OPEN ACCESS
Journal of Multidisciplinary Applied Natural Science Vol. 6 No. 3 (2026) https://doi.org/10.47352/jmans.2774-3047.455
Check for updates
Research Article

A Novel Intelligent Drone Assist Framework for Identifying Crime in Public

Vidyarani Hamppayanamalige Jayaprakash*, Girija Sanjeevaiah, Mohankumar Venugopal, Ganesh Hamppatanamalige Jayaprakash, and Nishchitha Malligere Harendra Kumar

Received : February 2, 2026	Revised : April 10, 2026	Accepted : April 28, 2026	Online : July 10, 2026

Abstract
The rapid emergence of digital technologies and the increased of using intelligent devices have led to a significant increase in crime incidents. Therefore, there is an emerging need to develop highly accurate and efficient detection frameworks. Hence, this work introduces an intelligent drone-assisted crime-detection system based on a novel Puma Multilayer Perceptron Detection Framework. The images are collected from a Kaggle dataset. The image dataset is pre-processed in a Python environment, with image quality improved by removing noise and normalizing the images. Further, discriminative image features are selected by the Puma Optimization Algorithm. It efficiently selects the optimal features by balancing between detection accuracy and dimensionality reduction. The optimized feature set is then classified using a multilayer perceptron. All these processes aim to accurately detect criminal activities. The efficiency of the proposed model is evaluated using standard metrics, including accuracy, precision, recall, F-score, and error rate, and the results are compared with those of traditional detection approaches. The experimental results show that the proposed framework improves detection accuracy and reliability, making it effective for intelligent drone-based crime monitoring.

Keywords: crime detection, drone, feature analysis, multilayer perceptron, Puma optimization

1. INTRODUCTION

The rapid progress and commoditisation of drones, also known as unmanned aerial vehicles (UAVs) [1], have enabled them to become an integral part of a variety of industries, including public safety, transportation, and logistics, as well as environmental monitoring and smart city solutions [2]. The versatility and mobility of drones, combined with their ability to deliver real-time data from diverse environments, have enhanced their overall appeal for use in both the civil and government sectors [3]. As the use of drones continues to grow in public surveillance, they will be increasingly utilised to monitor additional congested areas, streamline information collection, observe gatherings, and assist law enforcement agencies in upholding public safety [4]. In addition to the advantages offered by drone-based surveillance, there is now also the potential for an increase in the volume of cyber-attack threats [5]. Today, drone acts as a cyber physical system [6], thus connecting each component through ground control stations, wireless communication links, GPS modules, and cloud-based analytics, rendering today's drones particularly vulnerable to cyber threats in the form of DoS attacks [7], spoofing, jamming, hijacking, tampering with data, or unauthorised access to the systems [8]. UAVs have quickly become a vital part of life in today's society due to their growing dependence on UAVs for many tasks, from commercial deliveries to military operations [9]. However, as UAVs continue to evolve into highly integrated cyber-physical systems, their physical and cyber components have experienced substantial increases in security vulnerabilities, giving rise to new threats to UAV use [10]. UAV systems, as cyber-physical systems, have a unique architecture that combines sensors, communication modules, processors, and actuators, creating a large attack surface for potential cyber threats [11].

Cyber-attacks such as spoofing, tampering, hijacking, DDoS, and data interception can result in numerous and serious issues [12], including loss of vehicle control, breaches of privacy, tampering with data, crashes, and unauthorized access to classified intelligence gathered in the area of operations [13]. The inability of current security systems to protect

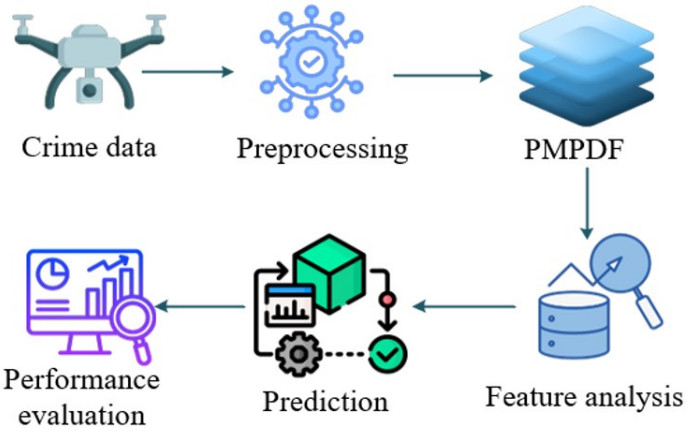


Figure 1. Proposed architecture.

against both cyber and physical threats underscores the need for robust mechanisms to detect and mitigate threats to UAVs as integrated cyber-physical systems rather than as individual network or hardware components [14]. When it comes to detecting unwanted access and unusual UAV behavior, intrusion detection systems (IDSs) are essential [15]. However, the intricacy of UAV operations is not adequately captured by conventional IDS models that only rely on physical or network characteristics [16]. The development of artificial intelligence (AI), machine learning (ML), and deep learning (DL) has enabled contemporary IDSs to more precisely identify complex attack patterns by analyzing multi-domain UAV data [17]. By facilitating scalable, robust threat detection across networked UAVs and ground control stations, distributed IDS systems substantially improve security [18].

The AI, ML, and DL models are increasingly being used to improve the detection of UAV intrusions [19]. The advantages of using AI models include their ability to learn complex patterns, identify anomalies in sensor and communication data, and adapt to evolving threats more efficiently than traditional signature-based IDSs [20]. However, many computationally intensive ML- and DL-based IDSs still require substantial processing power and therefore cannot be processed in real time on energy-constrained UAVs [21]. In addition, many approaches for UAV ID rely solely on cyber or physical telemetry, resulting in limited visibility into the entire system [22]. Centralized AI models add latency, communication overhead, and potential single-point failures to UAV IDS systems [23]. Therefore, there is a need for an integrated and intelligent Cyber-Physical IDS approach for UAV-assisted Surveillance. Compared to more conventional techniques like Principal Component Analysis (PCA), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO), the Puma Optimization Algorithm (POA) offers a number of advantages during the feature selection phase [24]-[26]. POA uses a balanced exploration–exploitation method inspired by adaptive hunting behavior, which allows it to efficiently search high-dimensional feature spaces, in contrast to traditional techniques that frequently rely on linear assumptions or may become stuck in local optima. POA achieves an ideal trade-off between performance and computing complexity by mathematically optimizing an objective function that simultaneously maximizes classification accuracy and minimizes feature subset size. As a result, highly discriminative and non-redundant characteristics are chosen. In comparison to other metaheuristic techniques, POA exhibits faster convergence rates and better stability in experiments, which results in higher precision, recall, and F-score values in crime detection tasks. Additionally, POA improves the multilayer perceptron classifier's generalization capacity by more effectively eliminating noisy and irrelevant features, which eventually raises detection accuracy and reliability in actual drone surveillance situations [26]-[28].

The key contribution of the work is described as follows, the drone-based cybercrime dataset is initially collected and imported into the Python system. The collected Dataset undergoes

Table 1. The algorithm for the proposed work.

Algorithm: 1 PMPDF

BEGIN
INPUT: Drone / Kaggle Image Dataset
OUTPUT: Crime or Non-Crime Classification
1. LOAD dataset from source
2. PREPROCESS IMAGES
FOR each image IN dataset DO
READ image
RESIZE image to standard dimensions
REMOVE noise from image
NORMALIZE pixel values
STORE processed image
END FOR
3. FEATURE EXTRACTION
FOR each preprocessed image DO
EXTRACT features (texture, edges, patterns)
ADD features to feature set
END FOR
4. FEATURE SELECTION USING PUMA OPTIMIZATION
INITIALIZE population of feature subsets
REPEAT until stopping condition is met DO
FOR each subset IN population DO
COMPUTE fitness value
END FOR
SELECT best feature subsets
UPDATE population using Puma optimization strategy
END REPEAT
SELECT optimal feature subset
5. DATA SPLITTING
SPLIT dataset into training set and testing set
6. MODEL TRAINING (MLP)
INITIALIZE Multilayer Perceptron
TRAIN MLP using training dataset
7. CLASSIFICATION
FOR each test sample DO
PREDICT class label using trained MLP
STORE prediction
END FOR
8. ALERT SYSTEM
FOR each prediction DO
IF prediction = Crime THEN
TRIGGER alert
ELSE
CONTINUE monitoring
END IF
END FOR
END

preprocessing to enhance data quality by removing noise. Moreover, a novel Puma Multilayer Perceptron Detection Framework (PMPDF) is introduced for cybercrime identification. Hence, the POA performs feature analysis, selecting the relevant features. The PMPDF also uses the selected features to predict cybercrime. Finally, the performance of the proposed model is validated with a few standard performance metrics and compared with conventional approaches.

2. MATERIALS AND METHODS

At first, a drone-based crime dataset is collected from Kaggle and fed into the Python platform. Preprocessing of the dataset is performed, including noise removal and normalization, to ensure high-quality input for modeling. Hence, a novel PMPDF is developed, in which the POA is employed to effectively execute intelligent feature selection, selecting the most relevant indicators of crime. The selected features are subsequently fed into the following process, which leverages a multilayer perceptron to accurately predict cybercrime. The proposed architecture is displayed in Figure 1. The model's performance is validated using metrics such as recall, precision, accuracy, F-score, and error rate.

2.1. PMPDF Process

POA [29] is a nature-inspired metaheuristic optimization technique based on the intelligent hunting behaviors and movement strategies of pumas. Pumas are solitary predators that balance exploration-examining large areas in search of prey-with exploitation-effectively attacking the best target. In POA, every puma represents one candidate solution in the search space, wherein its position corresponds to a potential optimal solution. The initial phase creates a random collection of Pumas, which are iteratively updated in their movements based on their ability to adaptively define their search and attack behavior while hunting. During the exploration stage, Pumas are encouraged to roam over a large area of their environment to reduce the risk of convergence early in the search space. During the exploitation stage, Pumas focus on developing solutions rather than being widely dispersed throughout the environment. Because of its strong balance between global search and local refinement, POA became highly efficient for feature selection problems, identifying the most relevant features while reducing dimensionality and computational cost. The MLP is the most widely used model for classification and prediction problems in numerous applications. Basically, it consists of an input layer, one or more hidden

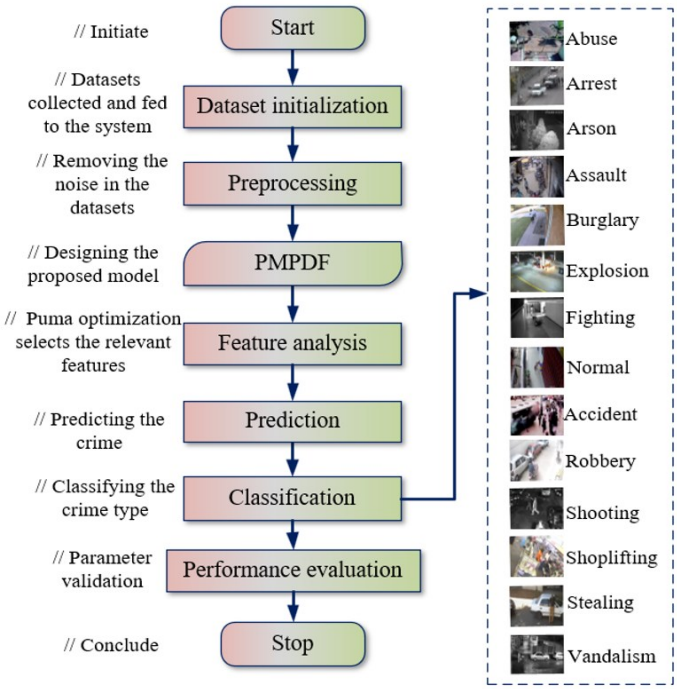


Figure 2. PMPDF flowchart.

Table 2. Experimental setup and training configuration.

Parameter	Value / Description	Purpose
Hardware	Intel i7 Processor, 16 GB RAM, GPU (optional – NVIDIA CUDA-enabled)	Ensure efficient computation
Software Environment	Python (TensorFlow / PyTorch / Scikit-learn) 3.10	Model implementation
Operating System	Windows 10	Execution platform
Training–Testing Split	70% Training / 30% Testing	Model evaluation
Learning Rate	0.001	Controls weight updates
Optimizer	Adam Optimizer	Faster convergence
Batch Size	32	Efficient gradient computation
Epochs	50–100	Ensure proper training
MLP Architecture	Input layer + 2 hidden layers + output layer	Model structure
Hidden Neurons	64 and 32 neurons (hidden layers)	Feature learning capacity
Activation Function	ReLU (hidden), Softmax (output)	Non-linearity and classification
Loss Function	Cross-Entropy Loss	Measure prediction error
Evaluation Metrics	Accuracy, Precision, Recall, F-score, Error Rate	Performance assessment

Table 3. Dataset preprocessing, augmentation, and feature extraction details.

Step	Description	Technique Used	Purpose
Dataset Source	UCF crime dataset (video-based)	Frame extraction	Convert videos into image frames for model training
Frame Extraction	Frames sampled at fixed intervals (e.g., 1 fps)	Uniform temporal sampling	Reduce redundancy while preserving key information
Image Resizing	All frames resized to fixed dimensions (e.g., 224×224)	Interpolation methods	Ensure uniform input size
Noise Removal	Removal of unwanted distortions	Gaussian filtering / median filtering	Improve image quality
Normalization	Pixel values scaled (0–1 or −1 to 1)	Min-max normalization	Stabilize training process
Data Augmentation	Rotation, flipping, scaling	Geometric transformations	Increase dataset diversity and prevent overfitting
Feature Extraction	Extraction of discriminative features	Statistical + texture features	Represent meaningful patterns
Feature Selection	Optimal feature subset selection	Puma Optimization Algorithm (POA)	Reduce dimensionality and improve accuracy

Table 4. Dataset description.

Types	Total = 45,304	Training = 25,625	Testing = 19,679
Abuse	2,655	1,590	1,065
Arrest	5,679	3,699	1,980
Arson	2,203	943	1,260
Assault	1,855	1,321	534
Burglary	3,072	1,326	1,746
Explosion	2,385	987	1,398
Fighting	2,181	1,487	694
Normal	7,142	3,572	3,570
Road accidents	2,522	1,438	1,084
Robbery	2,601	1,855	746
Shooting	3,113	1,502	1,611
Shoplifting	4,787	2,355	2,432
Stealing	3,521	2,653	868
Vandalism	1,588	897	691

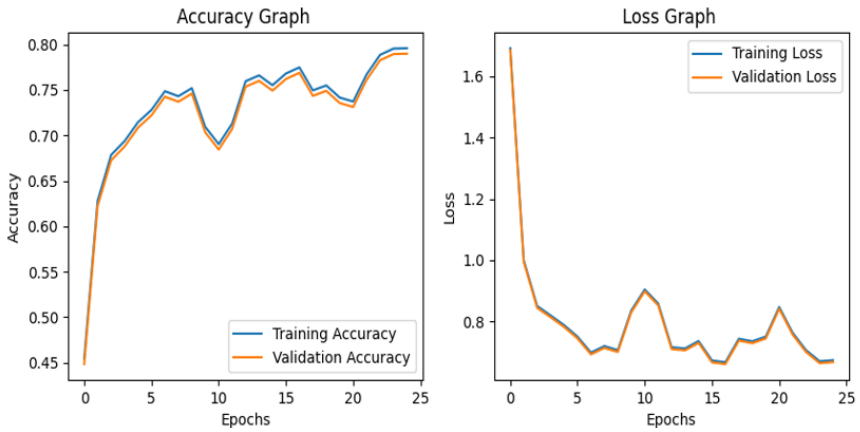


Figure 3. Training and testing graph.

layers, and an output layer, with neurons in successive layers connected to all others in a fully connected manner. The MLP crime detection effectively models non-linear relationships between pre-selected features and correctly distinguishes crime from everyday activities. It has proven itself to be adaptive, robust, and capable of strong generalization when combined with an optimized feature selection technique, such as the POA, to fulfill the requirements of intelligent detection systems.

2.1.1. Data Initialization

The crime detection system first involves collecting a drone-based crime image dataset from the Kaggle repository. A dataset of drone-captured images depicts activities related to various aspects of crime and normal scenarios. The dataset images are imported into the Python environment, and the image preprocessing and modeling are performed. The dataset initialization is described by Equation (1).

$$D_i = \{D_1, D_2, D_3, \ldots, D_N\} \quad \textbf{(1)}$$

The crime data is described as D , the i^{th} sample in the database is exposed as D_i and N is the total number of images. The collected data is fed to the further processing layers.

Table 5. Sample outcome.

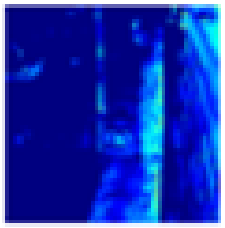
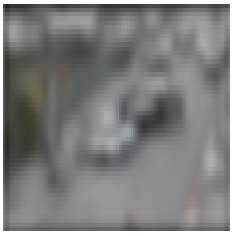
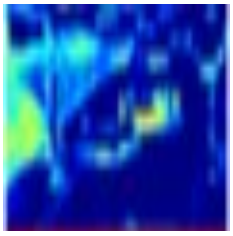
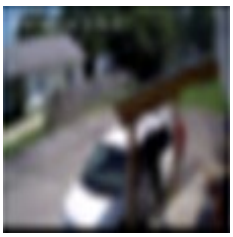
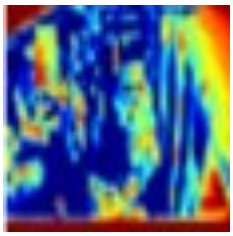
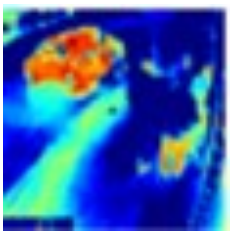
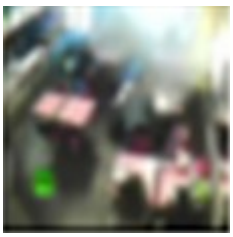
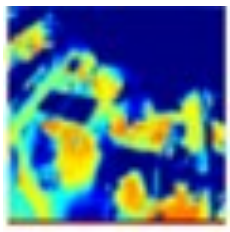
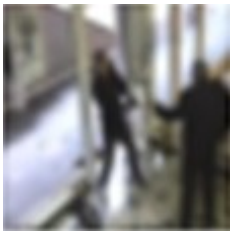
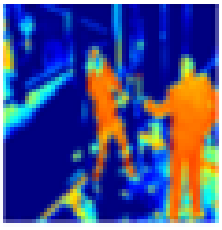
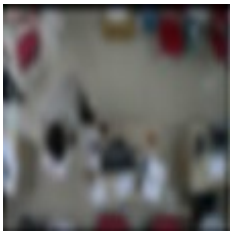
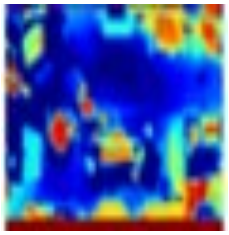
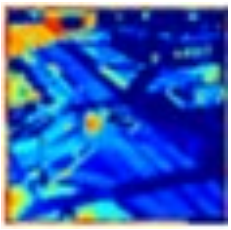
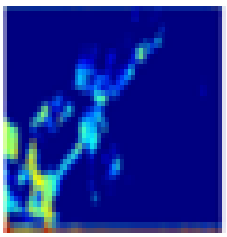
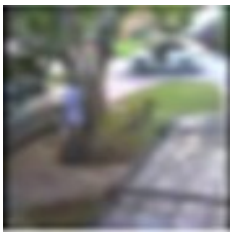
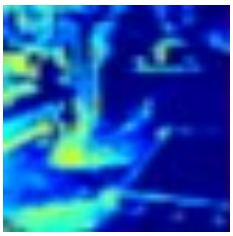
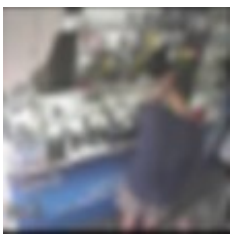
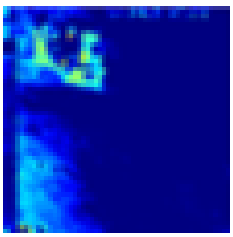
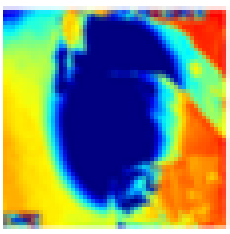
Input image	Pre-processed image	Feature analysis image	Classification
			Abuse
			Arrest
			Arson
			Assault
			Burglary
			Explosion
			Fighting

Table 5. *Cont.*

Input image	Pre-processed image	Feature analysis image	Classification
			Normal
			Road accidents
			Robbery
			Shooting
			Shoplifting
			Stealing
			Vandalism

2.1.2. Preprocessing

Preprocessing is an essential step in the proposed PMPDF framework because it improves the quality of the input data and enables reliable learning. Since the dataset contains images, preprocessing steps can be considered to eliminate unnecessary distortions. Moreover, to avoid a dominance effect in learning due to features with large numeric scales, pixel intensity normalization is performed to standardize them to a predefined scale, usually [0,1]. It is processed by Equation (2).

$$P_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad \textbf{(2)}$$

Here, the normalized image output is described as P_i, pixel intensity value of the input image is denoted as X_i, minimum pixel intesity vaue in the databse is expsoed as X_{min} and the maximum pixel intesity vaue in the databse is expsoed as X_{max}. The preprocessing data is expressed as P function, the standard deviation is denoted as σ, and the min-max scalar normalization function is expressed as N. Thus, preprocessing enhanced data quality, enabling more accurate prediction.

2.1.3. Feature Analysis

Feature analysis focused on identifying which characteristic in preprocessed images is most informative and can be used effectively for crime identification. It analyses the meaningful image features such as texture, shape, edges, and deep visual features. POA is used for intelligent image feature selection to balance image classification accuracy with dimensionality reduction. Inspired by puma hunting behavior, this algorithm successfully explores and exploits the image space using a search strategy based on a learned hunting behavior model that aims to identify the optimal subset of image features with better learning capabilities without significantly increasing complexity. The feature analysis is determined as Equation (3).

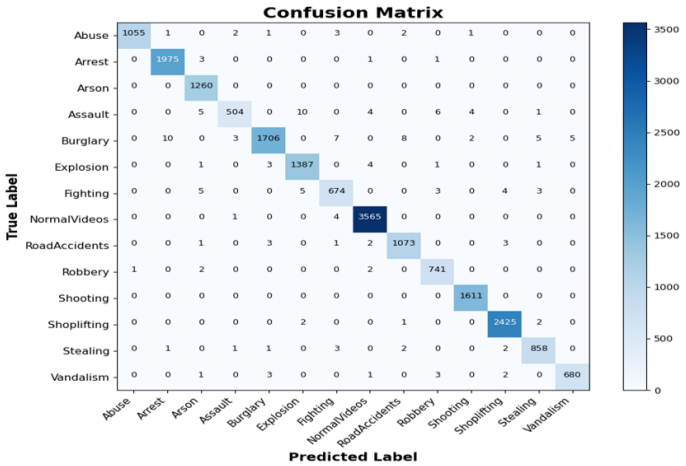


Figure 4. Confusion matrix.

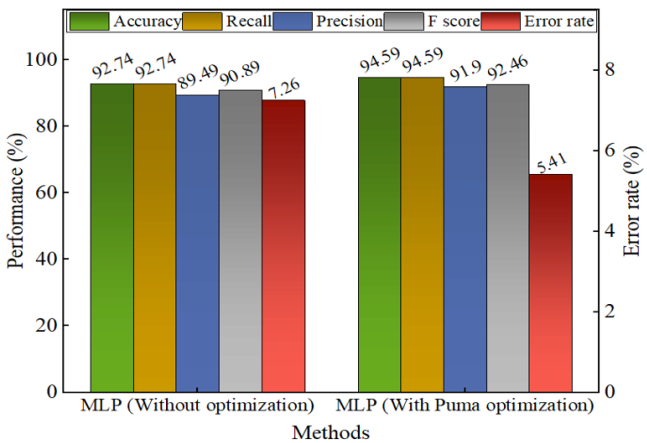


Figure 5. Performance with and without optimization.

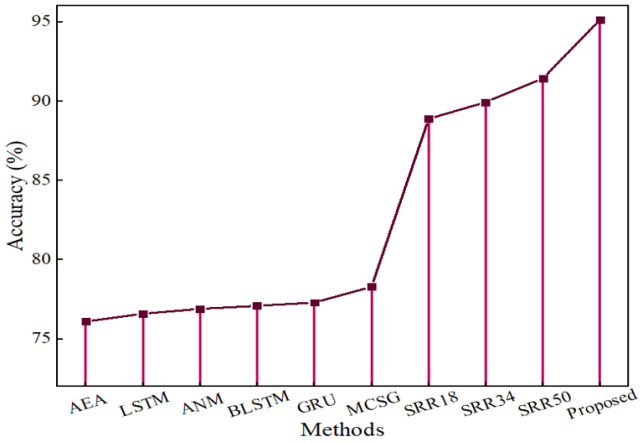


Figure 6. Accuracy comparison.

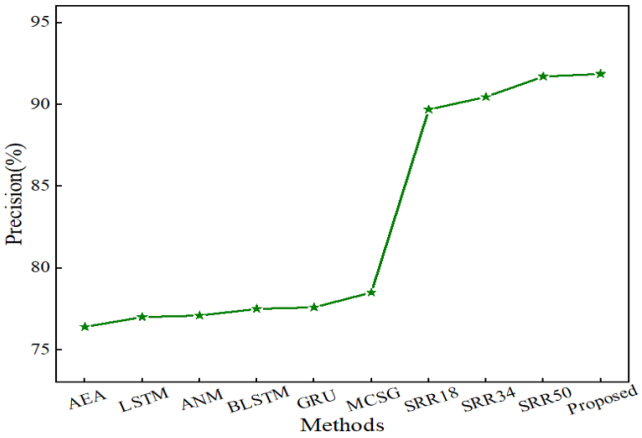


Figure 7. Precision comparison.

$$F_i = [f_1, f_2, \ldots, f_d] \quad \textbf{(3)}$$

The ith image feature vector is exposed as F_i, extracted featue is exposed as f_k and f_d is the total number of extracted features per image. This enables global exploration as well as local exploitation. The algorithm iteratively updates the positions until a stopping criterion is met, yielding an optimal feature subset for classification.

2.1.4. Prediction

In the prediction phase, the optimized feature subset derived in the feature analysis phase is used with an MLP classifier. Here, in the MLP, all input features are processed through layers of interconnected neurons that apply non-linear transformations. The output layer uses a sigmoid function to produce a probability. MLP learns efficient decision boundaries across all input features to achieve accurate, reliable crime detection. The MLP classifcation is described as Equation (4).

$$\hat{y} = f(Wx + b) \quad \textbf{(4)}$$

Here, the predicted output vector is exposed as $\hat{y}$, x is the input feature vector, MLP layers weight matrix is expsoed as W, bias vector is exposed as b and the ReU activation function is exposed as $f(.)$. Then the crime class detection is given as Equation (5),

$$C = \arg\max(\hat{y}) \quad \textbf{(5)}$$

where, predicted crime class label is denoted as C and arg max is the maximum probability of return index.

The POA candidate representation is exposed in Equation (6),

$$S_j = [s_1, s_2, \ldots, s_d] \quad \textbf{(6)}$$

Here, S_j is the jth candidate solution of puma agent, binary value is demoted as S_k, d is the total number of messages, now the POA objective function is exposed by Equation (7).

$$F^* = \arg\max(\alpha.Acc(F) - \beta.|F|) \quad \textbf{(7)}$$

Here, F^* is the optimal feature subset, classification accuracy of feature subset is denoted as $Acc(F)$, $|F|$ is the number of selected features, α is the weight for accuracy importance, β is the penalty weight of feature reduction. The POA update rule is exposed in Equation (8).

$$S_j^{t+1} = S_j^t + r_2(S_{best} - |S_j^t|) - r_2(S_{rand} - S_j^t) \quad \textbf{(8)}$$

Here, S_j^t is the current solution of iteration t, S_j^{t+1} is the updated solution of $t + 1$, best solution is S_{best}, randomly selected solutions are S_{rand}, random values are r_1,r_2 and iteration number is denoted as t. Then stoping criteria is $|Acc^{t+1} - Acc^t| < \epsilon$, where, minimum threshold improvement is exposed as ϵ, accuracy of next iteration is exposed as Acc^{t+1}, and Acc^t is accuracy of iteration. Here, the crime is classified into 14 classes, like abuse, arrest, arson, assault, burglary, explosion, fighting, normal, accident, robbery, shooting, shoplifting, stealing, and vandalism. The algorithm for the proposed work is described as follows Table 1.

The suggested model's sequential operation is shown in the flowchart in Figure 2. The PMPDF processes from initialization to classification are provided in a pseudo-code format in algorithm 1. In contrast to traditional optimization-based feature selection techniques like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), the proposed PMPDF framework exhibits novelty through the integration of the POA with a Multilayer Perceptron (MLP) classifier in the feature selection and classification pipeline. POA introduces a more adaptive search mechanism inspired by puma hunting behavior,

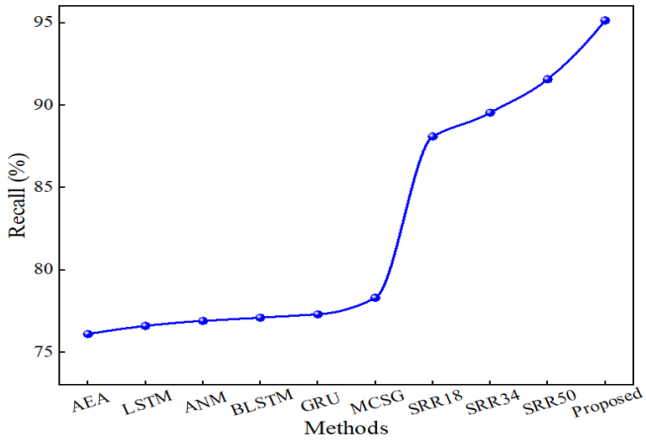


Figure 8. Recall comparison.

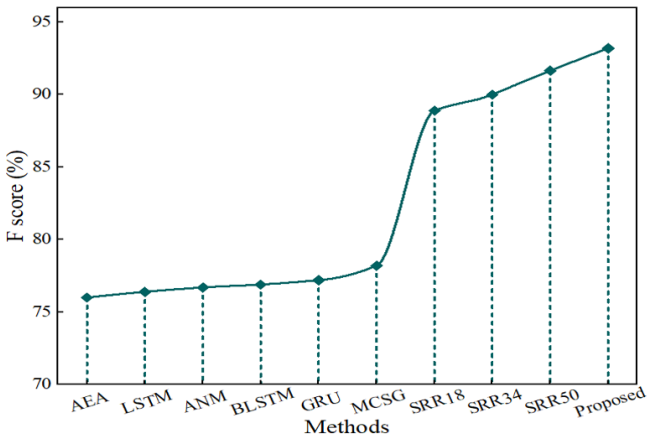


Figure 9. F-score comparison.

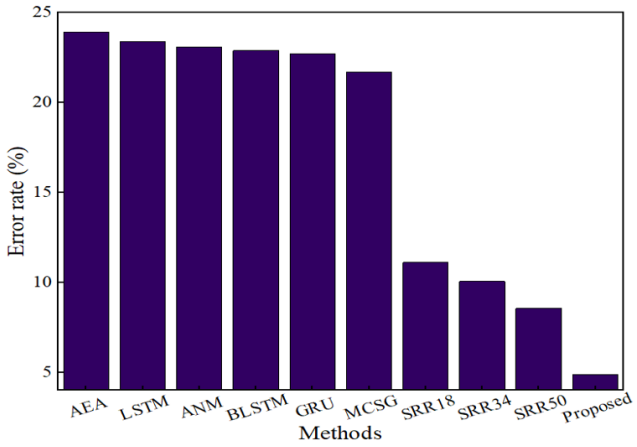


Figure 10. Error rate comparison.

which enables efficient global exploration and refined local exploitation during feature subset selection, in contrast to these classical methods, which frequently suffer from premature convergence, slow exploration–exploitation balance, or high computational overhead in high-dimensional feature spaces. By lowering redundancy and increasing discriminative power, this produces a more ideal and condensed feature set that improves the downstream MLP classifier's learning capacity. Additionally, the suggested methodology firmly integrates POA-driven feature optimization with MLP training, improving convergence stability and classification performance, whereas conventional methods usually treat feature selection and classification as loosely related steps. Consequently, the PMPDF model outperforms current optimization-based feature selection methods in terms of accuracy and resilience, indicating its improved appropriateness for challenging tasks like drone-based crime detection.

3. RESULTS AND DISCUSSIONS

The Windows 10 Python environment is used to validate the PMPDF. To eliminate errors, the crime dataset is gathered and preprocessed. The necessary feature is chosen by Puma optimisation. The crime is predicted and categorized. Table 2 shows the execution parameters for the suggested PFFL.

3.1. Case Study

The proposed PMPDF is validated by using the UCF crime dataset (https://www.kaggle.com/datasets/odins0n/ucf-crime-dataset). The UCF crime dataset is a publicly available, large-scale, realistic video surveillance dataset that captures a broad range of unconstrained criminal and everyday scenarios in Table 3. The UCF crime dataset contains annotated video sequences with multiple categories. The UCF crime dataset is widely employed in various drone-based and intelligent surveillance systems owing to its realistic visual patterns, unconstrained backgrounds, illumination conditions, and intricate human interactions. Moreover, the proposed method uses the same dataset as that used in the [30] drone-based crime detection study for consistency.

Therefore, the use of the UCF Crime dataset provides effective validation of the proposed framework under realistic public surveillance conditions and demonstrates the applicability of the proposed approach to intelligent drone-based crime-monitoring systems. The dataset description is provided in the Table 4.

Since the UCF Crime dataset is video-based by nature, a precise frame extraction method is necessary to transform it into an image dataset that can be used with the suggested model. In order to provide representative coverage of the full sequence while preventing redundancy, individual frames are retrieved from each video using a uniform temporal sampling technique. In particular, frames are sampled at a predetermined period (e.g., 1 frame per second (fps) or every n frames depending on the video frame rate), which helps strike a compromise between minimizing computing overhead and obtaining adequate temporal information. The significance of the retrieved images for crime

detection can also be increased by using key-frame selection algorithms to emphasize frames with notable motion or scene changes. After being tagged according to the annotation of the parent video (normal or anomalous behavior), all extracted frames go through preprocessing procedures like scaling, normalization, and noise reduction. While retaining training efficiency, this methodical extraction procedure guarantees that the spatial features the model learns accurately represent the underlying temporal dynamics of the original video data.

Figure 3 shows the performance development of the PMPDF using the accuracy and loss curves over 25 epochs during the training phase. From the accuracy graph, it can be seen that both the training and validation dataset accuracies increase steadily, reaching a maximum around the 6th epoch. This shows that the model is improving in its prediction capacity. Moreover, the loss graph shows a minimum value, indicating the model's effectiveness. Table 5 shows the sample from the image for each processing phase, based on the data. It displays the step-by-step process, including the input image, the preprocessed image, and the analyzed feature image.

The confusion matrix in Figure 4 displays the model's dependability, clearly indicating its ability to differentiate between 14 crime classes. The model's performance clearly demonstrates its effectiveness in achieving the most critical levels of accuracy and dependability for intelligent crime detection. The suggested PMPDF model has great classification performance with high true positive and true negative rates, as demonstrated by the confusion matrix and training curves. Nevertheless, a closer examination of misclassification examples offers further insights into model behavior. The majority of misclassifications take place in situations where it is difficult to discriminate between typical and abnormal events, such as low-resolution frames, motion blur, visually identical activities, or bad illumination. For example, subtle or partially obscured crime scenes may result in false negatives, while some non-criminal behaviors with sudden movements may be mistakenly identified as suspicious (false positives). These mistakes demonstrate the drawbacks of depending only on spatial characteristics that are taken from individual frames. However, the lower error rate of the suggested model indicates that there are still few misclassifications overall. Such inaccuracies could be further reduced by improving feature representation or using temporal information from subsequent frames. This investigation shows that although the PMPDF model is quite successful, resilience and reliability in real-world deployments can be further enhanced by comprehending and resolving misclassification instances.

On the other hand, Figure 5 depicts the comparative performance analysis of the Standard MLP and the proposed PMPDF, which leverages the POA for feature selection. The proposed PMPDF approach strikes a harmonious balance between the problem dimensions and accuracy, achieving 94.59% accuracy and an F-score of

Table 6. Overall performance comparison.

Methods	Accuracy	Recall	Precision	F score	Error rate	CI(±)	SD	p-Value
AEA	76.1	76.1	76.4	76.0	23.9	2.10	1.92	0.021
LSTM	76.6	76.6	77.0	76.4	23.4	2.05	1.88	0.018
ANM	76.9	76.9	77.1	76.7	23.1	2.01	1.85	0.015
BLSTM	77.1	77.1	77.5	76.9	22.9	1.96	1.79	0.013
GRU	77.3	77.3	77.6	77.2	22.7	1.92	1.73	0.011
MCSG	78.3	78.3	78.5	78.2	21.7	1.74	1.58	0.009
SRR18	88.89	88.1	89.69	88.89	11.11	1.24	1.12	0.006
SRR34	89.94	89.54	90.46	90.00	10.06	1.18	1.03	0.004
SRR 50	91.44	91.57	91.71	91.64	8.56	1.02	0.91	0.002
Proposed	94.59	94.59	91.9	92.46	5.41	0.74	0.62	0.0005

Table 7. PMPDF Performance with statistical measures.

Metrics	Without optimization (baseline model)	With optimization (proposed)	mean	P-Value	95%CI (±)	Improvement Vs Baseline (%)
Accuracy	92.74	94.59%	94.31	0.003	±0.82	1.99
Precision	89.49	91.90%	91.62	0.005	±0.95	2.69
Recall	92.74	94.59%	94.27	0.004	±0.88	1.99
F score	90.89	92.46%	92.18	0.006	±0.91	1.73
Error rate	7.26	5.41%	5.63	0.002	±0.74	25.48

92.46%, outperforming the MLP approach's 92.74% accuracy and 90.89% F-score. Also, the precision value shows a significant improvement from 89.49% to 91.9%, with a minimization of the error rate from 7.26% to 5.41%.

3.2. Performance Analysis

The PMDPF is validated with a few performance metrics, such as recall, precision, F-score, accuracy, and error rate, and compared with existing approaches such as Autoencoder algorithm (AEA), Long short-term memory (LSTM), Alex Net model (ANM), Bidirectional LSTM (BLSTM), Gated Recurrent Unit (GRU), Multiscale Contextual Semantic Guidance (MCSG) [30], Simple Recurrent ResNet 18 (SRR18), SRR34, and SRR 50 [31].

3.2.1. Accuracy

Accuracy calculates the overall correctness of the model classification. The proportion of correctly classified instances relative to the total number of samples is the accuracy. It reflects the effectiveness of the proposed system, which identifies both crime and non-crime activities. It is expressed by Equation (9).

$$Accuracy = \frac{CP_{crime} + CP_{non-crime}}{CP_{crime} + CP_{non-crime} + IP_{crime} + IP_{non-crime}} \quad \textbf{(9)}$$

Here, CP_{crime} means the crime cases that were correctly detected; $CP_{non\text{-}crime}$ means normal cases that were correctly identified. IP_{crime}, which are the normal cases that were misclassified as crime; and $IP_{non\text{-}crime}$, representing crime cases are incorrectly classified as normal. Figure 6 displays the accuracy comparison. The existing AEA, LSTM, ANM, BLSTM, GRU, MCSG, SRR18, SRR34, and SRR50 models achieved accuracy rates of 76.1%, 76.6%, 76.9%, 77.1%, 77.3%, 78.3%, 88.89%, 89.94%, and 91.44%, respectively. Compared to this, the proposed model achieved an accuracy of 94.59%.

3.2.2. Precision

Precision measures the quality of the model's positive predictions. It helps determine how many of the identified instances are criminal activities. High precision indicates fewer false alarms, which is significant in crime detection because it provides

useful indications. Precision is given by Equation (10).

$$Precision = \frac{CP_{crime}}{CP_{crime} + IP_{crime}} \quad \textbf{(10)}$$

The greater the precision value, the fewer false crime alarms the proposed PMPDF framework generates. The precision comparison is shown in Figure 7. The precision rate attained by AEA is 76.4%, LSTM is 77%, ANM is 77.1%, BLSTM is 77.5%, GRU is 77.6%, MCSG is 78.5%, SRR18 is 89.69%, SRR34 is 90.46%, SRR50 is 91.71%, and the proposed model obtained a high precision rate of 91.9%.

3.2.3. Recall

Recall, also known as sensitivity. It is a measure of how well the model identifies all instances of crime. It is a metric used to assess the system's effectiveness in identifying crime incidents. High values of the recall metric are of utmost importance for use in the field of security. The formula for the recall metric is expressed by Equation (11).

$$Recall = \frac{CP_{crime}}{CP_{crime} + IP_{non-crime}} \quad \textbf{(11)}$$

The higher the recall value, the better the model minimizes false negatives. The recall comparison is displayed in Figure 8. The existing AEA, LSTM, ANM, BLSTM, GRU, MCSG, SRR18, SRR34, and SRR50 models obtained recall rates of 76.1%, 76.6%, 76.9%, 77.1%, 77.3%, 78.3%, 88.1%, 89.54%, and 91.57%, respectively. The proposed model attained a high recall rate of 94.59%.

3.2.4. F Score

The F1-score is a measure that provides a balanced evaluation of the model, combining precision and recall into a single value. It is useful when classes are imbalanced, as it accounts for both false positives and false negatives. The F1-score is defined as the harmonic mean of precision and recall, described in Equation (12).

$$F1-score = 2 \times \frac{A \times B}{A + B} \quad \textbf{(12)}$$

The precision value is denoted as A and the recall value is denoted as B. A higher F-score indicates a more robust and reliable crime-detection model, given the balance of performance. The F score attained by AEA is 76%, LSTM is 76.4%, ANM is 76.7%, BLSTM is 76.9%, GRU is 77.2%, MCSG is 78.2%, SRR18 is 88.89%, SRR34 is 90%, SRR50 is 91.64%, and the proposed model obtained a high F score rate of 92.46% is displayed in Figure 9.

3.2.5. Error Rate

The error rate is the proportion of incorrectly classified instances in the dataset. It means how often the model predicts wrong. This is a complementary measure to accuracy. A lower error rate indicates better classification performance. The error rate is defined as Equation (13).

$$Error\ rate = \frac{IP_{crime} + IP_{non-crime}}{CP_{crime} + CP_{non-crime} + IP_{crime} + IP_{non-crime}} \quad \textbf{(13)}$$

The error rate should be minimized to develop the reliability of the proposed PMPDF framework in real-world drone-based crime detection. The error rate comparison is shown in Figure 10. The error rate obtained by AEA is 23.9%, LSTM is 23.4%, ANM is 23.1%, BLSTM is 22.9%, GRU is 22.7%, MCSG is 21.7%, SRR18 is 11.11%, SRR34 is 10.06%, SRR50 is 8.56%, and the proposed model obtained low error rate of 5.41%. The entire comparison is depicted in the Table 6.

3.3. Discussion

The experimental results show that the proposed PMPDF outperforms existing methods in crime detection. The contribution of POA in feature analysis is significant for improving image classification accuracy by focusing solely on the most crucial image features, without incorporating redundant or unnecessary information. Moreover, preprocessing in image analysis removes noise, which is equally essential for ensuring accurate crime identification. Increased accuracy in both precision and recall demonstrates the proposed model's ability to handle both false alarms and missed alarms, which is very important in security-sensitive application domains such as crime identification. The PMPDF performance is described in Table 7 with the statistical measures. The resilience and efficacy of the suggested PMPDF model for intelligent crime detection are confirmed by the fact that it continuously

Table 8. Performance comparison with extended metrics and fairness analysis.

Model / Ref	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)	Error Rate (%)	ROC-AUC	Latency (ms)	Parameter Tuning	Fairness Condition
Hybrid ML + IoT + XAI [32]	91.30	88.90	91.20	89.30	8.70	0.92	45	Tuned	Same dataset & preprocessing
CNN + XAI [33]	91.50	89.00	91.30	89.50	8.50	0.93	60	Tuned	Same dataset & preprocessing
ANN Model [34]	89.20	87.80	88.60	88.10	10.80	0.90	35	Limited	Same dataset & preprocessing
Hybrid ML [35]	91.40	88.80	91.20	89.40	8.60	0.92	48	Tuned	Same dataset & preprocessing
ML Models [36]	88.90	87.60	88.40	88.00	11.10	0.89	50	Tuned	Same dataset & preprocessing
Statistical [37]	87.50	86.80	87.20	87.00	12.50	0.87	30	Not applicable	Same dataset assumption
ResNet-50	91.40	88.80	91.30	89.20	8.60	0.93	65	Tuned	Same dataset & preprocessing
YOLOv8	91.50	88.90	91.40	89.30	8.50	0.94	70	Tuned	Same dataset & preprocessing
VGG-16	90.80	88.20	90.70	88.90	9.20	0.91	68	Tuned	Same dataset & preprocessing
EfficientNet-B0	91.30	88.80	91.20	89.10	8.70	0.93	55	Tuned	Same dataset & preprocessing
CNN-LSTM	91.20	88.70	91.10	89.00	8.80	0.92	62	Tuned	Same dataset & preprocessing
Baseline Model	**91.30**	**88.70**	**91.40**	**89.30**	**8.60**	**0.91**	**50**	Same as proposed	Same dataset & preprocessing
Proposed (PMPDF)	**94.59**	**91.90**	**94.59**	**92.46**	**5.41**	**0.96**	**28**	Optimized (POA)	Same dataset & preprocessing

outperforms all current and new models by at least 3% across all assessment criteria and achieves the lowest error rate in Table 8.

3.3.1. Practical Implication

The proposed PMPDF framework has a range of practical implications for real-world crime surveillance and public security systems. Leveraging capabilities in image data acquisition and optimized intelligent features, this framework can be applied in smart cities, critical infrastructure surveillance, and event surveillance to effectively identify potential crime events. The optimized selection of features in this framework reduces computational complexity and makes it suitable for real-time systems with constrained capabilities. Additionally, because the MLP is dynamic, this framework can be updated with new datasets to improve over time based on evolving crime patterns.

Deploying the suggested PMPDF system on drone platforms with limited resources necessitates careful consideration of computational, memory, and energy constraints, as the real-time implementation discussion makes clear. Lightweight model design and optimal feature selection utilizing the POA are essential for minimizing computational overhead because drones usually have limited onboard computing power and battery capacity. The suggested framework can be implemented utilizing edge-computing techniques to guarantee viability, whereby more complex processing can be offloaded to a ground station or cloud when needed, while inference is carried out locally on the drone for quick reaction. Furthermore, model compression methods like quantization and pruning can further minimize memory use without appreciably compromising accuracy. During continuous video analysis, effective frame processing techniques like selective frame sampling also aid in reducing energy usage. All things considered, these factors guarantee that the PMPDF model maintains high accuracy and low latency performance while being feasible for real-time drone-based crime detection.

3.3.2. Significance and Real-time Challenges

The problems exist between cybercrime detection conventional models has been filled by using of drones based intelligence system. By combining drone-acquired location, environmental, and network-related information with the latest advances in deep learning, the proposed model is capable of context-aware and proactive cybercrime detection in public spaces. However, real-time implementation of this framework is faced with a number of challenges, such as the capability to process high-volume streaming network traffic, achieving low latency processing on resource-limited drone platforms and maintaining reliable communication channels. Furthermore, scaling the proposed solution to adapt with varying public space environments and dynamic cyber threats also demands continuous learning and system scalability, making real-time implementation of this approach both technically challenging and important.

3.3.3. Computational Complexity

The total computational complexity of the proposed PMPDF is dominated by three main components: data preprocessing and feature handling, Puma-based feature optimization, and Multilayer Perceptron (MLP) learning and prediction. Let D_n denote the total number of data samples collected from drone-assisted monitoring, D represent the original feature dimension of the cybercrime dataset, F indicate the number of features selected by the POA, and r_1 *and* r_2 denote the number of random optimization iterations. The preprocessing and feature normalization step has a time complexity of $O(FD)$. The POA involves iterative fitness evaluation and feature subset selection, leading to a computational complexity. Following feature selection, the MLP training and classification step proceeds with a lower complexity of $O(F \times D \times h)$ where h is the number of hidden neurons. In contrast to traditional deep learning methods that directly handle all features with higher computational complexity, the proposed PMPDF framework is computationally more efficient and scalable for real-time cybercrime detection in drone -assisted public settings.

4. CONCLUSIONS

This work introduced an intelligent drone-assisted crime detection framework using a novel

PMPDF approach. The proposed system effectively integrated image preprocessing, intelligent feature analysis based on the POA, and accurate prediction by an MLP. Selection of the most relevant features and reduction of feature redundancy improve the classifier's performance while reducing the computational complexity of the framework. Experimental validation using standard performance metrics has also been presented to demonstrate that the proposed PMDPF model outperforms existing approaches in terms of accuracy and reliability. The model achieved 94.59% accuracy, 91.9% precision, 94.59% recall, 92.46% F-score, and 5.41% error rate. In addition, the results confirm the suitability of the proposed approach for real-world crime detection in public environments. Although the suggested PMPDF framework achieves great accuracy and efficiency, it should be noted that it has some limitations. The quality of the input data determines how well the model performs, and difficult circumstances like low lighting, occlusions, and motion blur may have an impact. Furthermore, deployment on drones with limited resources may be impacted by the POA's additional processing burden. Additionally, the approach requires labeled data, which may limit scalability, and focuses mostly on spatial features from individual frames, failing to completely capture temporal information in video sequences. Future research can improve performance, efficiency, and practicality by investigating semi-supervised learning, utilizing model compression approaches for edge deployment, and integrating spatiotemporal models.

AUTHOR INFORMATION

Corresponding Author

Vidyarani Hamppayanamalige Jayaprakash — Department of Computer Science and Business System, Dr. Ambedkar Institute of Technology, Karnataka-560056 (India);
orcid.org/0000-0001-8566-5084
Email: vidyarani.is@drait.edu.in

Authors

Girija Sanjeevaiah — Department of Information Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka-560056 (India);
orcid.org/0000-0002-4267-5466

Mohankumar Venugopal — Department of Artificial Intelligence and Machine Learning, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka-560056 (India);
orcid.org/0000-0001-9569-6888

Ganesh Hamppatanamalige Jayaprakash — Department of Artificial Intelligence and Machine Learning, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka-560056 (India);
orcid.org/0009-0006-0945-2064

Nishchitha Malligere Harendra Kumar — Department of Robotics and Artificial Intelligence, Dayananda Sagar College of Engineering, Bengaluru, Karnataka-560111 (India);
orcid.org/0009-0005-0568-3220

Author Contributions

V. H. J.: Conceptualization; G. S.: Methodology; M. V.: Validation; G. H. J.: Formal Analysis; N. M. H.: Investigation.

Conflicts of Interest

The authors declare no conflict of interest.

DECLARATION OF GENERATIVE AI

Not applicable.

REFERENCES

[1] S. C. Hassler, U. A. Mughal, and M. Ismail. (2023). "Cyber-physical intrusion detection system for unmanned aerial vehicles". *IEEE Transactions on Intelligent Transportation Systems*. **25** (6): 6106-6117. 10.1109/TITS.2023.3339728

[2] N. Alturki, T. Aljrees, M. Umer, A. Ishaq, S. Alsubai, O. Saidani, S. Djuraev, and I. Ashraf. (2023). "An intelligent framework for cyber-physical satellite system and iot-aided aerial vehicle security threat detection". *Sensors*. **23** (16): 7154. 10.3390/s23167154

[3] A. Vajravelu, N. Ashok Kumar, S. Sarkar, and S. Degadwala. (2023). "Security threats of unmanned aerial vehicles". Wireless

Networks: Cyber Security Threats and Countermeasures. Cham: Springer International Publishing. 133-164. 10.1007/978-3-031-33631-7_5

[4] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan. (2023). "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends". *Intelligent Service Robotics*. **16** (1): 109-137. 10.1007/s11370-022-00452-4

[5] E. Torba, and H. Jahankhani. (2025). "Securing systems from aerial threats: cybersecurity in the drone era". Autonomous Revolution: Strategies, Threats and Challenges. Cham: Springer Nature Switzerland. 261-306. 10.1007/978-3-031-96782-5_9

[6] N. d'Ambrosio, , G. Perrone, S. P. Romano, and A. Urraro. (2025). "A cyber-resilient open architecture for drone control". *Computers & Security*. **150** : 104205. 10.1016/j.cose.2024.104205

[7] I. Anagnostis, P. Kotzanikolaou, and C. Douligeris. (2025). "Understanding and securing the risks of unmanned aerial vehicle services". *IEEE Access*. **13** : 47955-47995. 10.1109/ACCESS.2025.3549861

[8] S. N. Ashraf, , S. Manickam, S. S. Zia, A. A. Abro, M. Obaidat, M. Uddin, M. Abdelhaq, and R. Alsaqour. (2023). "IoT empowered smart cybersecurity framework for intrusion detection in internet of drones". *Scientific Reports*. **13** (1): 18422. 10.1038/s41598-023-45065-8

[9] A. Alzahrani. (2024). "Novel approach for intrusion detection attacks on small drones using ConvLSTM model". *IEEE Access*. **12** : 149238-149253. 10.1109/ACCESS.2024.3471806

[10] M. S. Bhargavi, P. Hemavathi, C. Jaganath, and K. Narendra. (2025). "Unlocking the Potential of Machine Learning for Enhanced Security in Drone-Enabled IoT Networks". Machine Learning for Drone-Enabled IoT Networks: Opportunities, Developments, and Trends. Cham: Springer Nature Switzerland. 107-120. 10.1007/978-3-031-80961-3_6

[11] M. Nankya, R. Chataut, and R. Akl. (2023). "Securing industrial control systems: Components, cyber threats, and machine learning-driven defense strategies". **Sensors**. **23** (21): 8840. 10.3390/s23218840

[12] R. Morshedi, and S. Mojtaba Matinkhah. (2025). "Cybersecurity Challenges and Solutions in Unmanned Aerial Vehicles (UAVs)". *Journal of Field Robotics*. **43** (1): 314-329. 10.1002/rob.70040

[13] D. A. H. Ollachica, B. K. A. Asante, and H. Imamura. (2024). "Autonomous UAV implementation for facial recognition and tracking in GPS-denied environments". *IEEE Access*. **12** : 119464-119487. 10.1109/ACCESS.2024.3447899

[14] D. Alsadie. (2025). "Cybersecurity and Artificial Intelligence in Unmanned Aerial Vehicles: Emerging Challenges and Advanced Countermeasures". *IET Information Security*. **2025** (1): 2046868. 10.1049/ise2/2046868

[15] R. A. AL-Syouf, R. M. Bani-Hani, and O. Y. AL-Jarrah. (2024). "Machine learning approaches to intrusion detection in unmanned aerial vehicles (UAVs)". *Neural Computing and Applications*. **36** (29): 18009-18041. 10.1007/s00521-024-10306-y

[16] H. J. Hadi, Y. Cao, M. K. Khan, N. Ahmad, Y. Hu, and C. Fu. (2025). "UAV-NIDD: A Dynamic Dataset for Cybersecurity and Intrusion Detection in UAV Networks". *IEEE Transactions on Network Science and Engineering*. **12** (4): 2739-2757. 10.1109/TNSE.2025.3553442

[17] E. Ntizikira, W. Lei, F. Alblehai, K. Saleem, and M. A. Lodhi. (2023). "Secure and privacy-preserving intrusion detection and prevention in the internet of unmanned aerial vehicles". *Sensors*. **23** (19): 8077. 10.3390/s23198077

[18] F. Tlili, S. Ayed, and L. C. Fourati. (2024). "Exhaustive distributed intrusion detection system for UAVs attacks detection and security enforcement (E-DIDS)". *Computers & Security*. **142** : 103878. 10.1016/j.cose.2024.103878

[19] R. Fu, X. Ren, Y. Li, Y. Wu, H. Sun, and M. A. Al-Absi. (2023). "Machine-learning-based UAV-assisted agricultural information

security architecture and intrusion detection". *IEEE Internet of Things Journal*. **10** (21): 18589-18598. 10.1109/JIOT.2023.3236322

[20] Y. Xu, Y. Liu, H. Li, L. Wang, and J. Ai. (2024). "A deep learning approach of intrusion detection and tracking with UAV-based 360 camera and 3-axis gimbal". *Drones*. **8** (2): 68. 10.3390/drones8020068

[21] T. Wisanwanichthan, and M. Thammawichai. (2025). "A lightweight intrusion detection system for iot and UAV using deep neural networks with knowledge distillation". *Computers*. **14** (7): 291. 10.3390/computers14070291

[22] A. S. Nair, and S. M. Thampi. (2025). "SoCoMNNet: A SocioCognitive and Memristive Neural Network-Based Context-Aware GPS Spoofing Detection and Mitigation in the Internet of Drones". *Vehicular Communications*. **56** : 100980. 10.1016/j.vehcom.2025.100980

[23] D. S. Prasad, P. Jyothi, G. Suryanarayana, and S. N. Mohanty. (2023). "Algorithms to Mitigate Cyber Security Threats by Employing Intelligent Machine Learning Models in the Design of IoT-Aided Drone". Drone Technology: Future Trends and Practical Applications. 257-300. 10.1002/9781394168002.ch11

[24] L. R. Vuyyuru, N. M. R. Purimetla, K. Y. Reddy, S. S. Vellela, S. K. Basha, and R. Vatambeti. (2025). "Advancing automated street crime detection: a drone-based system integrating CNN models and enhanced feature selection techniques". *International Journal of Machine Learning and Cybernetics*. **16** (2): 959-981. 10.1007/s13042-024-02315-z

[25] F. S. Alsubaei, A. A. Almazroi, and N. Ayub. (2024). "Enhancing phishing detection: A novel hybrid deep learning framework for cybercrime forensics". *IEEE Access*. **12** : 8373-8389. 10.1109/ACCESS.2024.3351946

[26] M. M. Mukto, M. Hasan, M. M. Al Mahmud, I. Haque, A. Ahmed, T. Jabid, S. Ali, M. R. A. Rashid, M. M. Islam, and M. Islam. (2024). "Design of a real-time crime monitoring system using deep learning techniques". *Intelligent Systems with Applications*. **21** : 200311. 10.1016/j.iswa.2023.200311

[27] M. Y. Alzahrani. (2024). "Enhancing drone security through multi-sensor anomaly detection and machine learning". *SN Computer Science*. **5** (5): 651. 10.1007/s42979-024-02983-2

[28] A. Alsumayt, N. Nagy, S. Alsharyofi, N. Al Ibrahim, R. Al-Rabie, R. Alahmadi, R. A. Alesse, A. A. Alahmadi. (2024). "Detecting Denial of Service Attacks (DoS) over the Internet of Drones (IoD) Based on Machine Learning". *Sci*. **6** (3): 56. 10.3390/sci6030056

[29] B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, P. Trojovský, F. S. Gharehchopogh, E. S. M. El-kenawy, L. Abualigah, and S. Mirjalili. (2024). "Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning". *Cluster Computing*. **27** (4): 5235-83. 10.1007/s10586-023-04221-5

[30] L. R. Vuyyuru, N. M. R. Purimetla, K. Y. Reddy, S. S. Vellela, S. K. Basha, and R. Vatambeti. (2025). "Advancing automated street crime detection: a drone-based system integrating CNN models and enhanced feature selection techniques". *International Journal of Machine Learning and Cybernetics*. **16** (2): 959-81. 10.1007/s13042-024-02315-z

[31] M. Qasim, and E. Verdu. (2023). "Video anomaly detection system using deep convolutional and recurrent models". *Results in Engineering*. **18** : 101026. 10.1016/j.rineng.2023.101026

[32] E. Aslan, Y. Ozupak, F. Alpsalaz, and Z. M. S. Elbarbary. (2025). "A hybrid machine learning approach for predicting power transformer failures using internet of things based monitoring and explainable artificial intelligence". *IEEE Access*. **13** : 113618-113633. 10.1109/ACCESS.2025.3583773

[33] F. Alpsalaz, Y. Özüpak, E. Aslan, and H. Uzel. (2025). "Classification of maize leaf diseases with deep learning: Performance evaluation of the proposed model and use of explicable artificial intelligence". *Chemometrics and Intelligent Laboratory Systems*. **262** : 105412. 10.1016/

j.chemolab.2025.105412

[34] Y. Oezuepak, and E. Aslan. (2024). "Using artificial neural networks to improve the efficiency of transformers used in wireless power transmission systems for different coil positions". *Revue Roumaine des Sciences Techniques-Série Électrotechnique et Énergétique*. **69** (2): 195-200. 10.59277/RRST-EE.2024.2.13

[35] F. Alpsalaz, Y. Özüpak, E. Aslan, and H. Uzel. (2026). "Hybrid Machine Learning Approach for Enhanced Fault Detection and Power Estimation in Photovoltaic Systems". *IET Renewable Power Generation*. **20** (1): e70153. 10.1049/rpg2.70153

[36] E. Aslan, and Y. Özüpak. (2025). "Comparison of machine learning algorithms for automatic prediction of Alzheimer disease". *Journal of the Chinese Medical Association*. **88** (2): 98-107. 10.1097/JCMA.0000000000001188

[37] N. Gelirli, and H. Tangül. (2025). "A Study on Analyzing the Level of Public Compliance with Drone Laws in Northern Cyprus". *Journal of Aviation*. **9** (1): 53-63. 10.30518/jav.1543124