve local feature descriptors. SIFT is a robust local invariant feature descriptor that was primarily created for grayscale images. Edge features are less responsive to changes in illumination than colour descriptors. Texture features can also be utilised, although at the cost of additional processing time and procedures. There are other additional feature descriptors, such as biological properties, optical flow, etc. Features are employed to distinguish between foreground and background objects. Preparing a video for further processing is known as "pre-processing", and it is the first step in the process. The video will need to go through pre-processing before proceeding to the next step of processing. Pre-processing can help reduce mistakes and noise that are introduced into photos because of reading and scanning procedures, resulting in higher-quality photographs. Color correction, statistical analysis, and the convolution method are all pre-processing procedures. The video data would be ready for the next step, which would include dealing with challenging video sequence processing challenges. This would be accomplished by video data pre-processing.

### 3.2 Image Segmentation

Object detection is a computer vision system that identifies moving objects such as humans, vehicles, animals, and birds. Detecting objects is one of the first steps in

object tracking. Face detection and pedestrian detection represent the state of the art in object detection. Object recognition in computer vision involves identifying the target host. Combining the characteristics of an object with a model of the object yields recognition. Several factors, including scene consistency, the quantity of items in an image, and the likelihood of occlusion, influence the difficulty of object detection. Segmentation is a computer vision technique that divides an image into groups. These groups can all have the same colour pixels or border edges, as well as a common shape such as a line, circle, ellipse, or polygon. A picture is segmented when it is divided into groups based on traits that they share. Image segmentation can be further subdivided into subcategories, such as thresholding, edge detection, and region-based categorization. As a result of the segmentation procedure, a collection of tagged pixels is produced.

### 3.3 Image Enhancement

In the science of computer vision, "image augmentation" refers to approaches used to improve the quality of an image. Using a variety of image enhancement techniques, images can be changed to better suit the tastes of the viewer.

### 3.4 Shape-Based Classification

Classification based on shape applies exclusively to the geometry of an object, not its structural examination. Objects can be categorised based on the extracted regions'geometry, such as boxes, blobs, etc., that contain motion. Measures the accuracy and performance of the research of various shape characteristics. Zhao and Nevatia presented a method for tracking humans in a crowded scene with occlusion using human form models and camera models. A Bayesian framework and expansion of the mean-shift tracking with the shape model provide a principled technique to concurrently detect and track persons [38].

### 3.5 Motion-Based Classification

Classification based on motion helps reduce the dependency on the spatial primitives of the objects and provides a strong classification approach. It does not require established pattern templates but has difficulty identifying a stationary object. Even though motion-based classification has a modest level of accuracy, it is a computationally intensive method of classification. Johnsen and Tews suggested a vision-based tracking and classification system for objects. It was capable of handling occlusions and performed admirably across a variety of objects and weather situations [39].

### 3.6 Radial Basis Function Networks (RBFN)

Most modern techniques are typically very good in static situations. It is nevertheless difficult to detect motion during shifting conditions, such as those found in landscapes. Due to the shifting intensities of the background and foreground pixels and the difference between movable objects, the dynamic background is tedious. This makes it difficult to tell them apart.

In this section, we will go through our cutting-edge, RBF artificial neural network-based motion detection algorithm. This method separates the foreground from the background activity and static background, preventing dynamic backgrounds from being confused with moving objects in the scene. Figure 2 depicts an RBF neural network with In one such layered structure, there are input, hidden, and output layers. Even though it has some other benefits, its ability to approximate and its speed of learning are two of its best features.

Figure 3 represents the fact that our method employs not one but two main modules: one that generates various backdrops and another that scans the scene for moving objects. The multibackground generation (MBG) module employs the Euclidean distance to generate a probabilistic background model that the user can modify that exists between each input pixel and the candidates for the reference backgrounds that correspond to it. The hidden layer's neural hubs are responsible for communicating this data to the network. The addition of a hidden layer based on the probabilistic background model has improved the RBF network design. This model can represent the entire tonal range that each background pixel possesses.

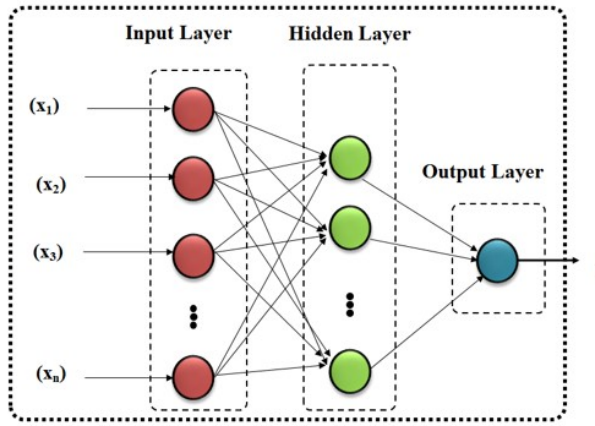


Figure 2. Radial basis function neural network

When the MBG component is deployed, the proposed MOD component for detecting moving objects is activated. This element must keep track of moving

targets. The MOD module uses a block alarm method to extract objects from the system to detect moving objects thoroughly and accurately. To accomplish this, two strategies can be used. After the block alarm technique is done, which means that the dynamic and static backdrop regions do not need to be explored any more, the object extraction process is done.

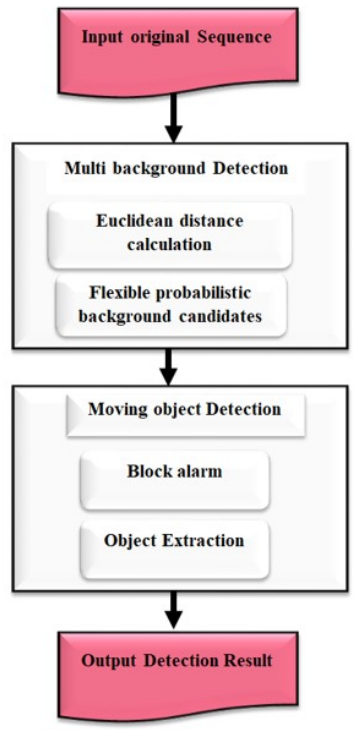


Figure 3. Overview of the modules involved in the proposed method

### 3.6.1 Multi-Background Generation

In this stage, an HSV colour space similar to what the human visual system can handle must be created for the input layer. The symbols will indicate colour data (hue, saturation, and value) for every pixel in every incoming frame. It is possible for the structure of a network to become overly complex, and performance can suffer as a result of having too many neurons. Because of this, it is very important to have a flexible probabilistic background model that can represent neurons that are not visible.

Using each input pixel intensity value, a flexible probabilistic backdrop model must be built.

To determine whether an input pixel is near background intensity candidates, we use the Euclidean distance of vectors in the HSV colour hexcone. This is a hexadecimal distance. All that is required is a vector comparison. This conclusion is reached by measuring the distance between pixels $q_i = (h_i, s_i, v_i)$ to pixel $q_j = (h_j, s_j, v_j)$ by

$$d(q_i, q_j) = \|(v_i s_i \cos(h_j),\ v_i s_i \sin(h_j), \sin(h_i)) - (v_j s_j \cos(h_j),\ v_j s_j \sin(h_j))\|_2^2 . \quad (1)$$

Using this metric allows one to avoid issues with hue periodicity as well as hue instability for low saturation values.

An empirical tolerance is used to determine whether an incoming pixel $q_t(x, y)$ is one of the background candidates $C(x, y)_k$, for some value $k \in \{1, n\}$. One possible statement of this decision-making criterion is

$$q_t(x, y) = \begin{cases} \in C(x, y)_k, & \text{if } d(q_t(x, y)), C(x, y)_k \leq \epsilon, \\ \notin C(x, y)_k, & \text{otherwise.} \end{cases} \quad (2)$$

Background candidates near the entering pixel $q_t(x, y)$ are represented by

$$C(x, y)'_k = (1 - \beta)C(x, y)_k + \beta p_t(x, y), \quad (3)$$

where $C(x, y)_k, C(x, y)'_k$ represent the initial and updated $k^{\text{th}}$ candidates at position $k(x, y)$, and $\beta$ is a predefined parameter. The method used to create the background in this case can be compared to an unsupervised learning process of the centres' location in the RBF network.

### 3.6.2 Moving Object Detection

1. Three inputs, three outputs, and M hidden neurons are present in this RBF network. The MBG module determines the number $M$ of neurons and their centres $(C1, \ldots, CM)$ in the hidden layer of the RBF network seen in Figure 2. The $q_t(x, y)$ entering pixel's HSV components serve as the input vector once structure has been established. H, S, and V stand in for them. After determining the basics, the function generates the output of each hidden neuron by calculating Euclidean distances between the input vector and its centre coordinates.

$$z_i(q) = \phi\left(\|p - B_i\|\right), \quad \text{where } i = 1, 2, \ldots, M. \quad (4)$$

The basics function is $C_i$, which stands for the $i^{\text{th}}$ neuron's center, the input vector is $p$, and the hidden neurons' number is $M \|q - C_i\|$ is an Euclidean distance between $p$ and $C_i$. Many other basic function types are frequently utilised in a variety of settings. In our method, we use the Gaussian function, which is by far the most popular type of basis function. The representative function is

defined as follows:

$$\phi\left(\|Q - C_i\|\right) = \operatorname{exq}\left(\frac{-\|p - E_i\|^2}{2\sigma^2}\right), \tag{5}$$

what is the empirical tolerance of the Euclidean distance, as well as the definition of this tolerance. This is because the probabilistic model, which has been updated to include additional background candidates, is correlated with a lower value. Lower standard division values may result in a smoother Gaussian curve. Long-term, this can prevent the output layer's summation from becoming too high, resulting in incorrect assessments of the dynamic background. As a result, it has a proportional relationship based on our findings, the symbol experimentally shown.

The Gaussian function is a good choice because it can be factorised and is also localised for the purposes of our application. As the output value rises, the probability that an incoming pixel will be in the foreground also rises. The entering frame is split into $w$ and $w$ blocks so that the moving and still backgrounds do not get looked at too much.

$$\delta = \sum_{p\in\mu}\sum_{i=1}\phi\left(\|q - B_i\|\right), \tag{6}$$

where $q$ is the number of different pixels in the relevant block, $M$ represents the number of hidden neurons, and resents the number of visible neurons. Setting $w$ to 4 equals four different pixels, which is the size of the block. When the estimated sum of block $I$ and block $J$ exceeds a predefined threshold, block $A(i, j)$ is marked with a 0 to indicate that no pixels from moving objects are present. This occurs when the total of blocks $I$ and $J$ surpasses the threshold. If not, the label for block $A(i, j)$ will be 1, which means that it has pixels that show objects that are moving.

$$A(i,j) = \begin{cases} 0, & \text{if } \delta \geq S, \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

By setting $S$ to 12, also known as the magic number, one can find blocks that may contain movable objects. To summarize, candidates who are in the shadows receive an update in the covert layer from

$$C(x,y)_k^t = \begin{cases} C(x,y)_k^{t-1}, & \text{if } p_t(x,y) \notin C(x,y)_k^{t-1}, \\ \alpha p_i(x,y) + (1-\alpha)C(x,y)_k^{t-1}, & \text{otherwise,} \end{cases} \tag{8}$$

where $C$ is a predetermined parameter and $C(x,y)_k^{t-1}, C(x,y)$ are the $k^{\text{th}}$ candidates at position $(x, y)$ of the previous and current flexible background models,

respectively. The decision rule is established in order to find whether or not $q_t(x, y)$ is a member of $C(x, y)_k^{t-1}$.

2. Object Extraction Procedure.

   Following the completion of the block alarm procedure, only blocks containing moving objects are processed by the object extraction technique. This ensures that no unnecessary tests are performed. In the final step of our method, the layer produced by running an RBF network is used to calculate the mask for binary motion detection. Our strategy will be complete once these steps are completed. The output layer figures out a function by putting the data from the hidden layer through a weighted linear combination.

$$F = \sum_{i=1}^{M} \omega_i \left( z_i(q) \right) + \omega_0, \tag{9}$$

   $\omega_0$ is a fixed threshold, and $z_i$ is the value produced by the $i^{\text{th}}$ hidden neuron. Once $\omega_i$ has been initialized, the value 1 is used for testing. The binary motion detection mask can then be made by doing the following:

$$D(x, y) = \begin{cases} 1, & \text{if } f(x, y) < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

   The pixel in question is a component of a moving object if you label $D(x, y)$ with a 1, which is labelled 0, and is considered to be part of the background. We will modify the weights for the processing of the incoming frame that comes after the one we are currently processing once the procedures for that frame are complete. Following the initial setting of each weight to 1, the weights are then modified as follows:

$$\omega_i^{t+1} = \left( \omega_i^t + \eta, z_i \right) \cdot \frac{M}{M + \eta \cdot \sum_{i=1}^{M} z_i}. \tag{11}$$

   Both the number of hidden neurons and the rate at which they receive new information are represented by the weight at frame $M$ between the $i^{\text{th}}$ buried neuron and the last layer. The connections between the output layer as well as the hidden neurons close to the input vector are strengthened after the weights are changed, whereas weak connections exist between the hidden neurons and the output layer.

### 3.7 Triple-DES

Its key length is far longer than that of the vast majority of other encryption techniques, which is advantageous. The Advanced Encryption Standard (AES) replaced the Data Encryption Standard (DES). As a result, DES is currently regarded as obsolete. We do this by using a single DES algorithm three times, along with three

subkeys and key padding as needed. A key must now have a minimum length of 64 bits. The triple DES inclusion is easily applied because of its adaptability and compatibility. Shown in Figure 4.

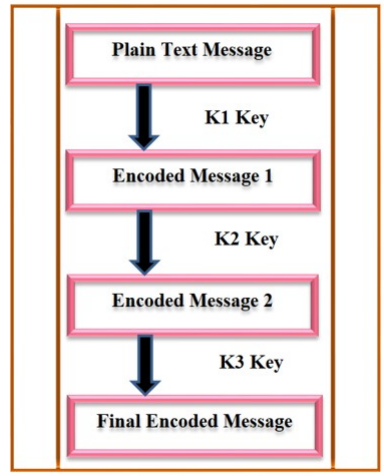


Figure 4. Block diagram for TDES

Triple DES encryption can take several forms, some of which are well-known:

- DES-EEE3, a variant of TDES encryption that employs three unique keys;
- 3TDES's three operations: encryption, decryption, and encryption use three distinct keys, similar to DES-EDE3.
- Decryption of DES-EEE2 and DES-EDE2 requires a special key.

Assume EK (I) is the DES encryption key used to encrypt I, and DK is the DES decryption key used to decrypt I. Decryption is abbreviated as DK, whereas encryption is abbreviated as EK. TDEA's encryption and decryption processes can be thought of as a compound operation that combines the encryption and decryption algorithms used by DES. The steps are carried out in the order that they are listed.

The following is a description of the process used to encrypt TDEA data: The following procedures must be followed to convert a 64-bit block I to a 64-bit block O:

$$O = E_{K3}\left(D_{K2}\left(E_{K1}(I)\right)\right). \tag{12}$$

TDEA decryption operation: a block I of 64-bit is transformed to a block $O$ of 64-bit which is characterized as follows:

$$O = D_{K1}\left(E_{K2}\left(D_{K3}(I)\right)\right). \tag{13}$$

**Algorithm 1** Three Keying Options

The standard outlines the various keying options for bundles that adhere to it.

a. Independent Keys $K1$, $K2$, and $K3$ of the First Keying Option.
b. Second Keying Option independent keys $K1$, $K2$, and key $K3$.
c. The Third Keying Options are $K1, K2$, and $K3$.

**Begin**

1. Option 1, the preferred option employs three mutually independent keys ($K1 \neq K2 \neq K3 \neq K1$). It gives key space of $3 \times 56 = 168$ bit.
2. Option 2 employs two mutually independent keys and third key that is the same as the first key ($K1 \neq K2$ and $K3 = K1$). This gives key space of $2 \times 56 = 112$ bit.
3. Option 3 is a key bundle of three identical keys ($K1 = K2 = K3$). This option is equivalent to DES Algorithm.

**End**

When the time required to decrypt data using the TDEA mode of operation equals the time required to decrypt data using the single DES mode, we say that the two modes are backwardly consistent with one another.

1. Any plaintext that has been encrypted and calculated using the single DES mode of operation can be decoded using the TDEA mode of operation.
2. Because the TDEA mode of operation is equal to a single DES mode of operation, any one can be used to correctly decrypt plaintext encrypted by TDEA.

When Keying Option 3 is turned on, all four modes are available, and they all work with the same single DES modes of operation (ECB, CBC, CFB, and OFB) in the same way.

## 4 RESULT AND DISCUSSION

### 4.1 Experimental Setup

This system evaluation was performed using the Ecole Polytechnique F'ederale de Lausanne (EPFL) in Switzerland EPFL dataset. The dataset's characteristics that point to a genuine setting include repetition, dense crowds of people, and changes in lighting. The EPFL dataset was recorded in the EPFL Rolex Learning Center using three static HD cameras. Unlike most of the existing multi-camera datasets, the cameras' fields of view are overlapping. Each camera has a resolution of $1\,920 \times 1\,080$ pixels and during the acquisition a frame rate of 60 frames per second was used. The first scene in this data set is set in a laboratory, in a large room with four people

who overlap and move in front of and behind the camera, shifting their positions in the image. The office setting is the second scene in this data set, it is a smaller space with two people who frequently cross paths and move around the frame. In the second case, eight students are crammed into a small hallway at a university. The camera is positioned at various distances from the subjects, and the lighting varies from clip to clip. Many of the images overlap.

### 4.2 Performance Metric

A variety of performance measures were used to evaluate classifier performance. The most basic and extensively used criterion for classifier evaluation is accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{14}$$

where TP is the number of samples correctly classified as positive, TN is the number of samples correctly classified as negative, FN and FP represent the amount of samples that were incorrectly labelled as positive or negative. Precision is defined as the proportion of positive samples to all samples classified as positive.

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{15}$$

The percentage of correctly labelled positively recognised samples in comparison to all positively identified samples is known as "recall".

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{16}$$

The F1-measure shows how important it is to find a good balance between recall and precision.

It is common practise to use the F1-score as a classification evaluation measure since it provides a value that finds a balance between accuracy and recall in a single value. The F1-score is calculated by weighting recall and precision and averaging them.

$$\text{F1-Score} = 2^* \frac{precision * recall}{precision + recall}. \tag{17}$$

The receiver operating characteristic curve is created by drawing a line from the percentage of correct diagnoses to the total number of false positives. The statistical term "area under the curve" refers to a specific type of data analysis, which ranges from zero to one and frequently surpasses 0.5.

**Encryption time.** Because the strength of an encryption algorithm has a negative association with the amount of time required for the encoding process, a method is considered more successful when it takes less time to complete the encoding process. This criterion can be evaluated in two ways:

Calculate the encryption time based on the size of the input (100, 200, 300, 400, and 500 kB).

After analysing the encryption time based on the input by variable-count characters, we compute the time it takes to encrypt and decode the data.

Deciphering a process requires less time than decrypting a process, as proven. The results demonstrated this, indicating that the length of time required to encrypt a file may increase linearly as the number of characters or file size increases. This shows that the proposed work is much easier to do in terms of computation than was thought before.

The suggested technology encrypts plain text quicker than previous strategies such as single-shot multibox detection (MobileNet-SSD), you only look once (YOLO-v3), CNN, DNN, and LSTM.

**Decryption time.** This demonstrates how long it takes to decrypt and reassemble the data using the cypher text as input. The amount of time required before success is referred to as the "temporal complexity" of the decrypting algorithm. The following formula can be used to calculate how long it takes to decrypt data.

$$Time\ consumed = end\ time - start\ time. \tag{18}$$

The suggested approach requires significantly less time to decode than previous encryption methods. So, the model shown can be used and is helpful for making sure that communication is secure.

**Power consumption.** This section investigates how much energy the embedded system consumes. The system can monitor its own power consumption and achieve its goal using an external power source. A device called the UpSquared2 that draws electricity from a 5-volt input voltage has been used to study how current consumption changes over time. Another study looked at how the VPU affected energy consumption. As a result, the node's consumption has been tracked in a variety of scenarios, including when it is idle, when the algorithm is running exclusively on the CPU, and when both the CPU and the VPU are in use.

### 4.2.1 Precision

Table 2 and Figure 5 show the results of a comparison of precision of the RBFN-TDES methodology to that of other existing approaches. The graph shows how the machine learning strategy resulted in increased precision. For example, with node 100, the precision value is 90.37 % for RBFN-TDES, whereas the MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM models have obtained precision of 72.67 %, 77.45 %, 81.45 %, 84.91 %, and 87.13 %, respectively. The maximum performance of the RBFN-TDES model was successfully demonstrated using a diverse set of nodes. Similarly, under the 600 nodes, the precision value of RBFN-TDES is 92.77 %, while

| No. of Nodes | MobileNet-SSD | YOLO-v3 | CNN | DNN | LSTM | RBFN-TDES |
|---|---|---|---|---|---|---|
| 100 | 72.67 | 77.45 | 81.45 | 84.91 | 87.13 | 90.37 |
| 200 | 73.14 | 78.18 | 81.53 | 84.12 | 87.34 | 91.22 |
| 300 | 73.23 | 77.34 | 80.89 | 84.22 | 88.12 | 92.98 |
| 400 | 74.18 | 80.33 | 82.98 | 85.66 | 89.34 | 91.45 |
| 500 | 75.23 | 80.56 | 82.18 | 86.19 | 88.22 | 93.88 |
| 600 | 76.56 | 81.13 | 83.28 | 85.19 | 89.65 | 92.77 |

Table 2. Precision analysis for RBFN-TDES method with existing systems

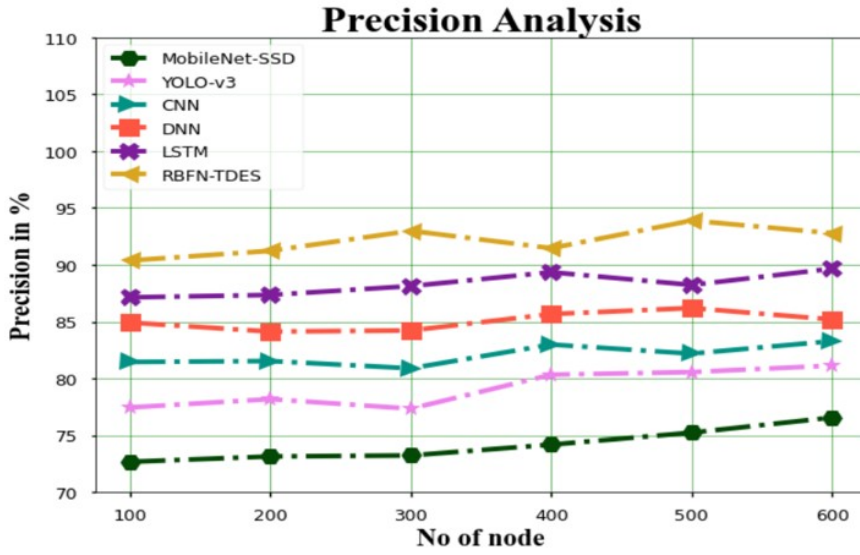


Figure 5. Precision analysis for RBFN-TDES method and existing systems

it is 76.56 %, 81.13 %, 83.28 %, 85.19 %, and 89.65 % for MobileNet-SSD, YOLO-v3, CNN, DNN and LSTM models, respectively.

### 4.2.2 Recall

The results of comparing the RBFN-TDES methodology to other existing methodologies with recall are shown in Figure 6 and Table 3. The graphic shows how implementing the machine-learning strategy enhanced recall performance. For example, with node 100, the recall value is 88.12 % for RBFN-TDES, whereas the MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM models have obtained recalls of 61.78 %, 66.12 %, 72.76 %, 78.45 %, and 84.55 %. The maximum performance of the RBFN-TDES model was demonstrated successfully with a diverse set of nodes.

| No. of Nodes | MobileNet-SSD | YOLO-v3 | CNN | DNN | LSTM | RBFN-TDES |
|---|---|---|---|---|---|---|
| 100 | 61.78 | 66.12 | 72.76 | 78.45 | 84.55 | 88.12 |
| 200 | 62.98 | 65.46 | 71.87 | 78.34 | 85.12 | 90.13 |
| 300 | 62.45 | 66.34 | 74.12 | 80.12 | 84.78 | 91.45 |
| 400 | 63.19 | 69.22 | 75.66 | 81.33 | 85.77 | 91.43 |
| 500 | 64.98 | 70.21 | 76.32 | 82.67 | 86.12 | 92.34 |
| 600 | 65.92 | 71.43 | 77.18 | 83.56 | 87.33 | 93.19 |

Table 3. Recall analysis for RBFN-TDES method and existing systems

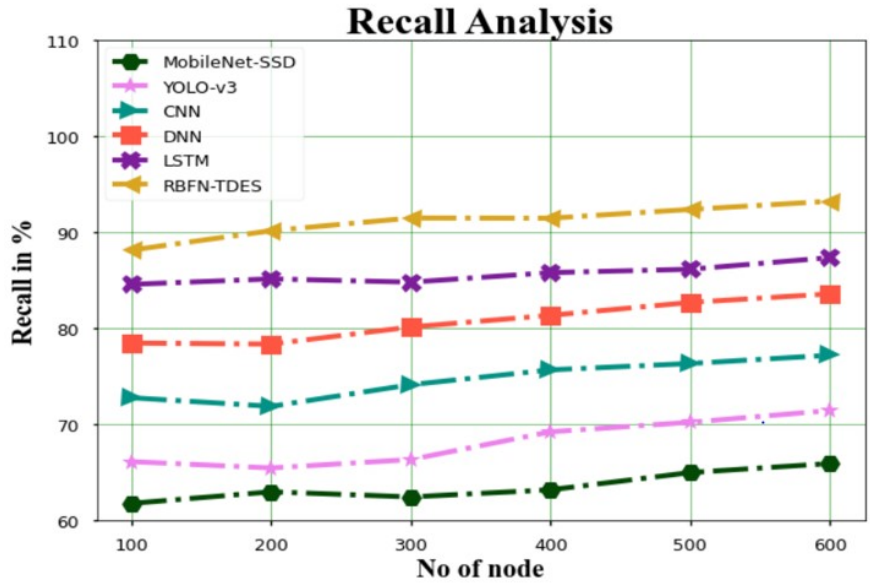


Figure 6. Recall analysis for RBFN-TDES method and existing systems

Similarly, under 600 nodes, the recall value of RBFN-TDES is 93.19 %, while it is 65.92 %, 71.43 %, 77.18 %, 83.56 %, and 87.33 % for MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM models, respectively.

### 4.2.3 F-Score

Figure 7 and Table 4 show a comparison of the RBFN-TDES methodology to other existing methodologies using F-score analysis. The data in the figure shows that using machine learning resulted in superior performance, as measured by the F-score. For example, with node 100, the F-score value is 92.67 % for RBFN-TDES, whereas the MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM models have obtained F-scores of 81.89 %, 85.34 %, 83.34 %, 87.12 %, and 89.66 %. The maximum

| No. of Nodes | MobileNet-SSD | YOLO-v3 | CNN | DNN | LSTM | RBFN-TDES |
|---|---|---|---|---|---|---|
| 100 | 81.89 | 85.34 | 83.34 | 87.12 | 89.66 | 92.67 |
| 200 | 81.43 | 85.67 | 83.89 | 87.56 | 89.34 | 93.87 |
| 300 | 80.34 | 85.12 | 83.45 | 87.34 | 90.45 | 93.15 |
| 400 | 81.66 | 85.45 | 83.12 | 88.11 | 91.12 | 94.66 |
| 500 | 80.97 | 86.89 | 82.45 | 89.45 | 91.56 | 95.78 |
| 600 | 81.67 | 86.12 | 83.13 | 89.56 | 91.33 | 96.12 |

Table 4. F-Score analysis for RBFN-TDES method and existing systems

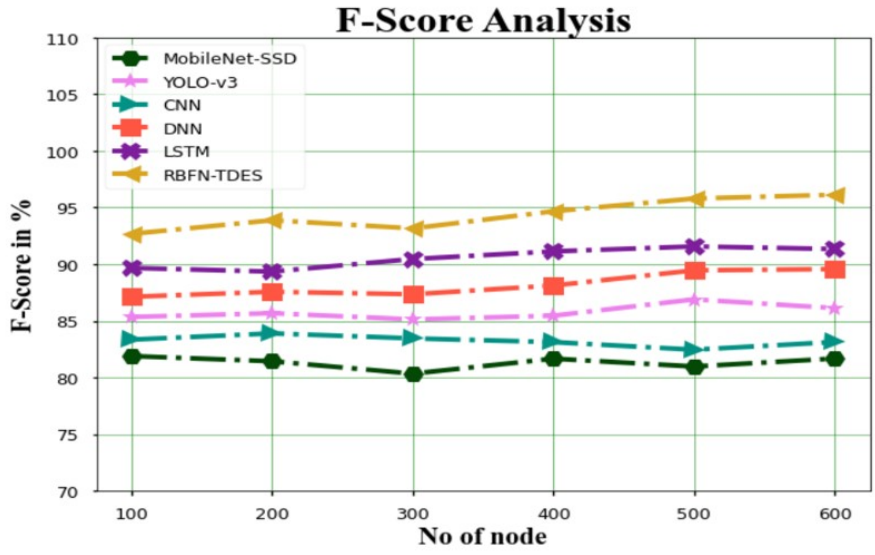


Figure 7. F-Score analysis for RBFN-TDES method and existing systems

performance of the RBFN-TDES model was demonstrated successfully using a diverse set of nodes. Similarly, under 600 nodes, the F-score value of RBFN-TDES is 96.12 %, while it is 81.67 %, 86.12 %, 83.13 %, 89.56 %, and 91.33 % for MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM models, respectively.

### 4.2.4 Encryption Time

Table 5 and Figure 8 show an analysis of how long it takes to encrypt data using the RBFN-TDES method in comparison to other methods. The nodes clearly show that the RBFN-TDES method has outperformed the other techniques in all aspects. For example, with 100 nodes, the RBFN-TDES method has taken only 2.678 sec to encrypt, while the other existing techniques like MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM have an encryption time of 13.521 sec, 11.871 sec, 8.567 sec,

| No. of Nodes | MobileNet-SSD | YOLO-v3 | CNN | DNN | LSTM | RBFN-TDES |
|---|---|---|---|---|---|---|
| 100 | 13.521 | 11.871 | 8.567 | 7.457 | 5.123 | 2.678 |
| 200 | 13.521 | 11.432 | 9.345 | 7.234 | 5.786 | 2.543 |
| 300 | 14.987 | 11.321 | 9.765 | 7.876 | 5.234 | 3.896 |
| 400 | 14.123 | 12.987 | 9.327 | 7.669 | 6.126 | 3.112 |
| 500 | 15.987 | 12.567 | 9.984 | 8.154 | 6.345 | 3.987 |
| 600 | 15.478 | 12.532 | 10.378 | 8.987 | 6.987 | 4.654 |

Table 5. Encryption time analysis for RBFN-TDES method and existing systems

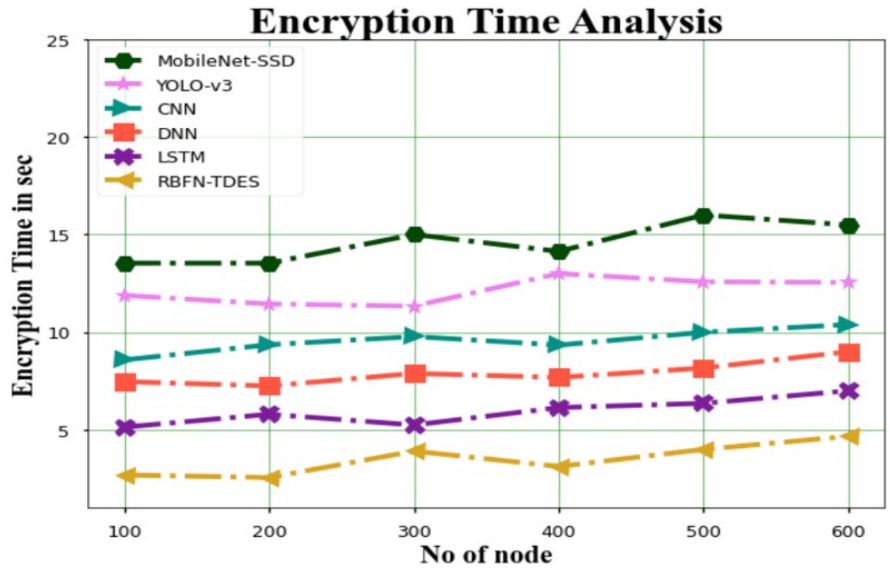


Figure 8. Encryption time analysis for RBFN-TDES method and existing systems

7.457 sec, and 5.123 sec, respectively. Similarly, for 600 nodes, the RBFN-TDES method has an encryption time of 4.654 sec, while the other existing techniques like MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM have 15.478 sec, 12.532 sec, 10.378 sec, 8.987 sec, and 6.987 sec of encryption time, respectively.

### 4.2.5 Decryption Time

Table 6 and Figure 9 show a comparison of the RBFN-TDES method's decryption time with that of other currently used methods. The nodes clearly demonstrate that the RBFN-TDES method has outperformed the other techniques in all aspects. For example, with 100 nodes, the RBFN-TDES method has taken only 3.567 sec to decrypt, while the other existing techniques like MobileNet-SSD, YOLO-v3, CNN,

| No. of Nodes | MobileNet-SSD | YOLO-v3 | CNN | DNN | LSTM | RBFN-TDES |
|---|---|---|---|---|---|---|
| 100 | 12.542 | 9.876 | 8.567 | 6.897 | 5.213 | 3.567 |
| 200 | 12.678 | 9.459 | 8.321 | 7.432 | 5.345 | 3.674 |
| 300 | 13.876 | 10.213 | 8.689 | 6.167 | 4.456 | 3.321 |
| 400 | 14.786 | 10.987 | 8.765 | 6.987 | 5.245 | 3.987 |
| 500 | 14.672 | 11.785 | 9.456 | 7.457 | 5.987 | 3.187 |
| 600 | 16.553 | 11.932 | 9.987 | 7.326 | 6.145 | 4.321 |

Table 6. Decryption time analysis for RBFN-TDES method and existing systems

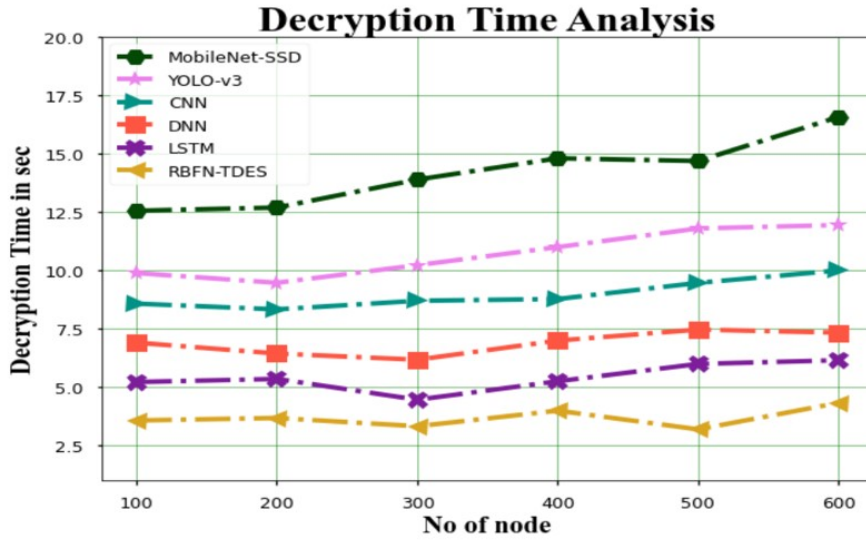


Figure 9. Decryption time analysis for RBFN-TDES method and existing systems

DNN, and LSTM have a decryption time of 12.542 sec, 9.876 sec, 8.567 sec, 6.897 sec, and 5.213 sec, respectively. Similarly, for 600 nodes, the RBFN-TDES method has a decryption time of 4.321 sec, while the other existing techniques like MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM have 16.553 sec, 11.932 sec, 9.987 sec, 7.326 sec, and 6.145 sec of decryption time, respectively.

### 4.2.6 Execution Time

Table 7 and Figure 10 show a comparison of the execution time of the RBFN-TDES technique with existing methods. The data clearly shows that the RBFN-TDES method has outperformed the other techniques in all aspects. For example, with 100 nodes, the RBFN-TDES method has taken only 2.543 sec to execute, while the other existing techniques like MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM

| No. of Nodes | MobileNet-SSD | YOLO-v3 | CNN | DNN | LSTM | RBFN-TDES |
|---|---|---|---|---|---|---|
| 100 | 9.123 | 8.456 | 7.345 | 5.987 | 4.542 | 2.543 |
| 200 | 9.456 | 8.564 | 7.567 | 5.235 | 4.198 | 2.654 |
| 300 | 9.732 | 8.125 | 7.134 | 5.675 | 4.678 | 2.987 |
| 400 | 9.321 | 8.478 | 6.987 | 5.876 | 4.321 | 3.132 |
| 500 | 9.416 | 8.652 | 7.456 | 6.124 | 4.987 | 3.987 |
| 600 | 9.441 | 8.473 | 7.786 | 6.543 | 5.234 | 3.176 |

Table 7. Execution time analysis for RBFN-TDES method and existing systems

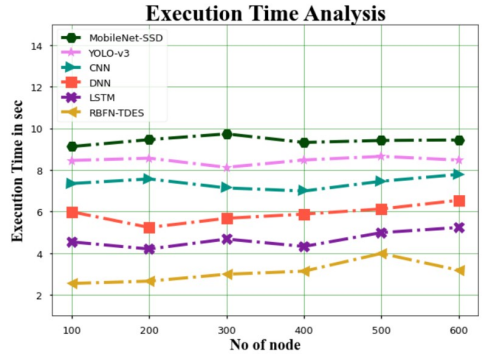


Figure 10. Execution time analysis for RBFN-TDES method and existing systems

have an execution time of 9.123 sec, 8.456 sec, 7.345 sec, 5.987 sec, and 4.542 sec, respectively. Similarly, for 600 nodes, the RBFN-TDES method has an execution time of 3.176 sec, while the other existing techniques like MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM have 9.441 sec, 8.473 sec, 7.786 sec, 6.543 sec, and 5.234 sec of execution time, respectively.

### 4.2.7 Power Consumption

| No. of Nodes | MobileNet-SSD | YOLO-v3 | CNN | DNN | LSTM | RBFN-TDES |
|---|---|---|---|---|---|---|
| 100 | 49.34 | 51.66 | 43.98 | 37.12 | 31.86 | 25.19 |
| 200 | 48.12 | 52.87 | 44.87 | 38.56 | 32.87 | 26.77 |
| 300 | 47.87 | 53.98 | 45.23 | 39.12 | 33.12 | 27.23 |
| 400 | 49.23 | 54.21 | 46.98 | 40.56 | 34.87 | 28.54 |
| 500 | 49.32 | 55.98 | 47.12 | 41.87 | 35.55 | 29.11 |
| 600 | 51.21 | 56.13 | 48.34 | 42.12 | 36.23 | 30.87 |

Table 8. Power consumption analysis for RBFN-TDES method and existing systems

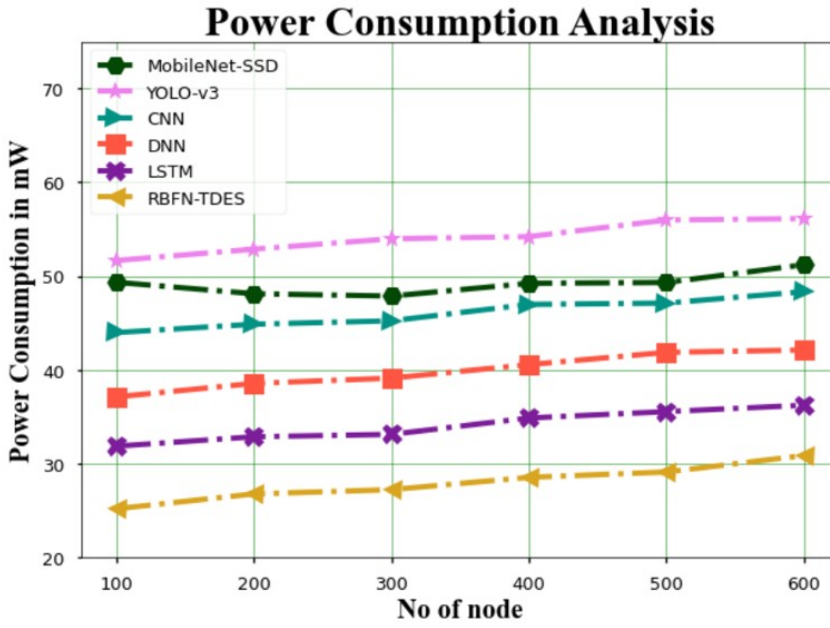


Figure 11. Power consumption analysis for RBFN-TDES method and existing systems

The outcomes of the study when comparing the power usage of the RBFN-TDES strategy to alternative ways is provided in Table 8 and Figure 11. The data results conclusively reveal that the RBFN-TDES approach is superior to the other strategies in every regard. For example, with 100 nodes, the RBFN-TDES method has consumed only 25.19 mW of power, while the other existing techniques like MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM have a power consumption of 49.34 mW, 51.66 mW, 43.98 mW, 37.12 mW, and 31.86 mW, respectively. Similarly, for 600 nodes, the RBFN-TDES method has a power consumption of 30.87 mW, while the other existing techniques like MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM have 51.21 mW, 56.13 mW, 48.34 mW, 42.12 mW, and 36.23 mW of power consumption, respectively.

### 4.2.8 Accuracy

Figure 12 and Table 9 show the results of an investigation that compares the accuracy of the RBFN-TDES method to that of other methods already in use. The graph depicts how the machine learning strategy improved accuracy. For example, with node 100, the accuracy value is 92.56 % for RBFN-TDES, whereas the MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM models have obtained accuracy of 58.34 %, 70.45 %, 64.89 %, 82.88 %, and 76.566 %, respectively. The maximum performance of

| No. of Nodes | MobileNet-SSD | YOLO-v3 | CNN | DNN | LSTM | RBFN-TDES |
|---|---|---|---|---|---|---|
| 100 | 58.34 | 70.45 | 64.89 | 82.88 | 76.56 | 92.56 |
| 200 | 59.37 | 71.22 | 65.13 | 83.12 | 77.13 | 93.67 |
| 300 | 60.12 | 72.67 | 66.87 | 84.55 | 78.45 | 94.45 |
| 400 | 61.67 | 73.98 | 67.34 | 85.66 | 79.32 | 95.23 |
| 500 | 62.33 | 74.55 | 68.19 | 86.12 | 80.67 | 96.87 |
| 600 | 63.44 | 75.12 | 69.32 | 87.43 | 81.33 | 97.21 |

Table 9. Accuracy analysis for RBFN-TDES method and existing systems

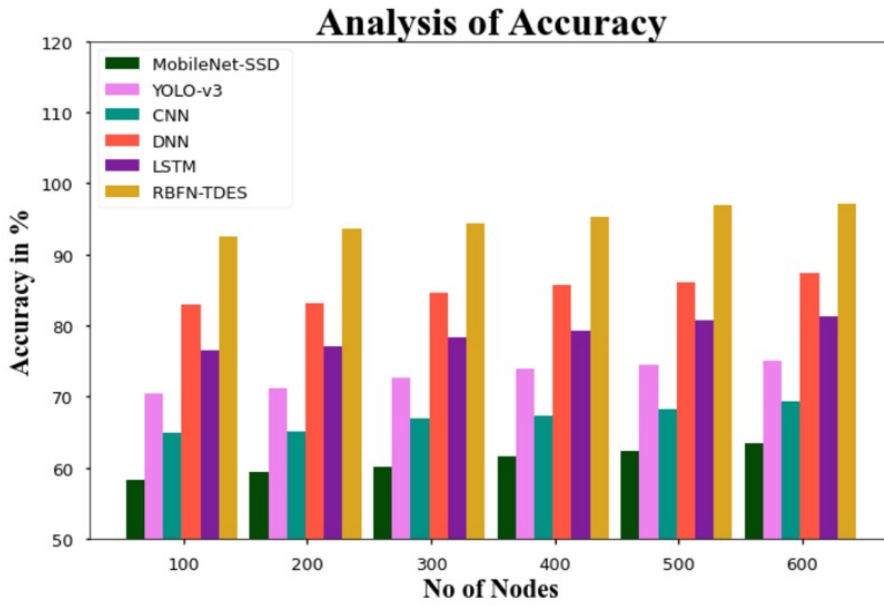


Figure 12. Accuracy analysis for RBFN-TDES method and existing systems

the RBFN-TDES model was demonstrated successfully with a diverse set of nodes. Similarly, under the 600 nodes, the accuracy value of RBFN-TDES is 97.21 %, while it is 63.44 %, 75.12 %, 69.32 %, 87.43 %, and 81.33 % for MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM models, respectively.

### 4.2.9 Training Testing Validation

Figure 13 shows the result during the training phase, the model is exposed to a dataset of CCTV footage images, which are encrypted by the Triple DES technique used in the RBFN training process. The graph depicts how the model changes over

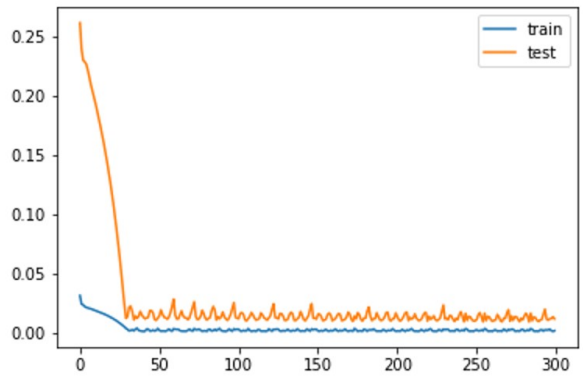


Figure 13. Training testing validation for RBFN-TDES method and existing systems

iterations, with the x-axis representing training epochs and the y-axis a relevant performance indicator, such as encryption correctness. The validation set, which consists of previously unseen data, evaluates the model's generalization skills, guaranteeing that it can properly encrypt new photos. The testing set then assesses the model's overall performance and ability to protect CCTV footage images beyond the training zone. The graph shows the model's convergence during training, its resistance to overfitting, and its effectiveness in protecting the confidentiality of CCTV data throughout the testing and validation stages.

## 5 CONCLUSIONS

The authors present a mobile video surveillance device with edge-based AI RBFN processing in this paper. This system is capable of reliably and robustly detecting and tracking people. In this paper, we look at the development of a secure CCTV approach that uses the Triple Data Encryption Standard (DES) with Radial Basis Function Networks (RBFN). The goal of intelligent video surveillance is to detect anomalies. The goal of this research was to propose that the original Data Encryption Standard (DES) algorithm be replaced with a method based on RBFN-TDES that allows for authenticated encryption. The purpose of this study was to see how easily hackers could defeat the original DES algorithm. The proposed approach, RBFN-TDES, also examines and investigates the security requirements that specific encryption properties must satisfy. As the topic's conclusion, the results of a comparison between the suggested system and the current systems are presented, along with an examination of each's security characteristics and performance metrics. Our proposed RBFN-TDES model is efficient enough to be used for securing CCTV pictures in a distributed computing environment. In this method, existing models such as MobileNet-SSD, YOLO-v3, CNN, DNN, and LSTM are used to

discover that the models have little impact on predictive performance. Based on the results of experiments using video data, snatching incidents could be identified with 97.21 % accuracy. These experiments took place in four different scenes, each with a different movement direction and either involving or not involving snatching. The proposed scheme will be tested in a variety of settings by the authors in future research. They want TPM-based protection for mobile agents. Modify the cloud computing architecture monitoring plan to improve dynamic security properties. When combined with policy-based management, the strategy can increase cloud computing security even further. More security features in unfamiliar environments may be required for the study. This may satisfy the requirements.

## 6 DECLARATIONS

**Funding:** No funds, grants were received by any of the authors.

**Conflict of interest:** There is no conflict of interest among the authors.

**Data Availability:** All data generated or analysed during this study are included in the manuscript.

**Code Availability:** Not applicable.

**Author's contributions:** Prasanna Deshpande and Ashwin G. Kothari contributed to the design and methodology of this study, the assessment of the outcomes and the writing of the manuscript.

## REFERENCES

[1] Ching, T.—Himmelstein, D. S.—Beaulieu-Jones, B. K.—Kalinin, A. A.—Do, B. T.—Way, G. P.—Ferrero, E.—Agapow, P. M.—Zietz, M.—Hoffman, M. M. et al.: Opportunities and Obstacles for Deep Learning in Biology and Medicine. Journal of the Royal Society Interface, Vol. 15, 2018, No. 141, Art. No. 20170387, doi: 10.1098/rsif.2017.0387.

[2] Zheng, Q.—Zhao, P.—Li, Y.—Wang, H.—Yang, Y.: Spectrum Interference-Based Two-Level Data Augmentation Method in Deep Learning for Automatic Modulation Classification. Neural Computing and Applications, Vol. 33, 2021, No. 13, pp. 7723–7745, doi: 10.1007/s00521-020-05514-1.

[3] Xie, Y.—Yu, J.—Guo, S.—Ding, Q.—Wang, E.: Image Encryption Scheme with Compressed Sensing Based on New Three-Dimensional Chaotic System. Entropy, Vol. 21, 2019, No. 9, Art. No. 819, doi: 10.3390/e21090819.

[4] Kaur, M.—Kumar, V.: A Comprehensive Review on Image Encryption Techniques. Archives of Computational Methods in Engineering, Vol. 27, 2020, No. 1, pp. 15–43, doi: 10.1007/s11831-018-9298-8.

[5] Gautam, K. S.—Thangavel, S. K.: Video Analytics-Based Intelligent Surveillance System for Smart Buildings. Soft Computing, Vol. 23, 2019, No. 8, pp. 2813–2837, doi: 10.1007/s00500-019-03870-2.

[6] Yu, W.—Liang, F.—He, X.—Hatcher, W. G.—Lu, C.—Lin, J.—Yang, X.: A Survey on the Edge Computing for the Internet of Things. IEEE Access, Vol. 6, 2018, pp. 6900–6919, doi: 10.1109/ACCESS.2017.2778504.

[7] Santamaria, A. F.—Raimondo, P.—Tropea, M.—De Rango, F.—Aiello, C.: An IoT Surveillance System Based on a Decentralised Architecture. Sensors, Vol. 19, 2019, No. 6, Art. No. 1469, doi: 10.3390/s19061469.

[8] Chen, N.—Chen, Y.: Smart City Surveillance at the Network Edge in the Era of IoT: Opportunities and Challenges. In: Mahmood, Z. (Ed.): Smart Cities: Development and Governance Frameworks. Springer, Cham, Computer Communications and Networks, 2018, pp. 153–176, doi: 10.1007/978-3-319-76669-0_7.

[9] Fitwi, A.—Chen, Y.—Sun, H.—Harrod, R.: Estimating Interpersonal Distance and Crowd Density with a Single-Edge Camera. Computers, Vol. 10, 2021, No. 11, Art. No. 143, doi: 10.3390/computers10110143.

[10] Chen, N.—Chen, Y.—Ye, X.—Ling, H.—Song, S.—Huang, C. T.: Smart City Surveillance in Fog Computing. In: Mavromoustakis, C. X., Mastorakis, G., Dobre, C. (Eds.): Advances in Mobile Cloud Computing and Big Data in the 5G Era. Springer, Cham, Studies in Big Data, Vol. 22, 2017, pp. 203–226, doi: 10.1007/978-3-319-45145-9_9.

[11] Fitwi, A. H.—Nagothu, D.—Chen, Y.—Blasch, E.: A Distributed Agent-Based Framework for a Constellation of Drones in a Military Operation. 2019 Winter Simulation Conference (WSC), IEEE, 2019, pp. 2548–2559, doi: 10.1109/WSC40007.2019.9004907.

[12] Fitwi, A.—Chen, Y.: Privacy-Preserving Selective Video Surveillance. 2020 29th International Conference on Computer Communications and Networks (ICCCN), IEEE, 2020, pp. 1–10, doi: 10.1109/ICCCN49398.2020.9209688.

[13] Kumar, V.—Svensson, J.: Promoting Social Change and Democracy Through Information Technology. IGI Global, 2015.

[14] Lin, L.—Purnell, N.: A World with a Billion Cameras Watching You Is Just Around the Corner. The Wall Street Journal, 2019.

[15] Fitwi, A.—Chen, Y.—Zhou, N.: An Agent-Administrator-Based Security Mechanism for Distributed Sensors and Drones for Smart Grid Monitoring. In: Kadar, I., Blasch, E. P., Grewe, L. L. (Eds.): Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII. Proceedings of SPIE, Vol. 11018, 2019, pp. 173–188, doi: 10.1117/12.2519006.

[16] Georgiou, T.—Liu, Y.—Chen, W.—Lew, M.: A Survey of Traditional and Deep Learning-Based Feature Descriptors for High Dimensional Data in Computer Vision. International Journal of Multimedia Information Retrieval, Vol. 9, 2020, No. 3, pp. 135–170, doi: 10.1007/s13735-019-00183-w.

[17] Ghosh, A. M.—Grolinger, K.: Edge-Cloud Computing for Internet of Things Data Analytics: Embedding Intelligence in the Edge with Deep Learning. IEEE Transactions on Industrial Informatics, Vol. 17, 2021, No. 3, pp. 2191–2200, doi: 10.1109/TII.2020.3008711.

[18] Shi, W.—Cao, J.—Zhang, Q.—Li, Y.—Xu, L.: Edge Computing: Vision and Challenges. IEEE Internet of Things Journal, Vol. 3, 2016, No. 5, pp. 637–646, doi:

10.1109/JIOT.2016.2579198.

[19] Yang, P.—Lyu, F.—Wu, W.—Zhang, N.—Yu, L.—Shen, X. S.: Edge Coordinated Query Configuration for Low-Latency and Accurate Video Analytics. IEEE Transactions on Industrial Informatics, Vol. 16, 2020, No. 7, pp. 4855–4864, doi: 10.1109/TII.2019.2949347.

[20] Kovač-Andrić, E.—Sheta, A.—Faris, H.—Gajdošik, M. Š.: Forecasting Ozone Concentrations in the East of Croatia Using Nonparametric Neural Network Models. Journal of Earth System Science, Vol. 125, 2016, No. 5, pp. 997–1006, doi: 10.1007/s12040-016-0705-y.

[21] Jia, W.—Zhao, D.—Ding, L.: An Optimized RBF Neural Network Algorithm Based on Partial Least Squares and Genetic Algorithm for Classification of Small Sample. Applied Soft Computing, Vol. 48, 2016, pp. 373–384, doi: 10.1016/j.asoc.2016.07.037.

[22] Barman, P.—Saha, B.: DNA Encoded Elliptic Curve Cryptography System for IoT Security. International Journal of Computational Intelligence & IoT, Vol. 2, 2019, No. 2, pp. 478–484, https://ssrn.com/abstract=3355530.

[23] Joglekar, J.—Bhutani, S.—Patel, N.—Soman, P.: Lightweight Elliptical Curve Cryptography (ECC) for Data Integrity and User Authentication in Smart Transportation IoT System. In: Karrupusamy, P., Chen, J., Shi, Y. (Eds.): Sustainable Communication Networks and Application (ICSCN 2019). Springer, Cham, Lecture Notes on Data Engineering and Communications Technologies, Vol. 39, 2020, pp. 270–278, doi: 10.1007/978-3-030-34515-0_28.

[24] Yogameena, B.—Nagananthini, C.: Computer Vision Based Crowd Disaster Avoidance System: A Survey. International Journal of Disaster Risk Reduction, Vol. 22, 2017, pp. 95–129, doi: 10.1016/j.ijdrr.2017.02.021.

[25] Machaca Arceda, V. E.—Fernández Fabián, K. M.—Laguna Laura, P. C.—Rivera Tito, J. J.—Gutiérrez Cáceres, J. C.: Fast Face Detection in Violent Video Scenes. Electronic Notes in Theoretical Computer Science, Vol. 329, 2016, pp. 5–26, doi: 10.1016/j.entcs.2016.12.002.

[26] Wang, T.—Bhuiyan, M. Z. A.—Wang, G.—Qi, L.—Wu, J.—Hayajneh, T.: Preserving Balance Between Privacy and Data Integrity in Edge-Assisted Internet of Things. IEEE Internet of Things Journal, Vol. 7, 2020, No. 4, pp. 2679–2689, doi: 10.1109/JIOT.2019.2951687.

[27] Ajay, B. S.—Rao, M.: Binary Neural Network Based Real Time Emotion Detection on an Edge Computing Device to Detect Passenger Anomaly. 2021 34th International Conference on VLSI Design and 2021 20th International Conference on Embedded Systems (VLSID), IEEE, 2021, pp. 175–180, doi: 10.1109/VLSID51830.2021.00035.

[28] Parate, M. R.—Bhurchandi, K. M.—Kothari, A. G.: Anomaly Detection in Residential Video Surveillance on Edge Devices in IoT Framework. CoRR, 2021, doi: 10.48550/arXiv.2107.04767.

[29] Ullah, W.—Ullah, A.—Hussain, T.—Muhammad, K.—Heidari, A. A.—Del Ser, J.—Baik, S. W.—De Albuquerque, V. H. C.: Artificial Intelligence of Things-Assisted Two-Stream Neural Network for Anomaly Detection in Surveillance Big Video Data. Future Generation Computer Systems, Vol. 129, 2022, pp. 286–297,

doi: 10.1016/j.future.2021.10.033.

[30] Zhao, Y.—Yin, Y.—Gui, G.: Lightweight Deep Learning Based Intelligent Edge Surveillance Techniques. IEEE Transactions on Cognitive Communications and Networking, Vol. 6, 2020, No. 4, pp. 1146–1154, doi: 10.1109/TCCN.2020.2999479.

[31] Wan, S.—Ding, S.—Chen, C.: Edge Computing Enabled Video Segmentation for Real-Time Traffic Monitoring in Internet of Vehicles. Pattern Recognition, Vol. 121, 2022, Art. No. 108146, doi: 10.1016/j.patcog.2021.108146.

[32] Chen, J.—Li, K.—Deng, Q.—Li, K.—Yu, P. S.: Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing. IEEE Transactions on Industrial Informatics, 2019, doi: 10.1109/TII.2019.2909473.

[33] Muhammad, K.—Khan, S.—Palade, V.—Mehmood, I.—De Albuquerque, V. H. C.: Edge Intelligence-Assisted Smoke Detection in Foggy Surveillance Environments. IEEE Transactions on Industrial Informatics, Vol. 16, 2019, No. 2, pp. 1067–1075, doi: 10.1109/TII.2019.2915592.

[34] Hu, Y. C.—Patel, M.—Sabella, D.—Sprecher, N.—Young, V.: Mobile Edge Computing – A Key Technology Towards 5G. ETSI White Paper No. 11. ETSI (European Telecommunications Standards Institute), 2015.

[35] Xie, K.—Ning, X.—Wang, X.—He, S.—Ning, Z.—Liu, X.—Wen, J.—Qin, Z.: An Efficient Privacy-Preserving Compressive Data Gathering Scheme in WSNs. Information Sciences, Vol. 390, 2017, pp. 82–94, doi: 10.1016/j.ins.2016.12.050.

[36] Gu, K.—Yang, L.—Yin, B.: Location Data Record Privacy Protection Based on Differential Privacy Mechanism. Information Technology and Control, Vol. 47, 2018, No. 4, pp. 639–654, doi: 10.5755/j01.itc.47.4.19320.

[37] Hamdi, S.—Bouindour, S.—Snoussi, H.—Wang, T.—Abid, M.: End-to-End Deep One-Class Learning for Anomaly Detection in UAV Video Stream. Journal of Imaging, Vol. 7, 2021, No. 5, Art. No. 90, doi: 10.3390/jimaging7050090.

[38] Zhao, T.—Nevatia, R.: Bayesian Human Segmentation in Crowded Situations. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 2003, doi: 10.1109/CVPR.2003.1211503.

[39] Johnsen, S.—Tews, A.: Real-Time Object Tracking and Classification Using a Static Camera. Proceedings of the IEEE International Conference on Robotics and Automation, Workshop on People Detection and Tracking (ICRA 2009), 2009.

**Prasanna Deshpande** is working as Engagement Manager in Tata Consultancy Service Pvt. Ltd. at Nagpur. He has 22+ years of IT industry experience and loves to work in the area of technology. His main area of interest is edge and cloud computing. Currently he is working on an ambitious project of cloud migration for one of his customers in the area of banking and finance domain. He started his career in data warehousing and business intelligence and his knowledge of databases and Big Data is helping him while migrating large applications on to cloud. He is also on the board of studies of Nagpur University and conducts different seminars, workshops and faculty development programs for different colleges as a part of Academic Interface Program of TCS. He enjoys teaching and guides students of various engineering colleges on topics related to cloud & edge computing, machine learning and Big Data.

**Ashwin G. Kothari** is working as Professor in the Electronics and Communication Engineering Department of Visvesvaraya National Institute of Technology and also had charge of associate dean of the IIIT Nagpur (Indian Institute of Information Technology), Nagpur, India. He is also one of the coordinators for the Center of Excellence of COMMBEDDED SYSTEMS: Hybridization of Communications and Embedded Systems established as World Bank assisted project of TEQIP 1.2.1. Rough sets, commbedded systems, communication, signal processing and antennas are his fields of interests. He has authored around 30 publications and have contributed three book chapters for reputed publications in the last five years. He has 21 years of experience in teaching and research. Six Ph.D. scholars have successfully completed their doctorate under his able guidance so far, and he is currently guiding five Ph.D.s. He has visited UC Barkley, University of Akron, NJIT, UT Austin (in the USA), and the University of Nagoya, Japan. He has been a resource person for many seminars, invited talks and workshops, etc. He has 8 patents filed to his credit so far. He has worked and also currently is working on many industry collaborative projects with indutries like Tektronix, Mathworks, NI, HCIT, L & T Metro, SAMEER Mumbai etc. He is also working on various projects of socio economic relevance with many social organizations like Gorakshan Sabha-Nagpur, Blind Relief Association-Nagpur, Saraswat Mandal-Nagpur and Rotary club-Tiger Capital-Nagpur under various capacities. He is a member of IEEE, IETE and ISTE and International Rough Set Society (IRSS).