Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Automated Crime Anomaly Detection in Smart Cities Using Sharkprey Optimization Algorithm and Ensembled-Machine Learning Approach

Ayush Singhal[1,2], Niraj Singhal[3], Pradeep Kumar[4]

[1]*Research Scholar, Department of Computer Science & Engineering, Shobhit Institute of Engineering & Technology, (NAAC Accredited Grade "A", Deemed to-be- University), India*
[2]*Assistant Professor, Department of Computer Science and Engineering, Meerut Institute Of Technology, India*
[3]*Director, Sir Chhotu Ram Institute of Engineering & Technology, C.C.S. University, Meerut, India*
[4]*Assistant Professor, JSS Academy of Technical Education, Noida, Uttar Pradesh, India*

Automated crime anomaly detection systems have been made possible by the influx of urban data streams brought on by the come up of smart cities. The use of machine learning to analyse and spot unusual patterns in criminal incidents is explored in this research. This system improves proactive law enforcement strategies, supports resource allocation, and aids in the development of safer and more secure urban environments by utilising real-time data from numerous sources.The first step is to gather video data from the network of security cameras that have been carefully placed throughout the smart city. With the help of this sizable video dataset, the automated crime anomaly detection system can be trained and improved so that it can learn and distinguish between typical and abnormal patterns of behaviour in a variety of urban settings. From the collected data, preprocessing is processed through Video-to-Frame Conversion, Non-Local Means (NLM) and contrast stretching approach. Sobel edge detection approach is used to Identify the Regions of Interest (ROI) in the frames for the Segmentation from the pre-processed data. To Extract features from the segmented regions, Improved Gradient Local Binary Patterns, Haralick and Gradient Interpolation-Based Hog Model is used. Refine the extracted features to remove any irrelevant or redundant features using the new hybrid optimization approach- Sharkprey Optimization Algorithm that combines the White Shark Optimizer and Osprey optimization algorithm. From the selected features, design a new ensembled-machine learning approach for crime anomaly detection by combining the K-Nearest Neighbors, Random Forest and optimized Artificial Neural Network. Tostrengthen the detection accuracy, the weight of ANN is fine-tuned by the Sharkprey Optimization. MATLAB is used for the implementation.

Keywords: Sharkprey Optimization Algorithm, Crime detection, Smart cities, Improved Gradient Local Binary Patterns, Gradient Interpolation-Based Hog Model.

1. Introduction

The rise in crime rates, when compared to other problems, is the major motives of dead lives and possessions in the twenty-first century. For the prompt and early identification of such peculiar events, an intelligent video surveillance system is the most favoredoption. The vast applications of abnormal event recognition in surveillance videos, such as crime inhibition, automated intelligent visual surveillance, and traffic security, call for a lot of attention [1,2]. Over the past few decades, numerous security cameras have been installed in both public and private spaces for efficient real-time monitoring in order to prevent accidents and ensure public safety [3].

Monitoring areas with a high likelihood of crime has become challenging due to the recent increase in urban population. There is an increase in crime and insecurity in these areas as a result of this lack of control. The development of smart city infrastructure presents a chance to develop original solutions to these issues [4].A metropolitan area that is instrumented, interconnected, and intelligent is referred to as a "smart city" and is outfitted with a variety of smart computing tech to enable more effective and supportablemanagement[5]. By providing city direction, public safety, welfare services, transit, real estate, healthcare, and tourism services, this vital infrastructure enhances the QoL for its citizens[6].A "smart city" is thought to be primarily driven by ICTs [7,8].

The term "anomaly" relates to certain unanticipated events or emergencies that differ from what is usual, expected, or standard. The management of smart cities, including traffic control and outlawexploration, heavily relies on irregularity detection [9]. Traditionally, in order to detect unusual events, it's necessary for professionals to continuously monitor the video. It always ends up being a difficult and time-consuming task. Because a workable detection method can cut down on the amount of people needed to watch videos, especially for surveillance systems, research activities allied to this task are of great reasonable significance [10].Any ecosystem for a smart city is centred on security and privacy. Every entity needs to be secured at all times; it must be ensured. A new security mechanism is being developed to cover the diverse nature of the current security requirements of a typical smart city because the classical security techniques are unable to cover all aspects and scenarios of a smart city due to a number of visible constraints like scalability, heterogeneity, power, storage, and computational capabilities [11].The risk of crime or other undesirable activities can be decreased [12] by anomaly detection, which can also increase security. ML algorithms are one of the many strategies created for anomaly detection [13];For an efficient anomaly detection system, these methods' performance must be further enhanced. Thus, developed a novel hybrid optimisation technique.

The paper is organised systematically into major sections, each of which contributes to the comprehension and assessment of the suggested automated crime anomaly detection system. The second section presents a thorough analysis of previous studies and finished work in the

field. The proposed crime anomaly detection methodology is thoroughly described in Section 3 to give readers a thorough understanding of the system's design.A comprehensive discussion of the outcomes obtained with the proposed model is provided in subdivision 4. The conclusion is found in 5th section.

The foremost contribution of the research work is: -

- To create a fresh ensembled ML strategy for crime anomaly detection.
- To remove any irrelevant features using Sharkprey Optimization Algorithm.
- The Sharkprey Optimization Algorithm is the combination of Osprey optimization algorithm& WSO.

2. Literature Review

In 2021, Cauteruccio et al.[14] have proposed initial attempt to look into anomalies in a MIoT. In order to facilitate suggested a new methodological framework. Next, defined the "forward problem" and "inverse problem" in the context of anomaly detection in a MIoT. The definition of these issues enables the investigation of the relationships between anomalies and inter-node distances, IoT network size, degree centrality, and closeness centrality of anomalous nodes. The method suggested here was used in a smart city scenario, which was a typical MIoT scenario.

In 2022, Girdhar et al.[15]have proposed a framework for CAVenabled mobility that was more adaptable and can support all cyber-based entities. Furthermore, the success of an autonomous system depends on the ability to make precise decisions in real-time, but cyberattacks on these targets can impair this ability, resulting in complex CAV accidents. Also suggested a 5Ws and 1Hbased investigation approach to deal with accidents related to cyberattacks because the current accident investigation frameworks were unable to comprehend and manage these accidents.

In 2017, Khatoun et al.[16] presented the core ideas behind smart city design while also reviewing current initiatives and projects. Then discussed various privacy and security issues, recommendations, and standards for smart cities and their services before identifying several security flaws and privacy concerns related to smart cities that was addressed.

In 2021, Ashraf et al.[17]have proposed new framework for IoT-based smart networks' defence against botnet attacks called IoTBoT-IDS. BMM and a Correntropy model were statistical learning-based techniques that IoTBoT-IDS uses to capture the typical behaviour of IoT networks. Any departure from the norm was recognised as an anomalous event. Three benchmark datasets produced by actual IoT networks were utilised to assess IoTBoT-IDS.

In 2022, Shin et al.[18] have displayed a security service's anomaly detection algorithm based on a map of surveillance. For unusual circumstances, a two-stage anomaly detection algorithm was created. This enabled the detection of anomalous circumstances, including unusual crowds and pedestrians, vehicle movement, strange objects, and temperature changes. The proposed data framework could be used for a variety of applications in the smart city because the surveillance map enables effective and integrated processing of large multimodal data from

a multi-agent.

In 2020, Rashid et al.[19] have investigated a ML-based attack and anomaly detection technique to counter IoT cybersecurity threats in a smart city. Then investigated ensemble methods like bagging, boosting, and stacking to improve the performance of the detection system, in contrast to earlier works that have concentrated on single classifiers. For the domain under discussionthe integration of feature selection, cross-validation, and multi-class classification, which has not received enough attention in the literature to date were considered.

In 2021, Ma et al.[20]have reviewed relevant literature on security in that technology as well as discuss explanations of cyber security, smart cities, and related topics. In order to achieve this, research concentrated on the four key elements of a smart city: smart grid, smart buildings, smart transportation, and smart healthcare. In particular, a summary of two deep learning methods, cyber-security initiatives, and the relationship between technology and smart cities were covered. In addition, functionally sound solutions for maintaining user privacy and cyber security in smart cities were described.

In 2018, Garcia-Font et al.[21] have evaluated SVM and Random forests as two ML algorithms to spot anomalies in a lab replicating a real use case for a smart city with heterogeneous devicesand network configurations. The experience has allowed us to demonstrate that, despite the importance of these techniques for smart cities, additional factors must be taken into account in order to accurately detect attacks. Table 1 shows research gaps byvarious authors reviews.

2.1. Research Gaps

Table 1: Various author's reviews in research gaps

Author	Aim	Research Gaps
Ullah et al. [1]	Efficient Framework for Recognising Anomalies	It is not investigated to use 3D models, GNN, or multi-instance learning formulations.
Xu et al. [2]	Multi-instance learning and an autoregressive integrated moving average model for abnormal visual event detection in edge-based Smart City surveillance	It is not focused on combining deep learning models with multi-instance learning to increase accuracy.
Hassan et al. [6]	Utilising DL and SNA Methods for Smart City Policing in Developing Continents	More data sources are not added.
Balfaqih et al. [8]	Challenges of the Development of Smart Cities and Related Information and Communication Technologies	It is not explored how the public's preferences and perceptions may differ depending on how knowledgeable they are about ICT, how innovative they are personally, and how familiar they are with the idea of smart cities.
Kumar et al. [12]	Framework for Privacy Protection and Security	It does not include creating a prototype to test the effectiveness.

3. Proposed Methodology

An important use of machine learning and computer vision technology is crime anomaly

detection in smart cities. With the development of smart city infrastructure, surveillance cameras are placed throughout the city to keep an eye on criminal activity and ensure public safety. However, manually keeping an eye on these cameras is a time-consuming and ineffective task. In light of this, automated systems for identifying criminal activity are highly desired. Systems for detecting crimes with anomalous behaviour, theft, and violence employ a number of different techniques. Typically, these systems proceed in the manner described in Figure 1.

Step 1: Input video collection

Gather video information from the smart city security cameras.

Step 2: Pre-processing

- Video-to-Frame Conversion: Convert the video data into individual frames using video-to-frame conversion technique.
- NLM: Denoise the frames using the NLM filtering approach.
- Contrast Stretching: Enhance the contrast of the image frames using the contrast stretching approach.

Step 3: Segmentation

- Identify the ROI in the frames using Sobel edge detection approach.

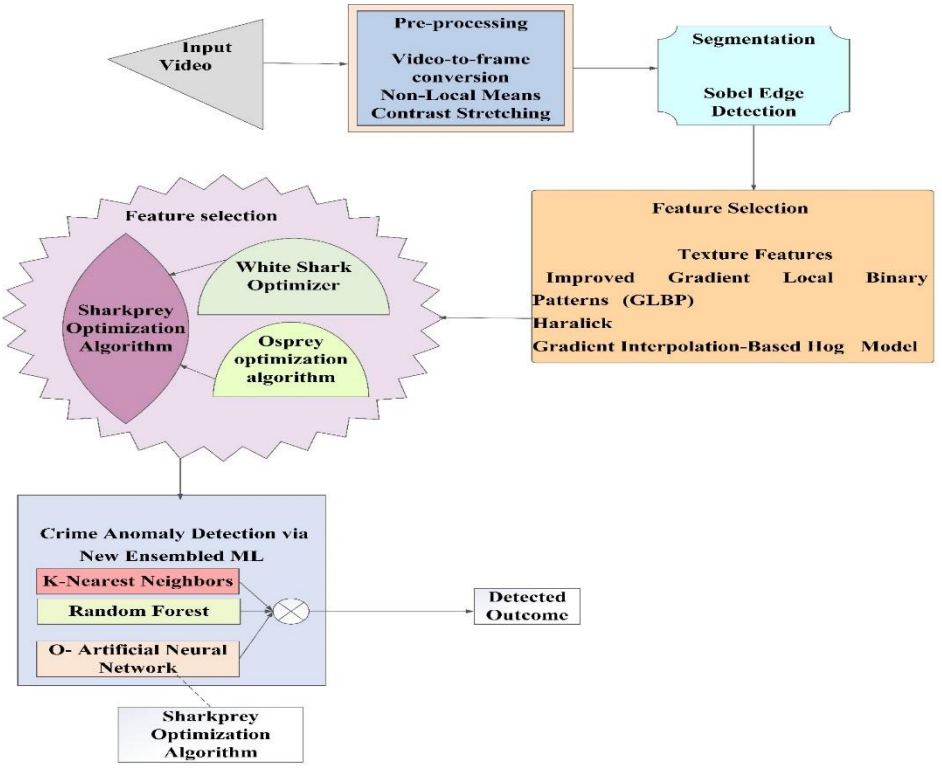

Figure 1: Proposed Flow

Step 4: Feature Extraction

Extract features from the segmented regions using various feature extraction techniques such as:

- Improved Gradient Local Binary Patterns (GLBP): Extract features such as local texture information from the ROI using the I-LBP algorithm.
- Haralick
- Gradient Interpolation-Based Hog Model (GI-HOG)

Step 5: Feature Selection

- Refine the extracted features to remove any irrelevant or redundant features using the new hybrid optimization approach- Sharkprey Optimization Algorithm (SOA) that combines the White Shark Optimizer (WSO) and Osprey optimization algorithm, respectively.

Step 6: Crime Anomaly Detection

- To design a new ensembled-machine learning approach for crime anomaly detection by combining the KNN, RF and O-ANN.
- TO enhance the detection accuracy, the weight of ANN is fine-tuned using the new hybrid optimization model.

3.1. Data collection

Real-time video of numerous urban areas, streets, and public spaces is taken when gathering video data from smart city security cameras. These cameras are carefully placed to observe and document various activities, including traffic flow, pedestrian activity, and security incidents. In order to improve situational awareness, optimise urban planning, and strengthen public safety measures, city authorities process, analyse, and use the video data they have collected.

3.2. Pre-processing

From the collected data, Pre-processing is followed through Video-to-Frame conversion, NLM and Contrast Stretching. The phase of pre-processing is depicted in Figure 2.

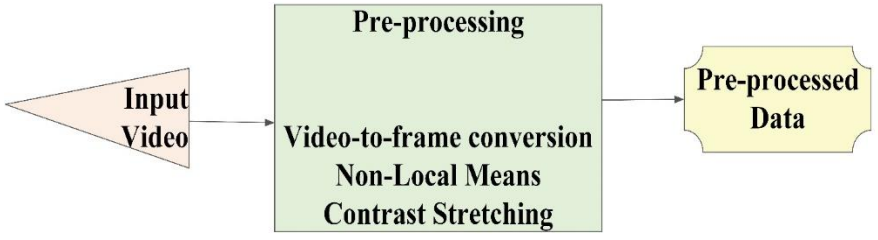

Figure 2: Pre-processing Phase

3.2.1. Video-to-Frame Conversion

The video can't be used straight for detection in order to carry out the work. In order to extract

feature extraction for rate prediction, which is crucial, the process involves modifying video toward image frames and then inside blocks.

3.2.2. NLM

The total variation-based method and the NLMmethod were introduced as noise reduction algorithms using the non-local filter method. NLM noise reduction algorithm was proposed to remove only the noise while minimising the loss of the image's inherent information. After placing areas with the same size mask all around the ROI, this algorithm compares the similarity of the intensity and edge information in an image. Additionally, the assigned weight used during image processing increases with the degree of similarity. Eq. (1) is used to represent the NLM method's fundamental equation.

$$\boldsymbol{NLM}\ |\boldsymbol{A}(\boldsymbol{m})| = \sum_{\boldsymbol{n}\in \boldsymbol{B_m}} \boldsymbol{w}(\boldsymbol{B_m}, \boldsymbol{B_n})\boldsymbol{A}(\boldsymbol{n}) \left(\boldsymbol{w}(\boldsymbol{B_m}, \boldsymbol{B_n}) = \frac{\mathbf{1}}{\boldsymbol{C}(\boldsymbol{m})} \boldsymbol{e}^{\frac{-\boldsymbol{Euc}}{\boldsymbol{h^2}}}\right) \quad (1)$$

$\boldsymbol{A}(\boldsymbol{n})$ is the intensity noise of the part of the $\boldsymbol{n}$th pixel. $\boldsymbol{B_m}$ is located close to $\boldsymbol{m}$th pixel.$\boldsymbol{w}(\boldsymbol{B_m}, \boldsymbol{B_n})$is a weighted similarity-based function.$\boldsymbol{C}(\boldsymbol{m})$is the normalization constant. $\boldsymbol{Euc}$is theEuclidean distance.

3.2.3. Contrast Stretching

Contrast stretching aims to enhance an image by extending the range of intensity values it contains in order to utilise all available values. The linear mapping of input to output values is the only one permitted for contrast stretching. Below are the steps for the contrast stretching implementation.

Step 1:Read each pixel of the image.

Step 2: Establish the upper and lower bounds for the extension of image intensity values. These lower and upper bounds is denoted by the letters $\boldsymbol{x}$ and $\boldsymbol{y}$.

Step 3:calculate the value limits in the original image (min = $\boldsymbol{m}$, max= $\boldsymbol{mx}$).

Step 4. Next for each pixel, the function below (shown in Eq. 2) is used to map the original value $\boldsymbol{r}$ to the output value $\boldsymbol{s}$.

$$\boldsymbol{s} = (\boldsymbol{r} - \boldsymbol{c})\left(\frac{\boldsymbol{y}-\boldsymbol{x}}{\boldsymbol{mx}-\boldsymbol{m}}\right) + \boldsymbol{a} \quad (2)$$

3.3. Segmentation

In this research work, From the pre-processed data, Segmentation is processed forIdentifying the ROI in the frames Sobel edge detection approach.

3.3.1. Sobel Edge detection

The Sobel edge detection operator, which computes an approximate gradient of the image intensity function, is a very popular first order edge detection operator. The Sobel operator's output at each point in the image is the corresponding norm of this gradient vector. As seen in Eq. (3) and (4), the Sobel Horizontal and vertical operator only takes into account the two orientations of $\mathbf{0}°$ and $\mathbf{90}°$ convolution kernels. The Sobel operator's advantage is its ease of calculation, but one of its drawbacks is its accuracy, which is only moderately high because it only used two convolution kernels to detect the edge of the image.

$$\boldsymbol{Hx} = \begin{bmatrix} \mathbf{-1} & \mathbf{0} & \mathbf{1} \\ \mathbf{-2} & \mathbf{0} & \mathbf{2} \\ \mathbf{-1} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (3)$$

$$\boldsymbol{Hy} = \begin{bmatrix} \mathbf{-1} & \mathbf{-2} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{2} & \mathbf{1} \end{bmatrix} \quad (4)$$

These filters in turn convolute the gradient components across the neighbouring lines or columns. Furthermore, local edge strength is determined by gradient magnitude, and the equation is given by (5).

$$\boldsymbol{GM(x,y)} = \sqrt{\boldsymbol{Hx^2 + Hy^2}} \quad (5)$$

The square and square root operations are approximated by absolute values, which are given by the Eq. (6), but performing square root and square operations for each and every pixel would be computationally expensive.

$$\boldsymbol{GM(x,y)} = \boldsymbol{|Hx| + |Hy|} \quad (6)$$

The above-mentioned process is applied individually to each and every pixel of an image, and it helps preserve the edges in images/video for which the final edge plot is calculated by absoluting the edge maps of all channels. This Eq. (6) is simple to process and aids in the preservation of edges in images/video.

3.4. Feature Extraction

In this research work, Extract features from the segmented regions using various feature extraction techniques such as:

3.4.1. Texture Features

3.4.1.1. Improved Gradient Local Binary Patterns (GLBP)

A useful texture descriptor for images is the LBP, which thresholds adjacent pixels based on the value of the current pixel. The local spatial patterns and the contrast in the grey scale in an image are effectively captured by LBP descriptors. To examine the texture pattern of images, LBP is helpful. The Gradient LBP method extends the traditional LBP by taking into account both the size and the direction of the centre gradient. Assume the eight patches that surround the centre patch, which is where the LBP concept was used. Determine the average magnitude $\boldsymbol{w}$, for each patch, which is determined by the mean of the 8 centre gradient magnitudes in a 3x3 patch as per Eq. (15).

$$\boldsymbol{w} = \frac{\mathbf{1}}{\mathbf{8}} \sum_{\boldsymbol{a=1}}^{\mathbf{8}} \boldsymbol{|b_a - b_c|} \quad (15)$$

It is calculated that the magnitude index $\boldsymbol{\alpha}$, which conceptually corresponds to a centre patch as per Eq. (16).

$$\boldsymbol{\alpha} = \sum_{\boldsymbol{a=1}}^{\mathbf{8}} \boldsymbol{v(w_c - w_a)}, \boldsymbol{v(t)} = \begin{cases} \mathbf{1}, & \boldsymbol{t \geq 0} \\ \mathbf{0}, & \boldsymbol{t < 0} \end{cases} \quad (16)$$

The average magnitudes of a patch's centre and border are given as $\boldsymbol{w_c}$ and $\boldsymbol{w_a}$ $(\boldsymbol{a = 1, 2, \ldots, 8})$

respectively. The surrounding patches that have a larger average magnitude than the centre patches are counted to create the magnitude index. Two bits, referred to as magnitude bits, are used to express this information, and they are added to the LBP binary code. As a result, the LBP number's range of [0, 255] is increased by the two extra bits that are closely related to the average magnitude to the range of [0, 1023].

Improved GLBP serves as a valuable tool for crime anomaly detection in smart cities. I-GLBP is employed to analyze video streams from surveillance cameras, aiming to identify irregular or suspicious activities. It operates by capturing local patterns and textures within image regions. GLBP extracts features from these regions, representing their unique characteristics.Anomaly detection algorithms are then utilized to discern deviations from expected or normal behavior. Whether it's identifying unexpected movements, unattended objects, or unusual crowd dynamics, GLBP can flag anomalies in real-time. Furthermore, integrating techniques like the Scharr gradient operator and dimensionality reduction enhances the method's precision and efficiency.By continuously monitoring video feeds from surveillance cameras, the Improved GLBP-based system can promptly raise alerts when unusual events occur. This proactive approach contributes to the development of safer and more secure Smart Cities by assisting law enforcement and city authorities in addressing criminal activities and security threats.

Algorithm 1 - IGLBP
Initialize parameters and data structures
Access video stream from surveillance cameras
Set the anomaly detection threshold
Initialize an empty list to store detected anomalies
Main loop for continuous surveillance
Capture the next frame from the surveillance feed
Step 1: Compute derivatives - Use Scharr operator
Step 2: Compute gradient magnitude
Step 3: Apply threshold and obtain SGLTP codes
Step 4: Compute positive and negative GLTP coded images
Step 5: Split coded images into m x n regions
Step 6: Compute positive and negative GLTP histograms
Step 7: Store histograms in respective lists
Step 8: Concatenate histograms from each region
Step 9: Apply PCA to reduce dimensionality
Step 10: Detect anomalies -Store the frame and anomaly score

Step 11: Check for continuous activity or specific duration to confirm anomaly

Step 12: Clear old anomalies

End of the main loop

3.4.1.2. Haralick

A set of statistical measures called Haralick features, also referred to as texture features, is used in image analysis to quantify the texture or pattern within an image region. They are derived from co-occurrence matrices and capture characteristics like contrast, homogeneity, and entropy, offering crucial data for tasks like object recognition, segmentation, and classification of images. These characteristics are especially helpful for characterising the fine details present in images in areas like computer vision and remote sensing.

- Variance

The spread or dispersion of pixel intensities within an image region is measured statistically as variance. Greater intensity level variability is indicated by higher variance values, which also reflect a variety of textures or patterns.The mathematical model is shown in Eq. (7).

$$\sum_i \sum_j (i-j)^2 P(i,j) \quad (7)$$

- Entropy

The randomness or uncertainty in the distribution of pixel intensities within an image region is measured by entropy. While lower values denote more consistent or predictable patterns, higher entropy values suggest a more complex and varied texture.The mathematical model is shown in Eq. (8).

$$\sum_i \sum_j P(i-j) \quad log \quad P(i,j) \quad (8)$$

- Energy

The sum of squared pixel intensities in a region of an image's co-occurrence matrix is represented by energy, also known as the sum of squared elements or contrast. Regions with distinct and well-defined texture patterns are correlated with high energy values.The mathematical model is shown in Eq. (9).

$$\sum_i \sum_j P^2(i,j) \quad (9)$$

- Homogeneity

In an image region's co-occurrence matrix, homogeneity quantifies how similar the pixel intensity values are. Indicating smoother and less complex textures when higher homogeneity values are seen, it quantifies how closely the intensities are spaced from one another.The mathematical model is shown in Eq. (10).

$$\sum_i \sum_j \frac{P(i-j)}{(|i-j|)} \quad (10)$$

- 3rd order Moment

The skewness of the distribution of pixel intensities in an image region is captured by the third order moment. It reveals the asymmetry of the intensity values around the mean and sheds light on the texture pattern's overall shape.The mathematical model is shown in Eq. (11).

$$\sum_i \sum_j (i-j)^3 P(i,j) \quad (11)$$

- Inverse Variance

The spread of pixel intensities within an image region is represented by inverse variance, which is the reciprocal of variance. It draws attention to areas with intensity values that are comparatively concentrated or closely clustered.The mathematical model is shown in Eq. (12).

$$\sum_i \sum_j P(i-j) \ / P(i,j)^2 \quad (12)$$

3.4.1.3. Gradient Interpolation-Based Hog Model (GI-HOG)

HOG is a feature descriptor used in computer vision for object detection and recognition. It represents the distribution of local gradient or edge orientation patterns in an image as a histogram. HOG features are commonly used in various machine learning models to recognize objects. HOG with a simplified formula is given by Eq. (1) and Eq. (2).

Gradient Calculation:For each pixel in an image, calculate the gradient magnitude and orientation.$|g| = \sqrt{(g_x^2 + g_y^2)}$ (1)

Orientation of Gradient (Θ): $\theta = arctan(g_y / g_x)$ (2)

Where,$g_x$and $g_y$represent the horizontal and vertical gradient components, which measure how pixel intensity changes in the x and y directions, respectively.In the gradient interpolation-based HOG model, interpolation utilizes the estimated gradient direction within each cell to improve accuracy, enhancing HOG for capturing local edge orientations.An edge pixel can be interpolated using Eq. (3).

$$M_h^e = \sum_{r=0}^{n_g-1} M_r^{gc} \times c_{g(r)}^{gc} * c_{d(r)}^{gc} \quad (3)$$

where the distance coefficient, gradient coefficient, and picture intensity of the RTH pixel with a gradient correlation less than T are represented by the values M_r^{gc}, $c_{g(r)}^{gc}$, and $c_{d(r)}^{gc}$. One may compute c_g and $c_d$using Eq. (4) to Eq. (6).

$$c_g = e^{-n\sqrt{d_g}} \quad (4)$$

$$\hat{c}_d = e^{-(x^2+y^2)/2\sigma^2} \quad (5)$$

$$c_d = \frac{\hat{c}_d}{max\,\hat{c}_d} \quad (6)$$

Here,n is set to 4, and σ is chosen as 1.5 based on simulations. The variables x and y in Eq.

(5) represent the pixel's distances from the center pixel. CD is computed only once for a fixed $\mathbf{9 \times 9}$ patch because each pixel in it has a constant distance from the center. However, $\boldsymbol{c_g}$is dynamically calculated for each pixel due to varying gradient correlations. Before interpolation, both $\boldsymbol{c_d}$and $\boldsymbol{c_g}$coefficients need normalization to satisfy specified conditions based on Eq. (7).

$$\sum_{r=0}^{n_g-1} c_{g(r)}^{gc} * c_{d(r)}^{gc} = 1 \quad (7)$$

After interpolating an edge pixel moving vertically, transitional pixels connected to the left and right are scanned and interpolated in the same direction. Unknown transitional pixels are categorized based on their position and interpolated accordingly. When an edge moves horizontally, transitional pixels above and below are interpolated similarly. In exceptional cases where no clear direction is determined, a transitional pixel is interpolated using the mean of the two closest pixels in that direction through line averaging.

3.5. Feature Selection

Combining the White Shark Optimizer (WSO) and Osprey optimisation algorithm, the Sharkprey Optimisation Algorithm (SOA) was developed to select features which are extracted. Figure 3 shows the phase of Feature Selection.

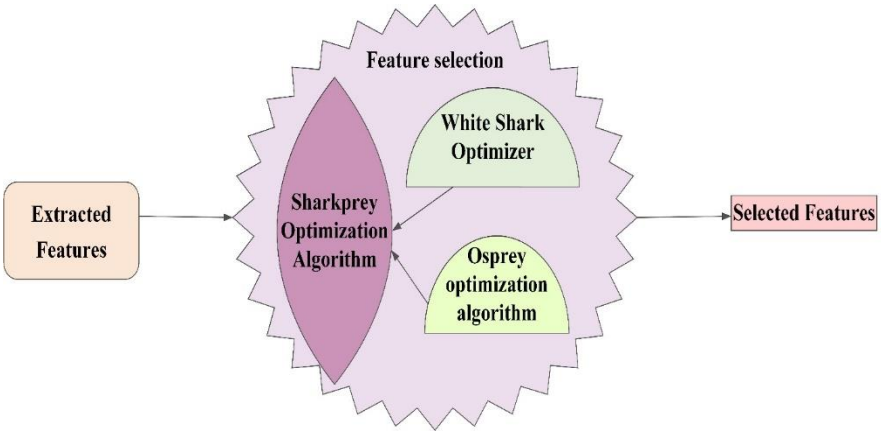

Figure 3: Feature Selection phase

3.5.1. Shark prey Optimization Algorithm

WSO is an intelligent metaheuristic model inspired by the hunting behaviors of white sharks, designed for solving optimization problems in continuous search spaces. It leverages principles of scent and vision in predation. OOAis an optimization method inspired by the hunting behaviors of ospreys, which are birds of prey known for their precision in catching fish.

Initialization:OOA is a population-based method for resolving optimization issues. In OOA, every individual in the population is an osprey and is viewed as a potential solution to the issue being optimized. These potential answers are mathematically expressed as vectors, where each member denotes a different variable in the problem. As per Eq. 8, the population of ospreys is kept in a matrix designated as $\boldsymbol{X}$, where each column denotes a problem variable and each row

denotes an osprey (a potential solution). Each osprey's position is iteratively modified during the optimization process to look for better answers inside the issue space. Each osprey's objective function is calculated in order to determine the quality of the candidate solutions, producing a vector $\boldsymbol{f}$ that contains the values of all ospreys' objective functions as given in Eq. (9). Finding the best candidate solution, or the osprey with the lowest objective function value (best value), is the aim of OOA. Similarly, the option with the greatest objective function value (worst value) represents the worst candidate solution. The best candidate solution is modified after each iteration of the algorithm because the position of the ospreys is changed to improve the candidate solutions.OOA seeks to converge to the best or nearly best solution for the given optimization problem by repeatedly going through this process. The method effectively explores and exploits the issue space through the use of population-based search, random initialization, and iterative updates, ultimately producing a workable solution to the given optimization problem.For each osprey in the population, the objective function is calculated. The objective function provides a numerical measure of how well the current osprey (potential solution) performs with respect to the problem's goals and constraints. It evaluates the fitness or quality of the candidate solution.

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_i} \\ \boldsymbol{x_n} \end{bmatrix}_{n \times m} = \begin{bmatrix} \boldsymbol{x_{1,1}} & \boldsymbol{x_{1,j}} & \boldsymbol{x_{1,m}} \\ \boldsymbol{x_{i,1}} & \boldsymbol{x_{i,j}} & \boldsymbol{x_{i,m}} \\ \boldsymbol{x_{n,1}} & \boldsymbol{x_{n,j}} & \boldsymbol{x_{n,m}} \end{bmatrix}_{n \times m} \tag{8}$$

$$\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_i} \\ \boldsymbol{f_n} \end{bmatrix}_{n \times 1} = \begin{bmatrix} \boldsymbol{f(x_1)} \\ \boldsymbol{f(x_i)} \\ \boldsymbol{f(x_n)} \end{bmatrix}_{n \times 1} \tag{9}$$

Phase 1: Identification of Positions and Hunting of Fish (Exploration)

In the first stage of population update, the behavior of ospreys was simulated. Ospreys, with their excellent eyesight, locate and chase fish underwater. By modeling this hunting behavior, the positions of the ospreys in the search space changed significantly, enhancing the algorithm's ability to explore and avoid local optima. Higher-value positions of other ospreys served as undersea fishes for each osprey in the OOA design. The set of such fish for each osprey was determined as per Eq. (10).

$$\boldsymbol{fp_i} = \{\boldsymbol{x_k} \mid \boldsymbol{k} \in \{\boldsymbol{1}, \boldsymbol{2}, \ldots, \boldsymbol{N}\} \wedge \boldsymbol{f_k} < \boldsymbol{f_i}\} \cup \{\boldsymbol{X_{best}}\} \tag{10}$$

Where, $\boldsymbol{fp_i}$represents the fish position set for the $\boldsymbol{ith}$ osprey,$\boldsymbol{X_{best}}$is the best solution found among the ospreys, Eq. (20) defines $\boldsymbol{fp_i}$as a set of positions for the $\boldsymbol{ith}$ osprey. It includes positions ($\boldsymbol{x_k}$) for which the corresponding function values ($\boldsymbol{f_k}$) are less than $\boldsymbol{f_i}$(the function value associated with the $\boldsymbol{ith}$ osprey), as well as the best solution $\boldsymbol{X_{best}}$among all ospreys. This set helps the $\boldsymbol{ith}$ osprey in its search and exploration process.An osprey randomly locates a fish and strikes it. Using Eq. (11), the osprey's new position is determined based on its movement toward the fish, as described in Eq. (12) when it improves the objective function.

$$\boldsymbol{x_{i,j}^{p1}} = \boldsymbol{x_{i,j}} + \boldsymbol{r_{i,j}} \cdot \left(\boldsymbol{sf_{i,j}} - \boldsymbol{ir_{i,j}} \cdot \boldsymbol{x_{i,j}}\right)$$

$$\boldsymbol{x_{i,j}^{p1}} = \boldsymbol{x_{i,j}} + \boldsymbol{v_{k+1}^{i}}$$

$$x_{i,j}^{p1} = \begin{cases} x_{i,j}^{p1}, lb_j \le x_{i,j}^{p1} \le ub_j \\ lb_j, x_{i,j}^{p1} < lb_j \\ ub_j, x_{i,j}^{p1} > ub_j \end{cases} \quad (11)$$

$$x_i = \begin{cases} x_i^{p1}, f_i^{p1} < f_i \\ x_i, else \end{cases} \quad (12)$$

In the first phase, x_i^{p1}denotes the ith osprey's updated position.$x_{i,j}^{p1}$ represents its jth dimension, while $f_i^{p1}$is its fitness function. sf_isignifies the fish selected for the ith osprey. These parameters contribute to the osprey's movement strategy and position optimization.WSO incorporates oscillating motions based on wave frequency detection, and this behavior is complemented with the velocity concept and it is calculated as per Eq. (13). This integration enhances the OOA method's capabilities in identifying positions and hunting for fish within the optimization process.

$$v_{k+1}^i = \mu\left(v_k^i + s_1\left[w_{gbest_k} - w_k^i\right]r_{i,j} + s_2\left[w_{best}^{v_k^i} - w_k^i\right]r_{i,j}\right) \quad (13)$$

Where, v_{k+1}^i and $v_k^i$are the updated velocities of the ith white shark in iterations $k+1$ and$k$. $s_1$and $s_2$represent the influences of white sharks monitoring $w_{gbest_k}$and $w_{best}^{v_k^i}$. $w_k^i$and w_{gbest_k}denote the global optimum position at the kth iteration and the location of the ith white shark in iteration k. $r_{i,j}$is the random numbers within [0,1]. $w_{best}^{v_k^i}$represents the ith best-defined position to the swarm during the kth frequency iteration, andμ is the constriction factor in WSO.

Phase 2: Carrying the Fish to the Suitable Location Position (Exploitation)

In the second stage, the osprey mimics its real behavior by carrying and consuming the caught fish in a favourable location. Simulating this, the population's positions are updated with slight changes, enhancing local search and promoting convergence to improved solutions. Each member's suitable fish-eating position is initially determined using Eq. (14), replicating the natural osprey behavior.As per Eq. (15), the element's position was updated if it improved the objective function.

$$x_{i,j}^{p2} = x_{i,j} + \frac{lb_j + r\cdot(ub_j - lb_j)}{t}, i = 1, 2, \dots, n, j = 1, 2, \dots, m, t = 1, 2, \dots, T$$

$$x_{i,j}^{p2} = \begin{cases} x_{i,j}^{p2}, lb_j \le x_{i,j}^{p2} \le ub_j \\ lb_j, x_{i,j}^{p2} < lb_j \\ ub_j, x_{i,j}^{p2} > ub_j \end{cases} \quad (14)$$

$$x_i = \begin{cases} x_i^{p2}, f_i^{p2} < f_i \\ x_i, else \end{cases} \quad (15)$$

In second phase, x_i^{p2}represents the ith osprey's new position, while$x_{i,j}^{p2}$ signifies its jth dimension. f_i^{p2}represents its fitness function, and$r_{i,j}$ are random numbers within [0,1]. $t$denotes the iteration count, and$T$ represents the total number of iterations.

3.6. Crime Anomaly Detection via New Ensembled ML Approach

To combine the KNN, RF, and O- ANN in a new ensembled machine learning approach for crime anomaly detection. The weight of the ANN is adjusted using the new hybrid optimisation model in order to improve detection accuracy.

3.6.1.KNN

Lazy learning is a type that includes the K-Nearest Neighbours algorithm. The neighbours are elect from a group of stuff or objects with similar characteristics or values; This collection of objects can be considered the training dataset. The Euclidean algorithm connects two points using a straight line as the distance within them. Before using the KNN algorithm on a dataset, it must first be prepared by scaling its parameters to an adjusted scale. The length of the line segment that connects points C and D is the euclidean distance between them. Eq.(16) provides the formula for Euclidean distance:

$$l(C,D) = \sqrt{(h_1 - h_2)^2 + (m_1 - m_2)^2} \quad (16)$$

3.6.2. RF

To make each tree dependent on the values of a random vector sampled independently and with the same distribution for all trees in the forest, RF is a combination of tree predictors using DT. The strength of each individual tree in the forest and the correlation between them determine the generalisation error of a forest of tree classifiers. They are more noise-resistant and robust. It is a supervised classification algorithm that is used for prediction, and it is regarded as the best because it uses a lot more trees in the forest and has better accuracy than DT.In most cases, the trees are trained independently, and their predictions are averaged together. Depending on the nature of the issue, RF can be used for regression as well as classification. The RF algorithm is provided below.

Step 1:Choose k features at random from all m features, where $k << m$.

Step 2:Determine the node "d" using the best split point within the k features.

Step 3:Utilise the best split to divide the node into daughter nodes.

Step 4:Up until there are l number of nodes, repeat steps 1 to 3 as necessary.

Step 5:Create a forest by reiterating steps 1 through 4 n times to produce n trees.

The first step is to subtract the k features from the total m features. The next step is to randomly choose k features from each tree in order to use the best split approach to locate the root node. The daughter nodes are then calculated using the same best split methodology as for the crime detection record. The tree is constructed similarly, starting with the root node and continuing until all of the leaf nodes are produced from the attributes. The RF used to make crime detection is formed by this randomly generated tree.

3.6.3 Optimized ANN

An ANN is made up of several processing units, also referred to as neurons, connected in a predetermined topology. It can learn by doing and generalise from small, noisy, and incomplete amounts of data. Many data-intensive applications have used ANN with success. A neural network consists of an input layer, several hidden layers, and an output layer. Every layer contains a large number of neurons.The input layer is where data enters the neural network; the hidden layers process it, and the output layer is where the output can be retrieved. Figure 4 depicts a typical neural network model with a hidden layer.

There are 'n' numbers of inputs available for a neuron in this network and every input is associated with a weight on it. $\boldsymbol{x_0}$ is the bias value which is added to the input of the activation function. Let $\boldsymbol{x_1, x_2, \ldots, x_n}$ are the inputs to a neuron and let are $\boldsymbol{w_1, w_2, \ldots, w_n}$ weights, bias is $\boldsymbol{bias}$and then 'a' is the out of the neuron it is shown in Eq. (17).

$$\boldsymbol{a = act\ (\sum_{i=0}^{n} w_i x_i + bias)} \qquad (17)$$

where $\boldsymbol{act}$ is the activation function used to obtain that layer's output and feed it as an input to the following layer. The nodes and weights that make up an ANN typically need to be learned based on the patterns that are provided.

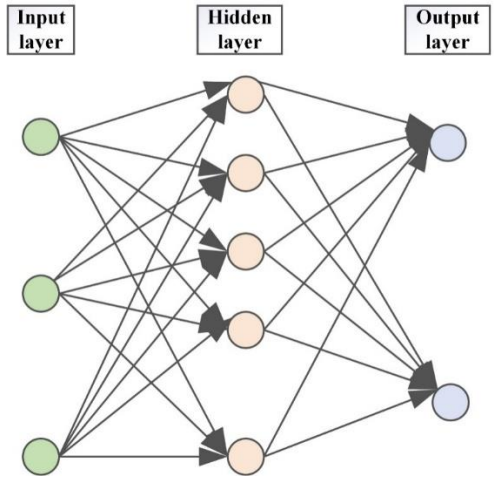

Figure 4: Structure of Neural Network

4. Result and Discussion

4.1. Dataset Description

The UCF-Crime dataset consists of 128 hours of videos and is quite extensive. 1900 uncut, lengthy actuality surveillance videos are included, along with 13 realistic anomalies such as mistreat, arson, assault, road accidents, robberies, eruptions, fights, stealing, shootings, robbery, and vandalism. The reason these anomalies were select that they considerably affect public safety.Two tasks can be proficient with this dataset. First, ananalytical overview of

anomalies is performed, taking into account all anomalies in one group and all regular activities in another. Second, for identifying each of the 13 odd behaviours.

4.2.Experimental Setup

The MATLAB programming is used to implement the suggested model. This section elaborates the graphical analysis for Automated Crime Anomaly Detection in Smart Cities. In this section, the execution metrics used for evaluating the performance of proposed model are discussed and it was compared with existing techniques.

4.2.1. Performance Metrics

i. Accuracy

The accuracy metric, which calculates the proportion of accurate predictions to all predictions, is one of the most straightforward Classification metrics to employ. The formula is mathematically expressed by Eq. (18).

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad (18)$$

ii. Precision

The percentage of false positives to all absolute cases is used to gauge precision. Precision formula is mathematically expressed by Eq. (19).

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad (19)$$

iii. Sensitivity

Sensitivity is the likelihood of a positive test result, presuming the subject is in reality positive. The sensitivity formula is mathematically expressed by Eq. (20).

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (20)$$

iv. Specificity

The ratio of accurately predicted negative outcomes over all negative outcomes is known as specificity. The formula for specificity is mathematically shown in Eq. (21).

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN+FP}} \quad (21)$$

v. F-Measure

The number strikes a balance between making sure that each definition explicitly only refers to one type of information item and fully determine each data bit. The formula for F-Measure is mathematically shown in Eq. (22).

$$\boldsymbol{F\_Score} = \frac{\boldsymbol{Presision.Recall}}{\boldsymbol{Presision+ Recall}} \quad (22)$$

vi. Matthew's correlation coefficient (MCC)

The effectiveness of binary classification models is assessed using the MCC metric. MCC takes TP, TN, FN, and FP into account, making it a trustworthy metric for evaluating the

effectiveness of binary classifiers. The level of correlation between the predictor and the actual labels is determined by MCC. The formula for MCC is mathematically shown in Eq. (23).

$$\boldsymbol{MCC} = \frac{(\boldsymbol{TP}*\boldsymbol{True\ Negative})-(\boldsymbol{FP}*\boldsymbol{FN})}{\sqrt{(\boldsymbol{TP}+\boldsymbol{FP})(\boldsymbol{TP}+\boldsymbol{FN})(\boldsymbol{FP}+\boldsymbol{TN})(\boldsymbol{TN}+\boldsymbol{FN})}} \quad (23)$$

vii. NPV

A statistical measure known as NPV assesses the dependability of a negative test result in a population of people who have a particular condition. By dividing the total population that is free from the condition by the number of actual negative effects, it is done on purpose. The effectiveness of detection is evaluated using NPV. The formula for NPV is mathematically shown in Eq. (24).

$$\boldsymbol{NPV} = \frac{\boldsymbol{TN}}{\boldsymbol{TN}+\boldsymbol{FN}} \quad (24)$$

viii. FPR

In order to determine the false positive rate, the entire number of bad cases is divided by the number of adverse facts that were mistakenly classified as unfavourable. Eq. (25) provides a mathematical representation of the FPR formula.

$$\boldsymbol{False\ Positive\ Rate} = \frac{\boldsymbol{FP}}{\boldsymbol{FP}+\boldsymbol{TN}} \quad (25)$$

ix. FNR

It is also known as the "miss rate," is a measurement of the likelihood that a test would fail to discover a true positive. Eq. (26) presents the mathematical formula for FNR.

$$\boldsymbol{False\ Negative\ Rate} = \frac{\boldsymbol{FN}}{\boldsymbol{TP}+\boldsymbol{FN}} \quad (26)$$

4.2.2. Overall comparison of the proposed and Existing methodology

The performance of the proposed method is compared with the existing techniques like ANN, KNN, Random Forest, Decision Tree and AdaBoost. The considered model is analysed and relating to Sensitivity, recall, MCC, precision, specificity, F- score, FNR, FPR, NPV and accuracy. Table 2 shows the proposed and Existing of performance metrics.

Table 2: Proposed Vs Exiting

Methods	Sen	Spec	Acc	Precision	F-Measure	NPV	FPR	FNR	MCC
Proposed	0.902	0.992	0.986	0.902	0.902	0.9923	0.007	0.097	0.894
ANN	0.854	0.988	0.979	0.854	0.854	0.988	0.011	0.145	0.843
KNN	0.819	0.986	0.974	0.819	0.819	0.986	0.013	0.180	0.805
Random Forest	0.834	0.987	0.976	0.834	0.834	0.987	0.012	0.165	0.821
Decision Tree	0.624	0.971	0.946	0.624	0.624	0.971	0.028	0.375	0.595
AdaBoost	0.697	0.976	0.956	0.697	0.697	0.976	0.023	0.302	0.674

The reported performance metrics are capable of accurately identifying positive instances, as evidenced by their sensitivity score of 0.902. It excels at accurately classifying negative cases

with a specificity of 0.992. The overall accuracy is 0.986, demonstrating the system's ability to make accurate predictions for both classes. An F-measure of 0.902 indicates a balanced performance between precision and sensitivity, and a precision score of 0.902 emphasises its precision in positive predictions. The technique's high NPV of 0.9923 further demonstrates its accuracy in predicting negative cases. Furthermore, a low FPR of 0.007 and FNR of 0.097 show how well it minimises both varieties of classification errors. Its overall quality of predictions is reflected in its MCC score of 0.894, emphasising its strong performance in this classification task.The proposed approach consistently outperforms existing techniques in a number of crucial areas. It outperforms the ANN method's sensitivity of 0.854, accurately identifying positive instances with a sensitivity of 0.902. Similar to the ANN method's specificity of 0.988, the proposed method displays exceptional specificity at 0.992, demonstrating its competence in correctly classifying negative instances. The Proposed method also maintains a significantly lower FPR of 0.007 and FNR of 0.097, underscoring its ability to reduce incorrect classifications.

The KNN method achieves a specificity of 0.986 and a sensitivity of 0.819, respectively, demonstrating its accuracy in classifying positive and negative instances. It shows a general ability to make accurate predictions with an accuracy of 0.974. The accuracy of its positive predictions is highlighted by its precision score of 0.819, and the F-measure of 0.819 highlights a balanced trade-off between precision and sensitivity. The method's NPV of 0.986 demonstrates how well it can predict unfavourable outcomes. It also keeps a low FNR of 0.180 and an FPR of 0.013, which together show that it is making an effort to reduce both types of classification errors. The MCC score of 0.805 reflects the general predictability of the data.With a sensitivity score of 0.834, Random Forest has the highest success rate in correctly identifying positive instances. It excels at precisely classifying negative instances, with a specificity of 0.987. The method still generates accurate predictions overall, as evidenced by its accuracy of 0.976. The F-measure of 0.834 indicates a balanced performance between precision and sensitivity, and its precision score of 0.834 emphasises its accuracy in positive predictions. The high NPV of 0.987 demonstrates its capacity to accurately forecast adverse cases. Further evidence of the method's ability to reduce both types of classification errors is provided by its low FPR of 0.012 and FNR of 0.165. The MCC score of 0.821 indicates the general predictability of the result.

The sensitivity of the decision tree is 0.624. It demonstrates its ability to precisely classify negative instances with a specificity of 0.971. The overall accuracy, which measures the ability to make accurate predictions for both classes, is 0.946. The precision score of 0.624, however, indicates that its positive prediction accuracy is comparatively lower. The 0.624 F-measure highlights a balanced trade-off between sensitivity and precision. The method is effective at correctly predicting negative cases, as shown by its NPV of 0.971. It minimises the misclassification of true negatives as positives, maintaining a relatively low FPR of 0.028. Despite being relatively high, FNR of 0.375 shows that it performs better at identifying false positives than true ones. The MCC score of 0.595, which summarises its overall prediction quality, concludes.The sensitivity of the AdaBoost method is 0.697, demonstrating its accuracy in identifying positive instances. It demonstrates its skill at accurately classifying negative instances with a specificity of 0.976. The overall accuracy, which measures the system's ability to produce accurate predictions for both classes, is 0.956. The accuracy of its

optimistic predictions is highlighted by the precision score of 0.697. The balanced trade-off between precision and sensitivity is highlighted by the F-measure of 0.697. The method's NPV of 0.976 demonstrates how well it can predict negative cases. It minimises the misclassification of true negatives as positives as evidenced by its relatively low FPR of 0.023. With a FNR of 0.302, it appears to identify fewer true positives. The MCC score of 0.674 reflects the general accuracy of the predictions. Figure 5 shows the graphical representation of Performance resulta.

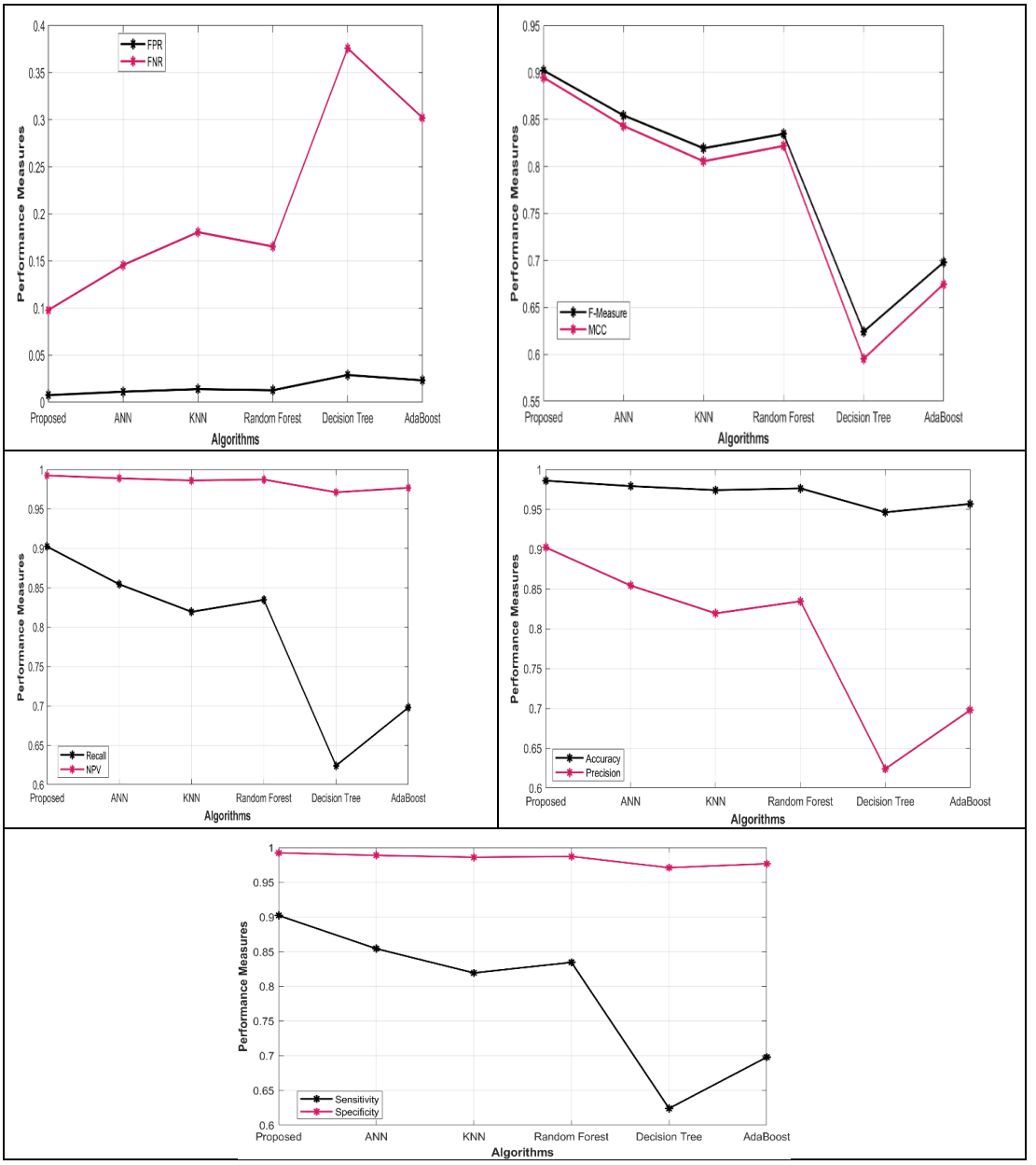

Figure 5: Graphical Representation of Performance Metrics

5. Conclusion

A powerful solution to the problem of crime anomaly detection in smart cities was provided by the combination of the Sharkprey Optimisation Algorithm with an Ensembled-Machine Learning Approach. With this cutting-edge pairing, anomaly identification was more accurate and efficient, and the system was shown to be flexible in a variety of urban environments.The network of security cameras that was positioned throughout the smart city must first be queried for video data. The automated crime anomaly detection system could be trained and enhanced with the aid of this sizeable video dataset in order for it to recognise normal and abnormal patterns of behaviour in a variety of urban settings. Preprocessing from the collected data is done using the Video-to-Frame Conversion, NLM, and contrast stretching approaches. To identify the ROI in the frames for segmentation from the pre-processed data, the Sobel edge detection approach was used. Utilising I-GLBP, Haralick, and GI-HOG, features from the segmented regions were extracted.Using the Sharkprey Optimisation Algorithm, which is the combination of WSO and Osprey optimisation algorithm, refine the extracted features to remove any unnecessary or redundant features. Created a new ensembled machine learning method for detecting crime anomalies from the chosen features by fusing the KNN, RF, and O-ANN. The weight of the ANN was adjusted using the new hybrid optimisation model in order to improve detection accuracy. The implementation employs throughMATLAB.

References

1. Ullah, W., Ullah, A., Hussain, T., Khan, Z.A. and Baik, S.W., 2021. An efficient anomaly recognition framework using an attention residual LSTM in surveillance videos. Sensors, 21(8), p.2811.
2. Xu, X., Liu, L., Zhang, L., Li, P. and Chen, J., 2020. Abnormal visual event detection based on multi-instance learning and autoregressive integrated moving average model in edge-based Smart City surveillance. Software: Practice and Experience, 50(5), pp.476-488.
3. Shao, Z., Cai, J. and Wang, Z., 2017. Smart monitoring cameras driven intelligent processing to big surveillance video data. IEEE Transactions on Big Data, 4(1), pp.105-116.
4. Kitchin, R. and Dodge, M., 2020. The (in) security of smart cities: Vulnerabilities, risks, mitigation, and prevention. In Smart Cities and Innovative Urban Technologies (pp. 47-65). Routledge.
5. Saravanan, K., Julie, E.G. and Robinson, Y.H., 2019. Smart cities & IoT: Evolution of applications, architectures & technologies, present scenarios & future dream. Internet of things and big data analytics for smart generation, pp.135-151.
6. Hassan, S.U., Shabbir, M., Iqbal, S., Said, A., Kamiran, F., Nawaz, R. and Saif, U., 2021. Leveraging deep learning and SNA approaches for smart city policing in the developing world. International Journal of Information Management, 56, p.102045.
7. Attaran, H., Kheibari, N. and Bahrepour, D., 2022. Toward integrated smart city: A new model for implementation and design challenges. GeoJournal, 87(Suppl 4), pp.511-526.
8. Balfaqih, M. and Alharbi, S.A., 2022. Associated Information and Communication Technologies Challenges of Smart City Development. Sustainability, 14(23), p.16240.
9. Chen, N. and Chen, Y., 2022. Anomalous vehicle recognition in smart urban traffic monitoring as an edge service. Future Internet, 14(2), p.54.

10. Zhao, Y., Man, K.L., Smith, J. and Guan, S.U., 2022. A novel two-stream structure for video anomaly detection in smart city management. The Journal of Supercomputing, 78(3), pp.3940-3954.
11. Ahad, M.A., Paiva, S., Tripathi, G. and Feroz, N., 2020. Enabling technologies and sustainable smart cities. Sustainable cities and society, 61, p.102301.
12. Kumar, P., Kumar, R., Srivastava, G., Gupta, G.P., Tripathi, R., Gadekallu, T.R. and Xiong, N.N., 2021. PPSF: A privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities. IEEE Transactions on Network Science and Engineering, 8(3), pp.2326-2341.
13. Islam, M., Dukyil, A.S., Alyahya, S. and Habib, S., 2023. An IoT Enable Anomaly Detection System for Smart City Surveillance. Sensors, 23(4), p.2358.
14. Cauteruccio, F., Cinelli, L., Corradini, E., Terracina, G., Ursino, D., Virgili, L., Savaglio, C., Liotta, A. and Fortino, G., 2021. A framework for anomaly detection and classification in Multiple IoT scenarios. Future Generation Computer Systems, 114, pp.322-335.
15. Girdhar, M., You, Y., Song, T.J., Ghosh, S. and Hong, J., 2022. Post-accident cyberattack event analysis for connected and automated vehicles. IEEE Access, 10, pp.83176-83194.
16. Khatoun, R. and Zeadally, S., 2017. Cybersecurity and privacy solutions in smart cities. IEEE Communications Magazine, 55(3), pp.51-59.
17. Ashraf, J., Keshk, M., Moustafa, N., Abdel-Basset, M., Khurshid, H., Bakhshi, A.D. and Mostafa, R.R., 2021. IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. Sustainable Cities and Society, 72, p.103041.
18. Shin, H., Na, K.I., Chang, J. and Uhm, T., 2022. Multimodal layer surveillance map based on anomaly detection using multi-agents for smart city security. ETRI Journal, 44(2), pp.183-193.
19. Rashid, M.M., Kamruzzaman, J., Hassan, M.M., Imam, T. and Gordon, S., 2020. Cyberattacks detection in iot-based smart city applications using machine learning techniques. International Journal of environmental research and public health, 17(24), p.9347.
20. Ma, C., 2021. Smart city and cyber-security; technologies used, leading challenges and future recommendations. Energy Reports, 7, pp.7999-8012.
21. Garcia-Font, V., Garrigues, C. and Rifà-Pous, H., 2018. Difficulties and challenges of anomaly detection in smart cities: A laboratory analysis. Sensors, 18(10), p.3198.
22. https://www.dropbox.com/sh/75v5ehq4cdg5g5g/AABvnJSwZI7zXb8_myBA0CLHa?dl=0