Intelligent Detection Of Crime Anomalies In Smart Cities Using Hybrid MachineLearning With Improved Segmentation And Feature Extraction Techniques

Ayush Singhal [1] Nidhi Tyagi[2]

[1] Research Scholar, Department of Computer Science & Engineering, Shobhit Institute of Engineering & Technology, (NAAC Accredited Grade "A", Deemed to- be- University), Meerut (250110), India
[2] Professor , School of Computational Sciences and Engineering Shobhit Institute of Engineering and Technology

Abstract

The rise in population in urban areas has resulted in difficulties in policing and monitoring high-crime probability areas, leading to an increase in criminal activity and insecurity. To enhance security, smart cities have integrated crime detection systems with videosurveillance as the standard method. The backlog of video data that must be monitored by supervising officials can lead to an increase in error rates. To address this issue, a proposed solution involves using meta-heuristic optimization with a Hybrid Machine Learning algorithm. This solution analyzes video stream data quickly and accurately, facilitating the identification of criminal activity. This approach is expected to improve the efficiency and effectiveness of video surveillance systems. The proposed method involves pre-processing the video data using techniques such as Video-to-Frame Conversion, Resizing, and Normalization, followed by segmentation of the frames using an optimized Semantic Segmentation-Optimized FCN algorithm. Features are then extracted from the segmented regions using techniques such as SIFT and the proposed Improved Histogram of Oriented Gradients algorithm. The extracted features are refined using the new improved Relief Algorithm for feature selection. Lastly, a new hybrid machine learning approach is designed using a combination of transformer model, SVM, and ANN for crime anomaly detection. The proposedmethod is implemented using the Python programming language.

Keywords: Support Vector Machine, Artificial Neural Network, FCN,SIFT, Crime detection, Improved Histogram of Oriented Gradients, Relief Algorithm.

1. INTRODUCTION

Technologies for smart cities (SC) can deliver the right services to the needs of the populace. One of the primary enablers in a SC has IoT technology, that enables a large numberof devices to connect [1]. However, because of the varied form and complexity of anomalous events, recognising them automatically in a real-world situation is extremely challenging. Thisresearch effort presents an effective and robust approach for identifying anomalies in surveillance large video data using Artificial Intelligence of Things [2]. A critical and difficultproblem in IoT systems is anomaly detection because of the intricate structures and high- dimensional data they generate [3]. Anomalies are described as data structures that do not adhere to well-defined characteristics of typical data patterns. “An observation which deviates

so much from the other findings as to arouse doubts that it was generated by a different mechanism is the definition of an anomaly.[4][5]. To detect anomalous activities [6], It isnecessary to develop a computer vision-based technique that can effectively classify normaland abnormal events without the involvement of humans. In addition to being useful for monitoring, such an automated approach also minimizes the amount of human labour needed to maintain manual observation on a round-the-clock basis [7].

IoT devices are widely used in smart cities due to the IoT's recent growth [8]. Real-world timeis used as the basis activities of a smart city are designed to enhances the efficiency and qualityof life in urban areas. Since the smart city network traffic through the IoT system is linked to sensors that are directly connected to huge cloud servers is growing quickly and posing new challenges for cyber-security [9]. Physical and/or cyberattacks are the two main types of attackson smart cities. In a physical attack, the attacker is in close proximity to the IoT sensors and can easily modify them or tamper with the communication-related IoT devices and sensors. This attack includes permanent denial of service, malicious code injection, & fake node injection. Obtaining unauthorised access to smart city network components, the adversary in cyberattacks tries to introduce malware or other harmful software [10]. IoT devices frequentlycommunicate with one another to guarantee good QoS in an IoT environment for smart cities. Continuous communication between various IoT devices within the IoT environment presents serious security risks, including DoS attacks, malicious scans, malicious controls, malicious operations, data probing, spying and incorrect set. These anomalies have the potential to causeserious risks at any time and disrupt communication as usual. Because of these potential risks,IoT communication remains in an unsafe state. To ensure good QoS in smart cities, it is crucialto monitor and identify anomalies [11][12]. The most recent advances in AI, ML, and DRL- based smart city applications. The research focused on the function and influence of previous

initiatives on the most important components of smart cities [13]. To address these issues, researchers provide a standard-based process for creating and implementing Fog Flow, a novelfog computing-based framework for IoT smart city platforms [14]. With the development of smart technologies, the government has began constructing smart cities with the use of IoT devices, to record the vulnerable activities occurring in and around the surroundings in order to decrease the crime rate [15].

This research extensively discusses a proposed method and its practical implementation, providing a detailed analysis of its functioning. Section 2 presents the latest research linked to the approach, while Section 3 elaborates on the proposed methodology. TheSection 4 summaries the results and discussion of the technique, and Section 5 finishes the research.

2. Literature Review

In 2021, Ahmad and Zhang et al. [16] have proposed IoT implementation in energy business models can lead to increased efficiency, intelligence, and connectivity. It can support integratedsolutions for mitigating environmental impact and regulating energy usage. IoT is also Increasing operational efficiency and process automation in areas such as retail, electricity, manufacturing, and logistics. Using IoT in the Energy Sector was a promising step toward building sustainable and efficient energy systems.

In 2015, Serrano et al.[17] have presented a new strategy on base of the LOF density-based clustering algorithm was proposed for detecting attacks. The researchers found that theapproach was effective in determining threats while decreasing false positives and that not all counters were useful in identifying malware. The study highlights the potential of using clustering algorithms in intrusion detection systems, improving their effectiveness in identifying and preventing cyber-attacks.

In 2020, Alahakoon and Nawaratne et al. [13] presented Self-building AI, a potential solutionto overcome major constraints in processing, integrating, and analyzing large amounts of datafrom several sources. Big Data's huge volume, variety, and volatility require technology that can self-adapt, as human involvement becomes impractical. This technology has the potential to revolutionize data analysis and make it more efficient and effective.

In 2018, Mohammadi and Al-Fuqaha et al. [18] have introduced a hierarchical architecture, incorporating “machine learning techniques” to manage big data in smart cities. A semi- supervised “deep reinforcement learning framework” was developed to address related challenges, and its potential was demonstrated across multiple smart city applications. This research provides insights into the effective implementation of machine learning techniques inmanaging Smart cities huge data.

In 2020, Sarker et al. [19] have focused on the concept of "Smart City Data Science" which involves data mining from city sensors and associated devices to gain insights and correlations.This enables improved decision-making procedures that are more sophisticated for residents. To do this, several machine learning models can be used to provide a deeper knowledge of municipal data, making the computing process more actionable and intelligent in real-world services that were offered by cities.

In 2022, Qarafi and Alrowais et al. [20] have proposed the OMLIDS-PBIoT system uses machine learning to improve smart city security by detecting and preventing intrusions.. It is particularly useful for privacy-preserving BIoT applications, such as “traffic management, emergency response, healthcare, waste management, air quality monitoring, and disaster management”. This model could play a vital role in maintaining the integrity and safety of smart city systems.

In 2019, Yang and Elisa et al. [22] have proposed to forward a plan to safeguard the privacy and security of e-government systems from potential threats in smart city environments, identifying the challenges in their adoption. Technical solutions were examined to address these issues, and the framework utilizes blockchain and AI breakthroughs to create a decentralized and secure e-government system. This research provides insights for policymakers and stakeholders to improve the security and privacy of e-government systems, measures such as encryption and access controls can be implemented in smart cities.

In 2021, Palanivinayagam and Gopal et al. [21] have used to detect crime, four attribute- generation strategies were employed. By applying K-means clustering, crime hotspots were identified in a specific location based on the distribution of criminal incidents. A crime ratio matrix was then created, which helped predict the probability of crime when fed into a machine-learning model. The dataset was analyzed using these techniques to identify potential crime patterns and prevent future incidents.

In 2021, Sharma and Dhankhar et al. [22] have developed a fog-enabled network's anomaly detection model, allowing for local anomaly identification using fog nodes. This made it easierto locate rogue nodes for future research. The use of statistical techniques for detecting anomalies, particularly DDoS attacks, was explored in subsequent parts. Time-series datasets exhibiting patterns in incoming packets were analyzed using these algorithms.

In 2022, Srihith and Kumar et al. [23] have introduced the importance of ICT, AI, and machinelearning in achieving

smart city goals related to policymaking, decision-making, plan execution, and service delivery. Educational technologies like DRL and ML can assist in developing effective regulations that promote efficiency and sustainability in smart cityoperations.

2.1. Research Gaps

Table 1: Various Authors Review

Author	Aim	Research Gaps
Guo et al. [3]	A framework for classifying and detecting anomalies in multiple IoT scenarios	• The analysis of networks withboth humans andintelligent/social objects is not done.
Tukur et al. [5]	"Edge-Based Blockchain- Enabled Anomaly Detection for Internet of Things Insider Attack Prevention"	• full-fledged anomaly detection is not performed.
Ullah et al. [7]	CNN Features for "Real-Time Anomaly Detection in Surveillance Networks" with Bi-Directional LSTM	• does not address how motion features, robust visual features, and two-stream CNN networks can aid in overcoming lower variation for anomaly recognition.
Kumar et al. [10]	"A Privacy-Preserving and Secure Framework for IoT- Driven Smart Cities Using Blockchain-based Machine Learning"	• Efficiency is less.
Xu et al. [11]	anomaly detection in IoT services utilizes Long Short- Term Memory to improve concept drift adaptation.	• Investigation of the "drift adaptive method" is necessary.

1. Proposed Methodology:

In this proposed methodology for crime anomaly detection, we aim to develop a new hybrid deep learning approach that combines the strengths of Convolutional Neural Networksand Long Short-Term Memory networks for improved performance. The first step in this approach is data pre-processing, which involves converting the video data into individual frames using OpenCV video-to-frame conversion techniques. The frames are then resized to reduce the complexity of the input data using bicubic interpolation. Normalization is applied to the frames using standard deviation normalization to improve the model's accuracy. Next, the Regions of Interest (ROI) in the frames are identified using an optimized Semantic Segmentation- Optimized FCN algorithm. The optimized FCN algorithm segments the framesinto meaningful regions, making it easier to extract relevant features. Features are thenextracted from the segmented regions using two different techniques: Improved Histogram of Oriented Gradients and Scale-Invariant Feature Transform. The SIFT algorithm extractsfeatures such as edges, corners, and blobs from the ROI, while the HOG algorithm extracts feature such as object shape and orientation. To remove any irrelevant or redundant features,the extracted features are refined using a new improved Relief Algorithm. The final step is the crime anomaly detection process. In this proposed methodology, a hybrid deep learning approach is proposed, which integrates CNNs and LSTM networks for better anomaly detection. CNNs extract spatial features from ROIs, while LSTM networks capture temporal dependencies in the data. The proposed hybrid model outperforms existing models by combining the strengths of CNNs and LSTM networks, resulting in better accuracy and improved anomaly detection capabilities. The model can detect and classify various types of crimes, such as robbery, theft, and assault, with high accuracy, making it an effective tool for crime prevention and surveillance. Figure 1 demonstrates the overall flow diagram of the proposed model.

Pre-processing

- Video-to-Frame Conversion: Convert the video data into individual frames using video- to-frame conversion techniques such as OpenCV.
- Resizing: Resize the frames to reduce the complexity of the input data using imageresizing techniques such as bicubic interpolation.

- Normalization: Normalize the frames to improve the model's accuracy usingnormalization techniques such as standard deviation normalization.

Segmentation

- Identify the Regions of Interest (ROI) in the frames using optimized SemanticSegmentation- Optimized FCN (proposed) to segment the frames into meaningful regions.

Feature Extraction

- Extract features from the segmented regions using various feature extraction techniquessuch as:
- Scale-Invariant Feature Transform (SIFT): Extract features such as edges, corners, andblobs from the ROI using the SIFT algorithm.
- Improved Histogram of Oriented Gradients (HOG) (proposed): Extract features such asobject shape and orientation using the HOG algorithm.

Feature Selection

- Refine the extracted features to remove any irrelevant or redundant features using thenew improved Relief Algorithm

Crime Anomaly Detection

To design a “new hybrid deep learning” approach (proposed) for crime anomaly detection bycombining transformer model and hybrid machine learning with SVM and ANN.

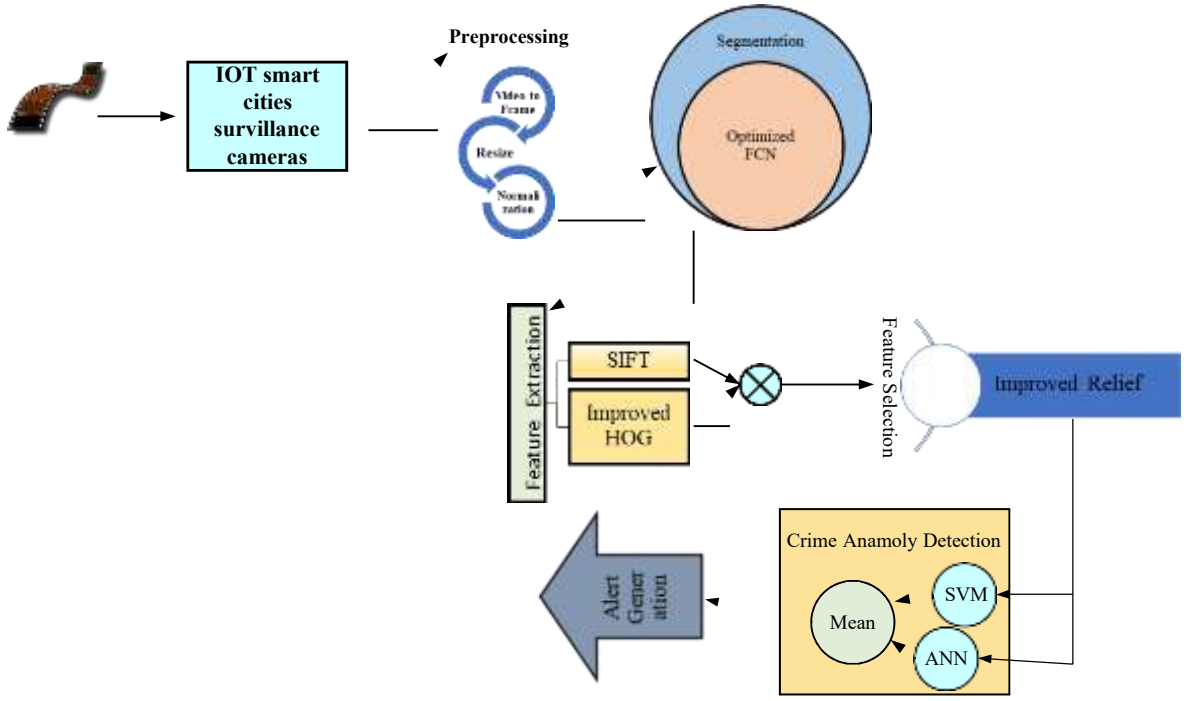


Figure 1: Overall Flow of the proposed model

3.1 Pre-processing

The first step in our proposed methodology involves video pre-processing, which converts it into individual frames using OpenCV. These frames are then resized using bicubic interpolation, and normalization techniques like standard deviation normalization are applied to enhance model accuracy.

3.1.1 Video to frame conversion

Video-to-frame conversion is a common technique used in video processing where a video is broken down into individual frames or images. This process can be used as a pre- processing step before performing various computer vision tasks like object detection, image recognition, or video analysis.to generate unique content from the video frames, you could usetechniques like image augmentation or generative adversarial networks (GANs) to create newvariations of the original frames. Image augmentation involves applying random transformations to the

video frames, such as “rotation, scaling, flipping, and shearing”. By applying data transformations, the model can be made more robust to input variations and enhance the data diversity.

OpenCV is an extensive open-source library that is crucial in modern systems for real-time operations, including computer vision, machine learning, and image processing. Its integration with libraries such as NumPy enables efficient analysis of image and video data. By using vector space, mathematical operations can be performed on image features to identifypatterns and unique characteristics.

3.1.2 Resize

Resizing the frames is a crucial step in our proposed methodology as it reduces the complexity of the input data. By employing image resizing techniques such as bicubicinterpolation, we can adjust the resolution of the frames while maintaining the essential featuresthat are needed for our model. This resizing process allows for a more efficient analysis of theframes, resulting in faster processing times and more accurate results. Additionally, it reducesthe computational burden on the system, making it easier to handle and process large datasets.

3.1.3 Normalization

Performance enhancement often involves utilizing common optimization techniques inthe Normalization of machine learning models, including those used for image processing. Onecommon approach is standard deviation normalization, which involves scaling the pixel valuesof an image, Normalization transforms data to a mean of zero and a standard deviation of one.This helps to ensure that the pixel values are within a comparable range, which can improve the model's image classification and segmentation accuracy is crucial. By normalizing the frames in this way, the proposed model for Alzheimer's disease segmentation and classificationcan be expected to achieve higher accuracy, precision, recall, F1-measure, and other performance metrics, ultimately resulting in a more effective diagnostic tool.

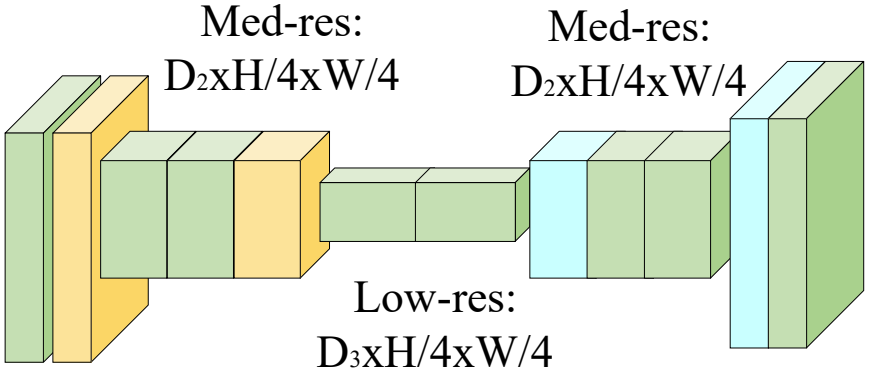
3.2 Segmentation (ROI Identification)

ROI identification, also known as segmentation, is a crucial step in computer vision that involves isolating specific regions within an image or video extracting frames pertinent tothe task is vital. This process helps reduce the amount of irrelevant information and improves the model's efficiency and accuracy are both critical factors. ROI identification can be performed using a variety of techniques, such as thresholding, edge detection, semantic segmentation, or object detection. These methods help to identify and isolate important featuresor objects within an image or video frame, and eliminating noise enhances the model's ability to focus on pertinent data.

In video processing, ROI identification can help to extract relevant information from specific frames or time intervals, which can be useful in applications such as “action recognition, object tracking, and video summarization”. For example, in object tracking, ROI identification can help track the location and movement of specific objects within a video, while in video summarization, it can help to identify key frames that represent the most important aspects of the video. However, it's important to note that ROI identification should be performed carefully and validated to ensure that the identified regions accurately represent the target features. Incorrect identification or mislabeling of regions can lead to inaccurate results and negatively affect the performance of the model. Therefore, it's crucial to use appropriate techniques and carefully validate the results to ensure the accuracy and reliability of the model.

3.2.1 Optimized FCN

Fully Convolutional Networks (FCNs) are utilized for image semantic segmentation, multimodal medical image analysis, and classification and segmentation of high-resolution andmultispectral satellite pictures. Contemporary classification networks (Alex Net, VGGNet, and Google Net) were converted into FCNs, and their learned representations were transferred theytackled the segmentation issue by fine-tuning and devised an innovative architecture that combines deep, coarse layers' semantic data with shallow, fine layers' appearance data.



High-res: D1xH/2xW/2

High-res:
D1xH/2xW

Figure 2: Architecture of FCN with in-network down-sampling and up sampling

FCNs are made up of locally connected convolutional, pooling, and transposition layers. Thisarchitecture employs no dense layer (URL3). Because there are no dense layers, changeable inputs can be fed into the network (URL-3). Figure 2 demonstrates the Architecture of FCN with in-network down sampling and up sampling. An FCN is made up of two parts:

- Downsampling path
- Upsampling path (URL-4)

Extract and evaluate the context via the down-sampling path, as specified in URL-4. Convolutional and max-pooling layers make up the down-sampling process. The upsampling route allows for exact feature localization. The convolutional, convolutional transpose, and concatenate layers make up the upsampling path. For skip connections, concatenation layers are employed. A skip connection is a connection that skips at least one layer. They are frequently employed to transmit local data from the down-sampling path to the upsampling path.

The regularization strength of FCN is optimized via Adam Optimizer to balance the trade-off between training accuracy and generalization performance.

3.2.1.1 Adam optimizer

The Adam optimizer is a popular deep-learning optimization algorithm that combines the advantages of AdaGrad and RMSProp algorithms. Adam optimizer adapts the learning rateper parameter dynamically, based on the gradient of the parameters, and uses a moving averageof the gradients' first and second moments, Adam optimizer provides a precise gradient estimation. Adam is known to converge faster than traditional gradient descent algorithms and is used in various deep learning applications such as “image classification, object detection, and natural language processing”.

The Adam algorithm may exhibit bias towards zero in the initial stages of training due to the initialization of $n1_s$ and $u1_s$ to 0 in equations (1) and (2). Furthermore, if the decay rates β1 and β2 are close to zero, the learning process may slow down.

$$n1_s = \beta_1 \cdot n1_{s-1} + (1 - \beta_1) \cdot h1_s \quad (1)$$

$$u1_s = \beta_2 \cdot u1_{s-1} + (1 - \beta_1) \cdot h1^2_s \quad (2)$$

To mitigate the problem of bias towards zero in Adam's algorithm, the bias-corrected estimates

$n1\hat{}\,s$ and $u1\hat{}s$ are used in weight updates, as demonstrated in equations (3) and (4). This correction helps ensure that the algorithm operates correctly, particularly during the initial stages of training. corrected estimates, the Adam algorithm is better able to ensure that the learning rate adaptsappropriately to the gradient of the parameters during training.

$$\theta_{S+1} = \theta - \frac{y}{\sqrt{\hat{u}_{S+}}}$$
g

3.3. Feature Extraction

The most crucial stage is featuring extraction in many computer vision tasks where informative features are extracted from segmented regions. Techniques like Scale-Invariant Feature Transform and Improved Histogram of Oriented Gradients can be used to extract features such as edges, corners, blobs, object shape, and orientation. These machine-learning models can be trained thru features for tasks like object recognition, detection, and tracking.

3.3.1 Scale Invariant Feature Transform

Scale-Invariant Feature Transform: SIFT is a computer vision method used to discover and describe local features in images. It is a characteristic that is commonly employed in image processing.SIFT operations include "Gaussian (DoG) Space Generation, Key Point Detection, and Feature Description".

3.3.1.1 SIFT features

SIFT is a popular feature detection technique developed by Lowe. It collects robust feature points that are not affected by "image rotation, scaling, restricted affine distortion, lighting change, or projective transformation". The following are the main steps in feature extraction using the SIFT algorithm

Scale-space extrema detection: A SIFT detector is used to detect feature points by searching for local maxima via the difference of Gaussian (DoG) at different scales of the studied image. If we use u(c, b) to represent the input image, the equivalent scale space is specified as:

$$(\mathbf{c}, \mathbf{b}, \sigma) = (\mathbf{c}, \mathbf{b}, \sigma) * \mathbf{u}(\mathbf{c}, \mathbf{b}) \quad (6)$$

3.3.2 Improved HOG (I-HOG algorithm)

The HOG technique is used to extract features from improved histogram-based characteristics. HOG determines the direction and size of a gradient in an image, which is thenused to generate histograms of gradient directions. By adding an adaptive histogram binning mechanism, which boosts the features' discriminative ability, the Improved Histogram-based features (I-HOG) proposed technique improves upon HOG. HOG features are descriptors thatare extensively employed in image processing and computer vision for object detection. These descriptors are justified by the idea that the "distribution of intensity gradients or edge directions" can adequately characterize the appearance and shape of nearby objects. The method counts instances of gradient orientation in specific areas of an image. For greater accuracy, HOG uses overlapping local contrast normalization on a dense grid of cells with equal spacing.

- **Image Preprocessing**

The image is preprocessed by applying various operations to make it suitable for feature extraction. One common operation is grayscale conversion, which is represented as:

$$I_{gray} = 0.2989 * I_{red} + 0.5870 * I_{green} + 0.1140 * I_{blue} \quad (7)$$

where I_{red}, I_{green}, and I_{blue} are the original image's red, green, and blue color channels, and

I_{gray} is the resulting grayscale image.

- **Compute Gradients** In image processing, computing gradients is a crucial step in feature extraction. It involves using a filter like the Sobel filter to determine the gradientmagnitude and direction for each pixel in the image.

$$Gx = I * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (8)$$

$$Gy = I * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (9)$$

where I is the grayscale image, and G_x and G_y are the gradients in the x and y directions,respectively. The gradient's magnitude and direction at each pixel are given by:

$$G_{mag} = sq(G_x2 + G_y2) \quad (10)$$

$$G_{dir} = atan2(G_y, G_x) \quad (11)$$

- **Divide Image into Cells** To extract features from an image, it can be divided into a gridof cells, where each cell has a predetermined number of pixels. The cell size and the number of cells can be modified depending on the needs of the application.
- **Calculate Histograms** To create a feature vector from the gradient information of an image, the histogram of gradient directions is computed for each cell. This involves quantizingthe gradient directions into a fixed number of bins (e.g. 9 bins) and accumulating the gradient magnitudes for each bin. The resulting histogram for each cell is represented as a vectorh_i withseveral elements equal to the number of bins k.

$$h_i = [h_{i1}, h_{i2}, \dots \dots h_{ik}] \quad (12)$$

$$h_{ij} = s(G_{mag}(p, q) if\ bin(G_{dir}(p, q)) == j) \quad (13)$$

where p and q are the pixel coordinates in the cell, $bin(G_{dir}(p, q))$is the bin index corresponding to the gradient direction at (p, q), and h_{ij} is the bin count for the jth bin in the
ith cell.

- **Block Normalization**

To normalize the histograms of neighboring cells in an image, the cells are first grouped into blocks. The histograms within each block are then normalized using a normalization factor that's calculated based on the histograms in the block. This ensures that the feature vector is invariant to changes in lighting and contrast.

$$V = [h_{i1}, h_{i2}, \dots \dots \dots h_{ik}; h_{(i+1)1}, h_{(i+1)2}, \dots h_{(i+1)k}; h_{(i+m-1)1}, h_{(i+m-1)2}, \dots \dots h_{(i+m-1)k}] \quad (14)$$

$$V_{norm} = \frac{V}{\sqrt{s(v)^2+\sigma^2}}$$

where m is the number of cells in the block, σ is an insignificant constant that prevents divisionby zero, and V_{norm} is the normalized histogram vector.
(15

- **Concatenate Histograms** After generating normalized histograms from all cells, theyare combined to form a feature vector for the image, which can be applied to tasks such as object detection, recognition, and tracking. The resulting feature vector is represented as F, with n elements that correspond to the total number of cells in the image. This approach enablesefficient processing of large image datasets and has been widely adopted in computer vision applications.

$$F = [V1; V2; \dots \dots V_n) \quad (16)$$

3.4 "Feature Selection"

The selection of features is an important stage in reducing dimensionality and improving the accuracy of machine learning models. The new improved Relief Algorithm canbe used to refine the extracted features by eliminating irrelevant or redundant features. This

algorithm evaluates each feature's relevance based on its ability to distinguish between classesand selects only the most informative features for further analysis.

3.4.1 Improved Relief Algorithm (Jaccard-Kernel Relief (JKR) algorithm)

The Jaccard distance and Kernel smoothing-based Relief algorithm is an enhanced version of the Relief algorithm that leverages Jaccard distance and kernel smoothing to refine the feature selection process for machine learning applications.

The algorithm works as follows:

Calculate the Jaccard distance between each pair of instances in the dataset. Jaccard distance is a measure of the difference between two sets that can be defined as the ratio of the size of the sets' intersection to the size of their union. The Jaccard distance between two sets A and Bis specified as:

$$(A, B) = |A \cap B|/|A \cup B| \quad (17)$$

where $|A|$ and $|B|$ represent the cardinality (number of elements) of sets A and B, respectively. This formula gives a value between 0 and 1, where 0 indicates that the sets are identical and 1indicates that they have no elements in common. For each instance in the dataset, find the
k nearest neighbors based on Jaccard distance. The k nearest neighbors are the instances withthe smallest Jaccard distance to the current instance. Kernel smoothing is a technique for assigning weights to neighboring instances based on theirsimilarity to the current instance. To calculate the weight of each feature i, we first find the setof k nearest neighbors of x based on Jaccard distance, denoted as (x). Then, for each neighborin (x), we calculate a weight using a kernel function that depends on the Jaccard distance between x and the neighbor. Finally, a weighted average is determined using weights of the feature values for feature i. The weights for different features can vary depending on their relevance to the current instance.

$$w(x) = \left(\frac{1}{\quad}\right) * \sum^{k} K((x,)) * x \quad (18)$$

i $|(x)|$ j=1 j ji

where K is the kernel function, $J(x, x_j)$ is the Jaccard distance between x and x_j, and $x_{ji}$is the value of feature i for instance x_j.

The kernel function K is a function that assigns weights to instances based on their similarity to the current instance. A commonly used kernel function is the Gaussian kernel, which is defined as:

$(u) = \exp(^{-u^2}$ ___) (19)

$2\sigma^2$

where u is the Jaccard distance between two instances, and σ is a parameter that regulates thekernel's width. The Relief algorithm updates the weight of each feature by calculating the difference in feature values between the present instance and its k nearest neighbours. It then uses this difference to distinguish between positive and negative nearest neighbors and computes the average difference for each group. Finally, the weight of each feature i is updated as the difference in average values between positive and negative neighbors for that feature. This process can help identify the most important features for classification or prediction tasks.

$\Delta(x) = \sum^{k}_{j=1} w_i(x) * (x_i - x_{ji})$ (20)

where x_i is the feature value i for the current instance x, and x_{ji} is the value of feature i for the

jth nearest neighbor of x. The positive and negative sets for feature i are defined as:

$(x) = \{x_j | x_{ji} = x_i\}$ (21)

$(x) = \{x_j | x_{ji} \neq x_i\}$ (22) The

weight of feature i is then updated as:

$w\_(x) = w\_i(x) + (1 / k) * (\Sigma\_x\_j \in P\_i(x) \Delta\_i(x\_j) / |P\_i(x)| - \Sigma\_x\_j \in N\_(x) \Delta\_i(x\_j) / |N\_i(x)|)$ (23)

$w(x) = w(x) + (\frac{1}{}) * (\sum \Delta i(xj) - \sum \Delta i(xj)$ (24)

$\overline{i}$ i k $xj \epsilon P_{i(x)} |Pi(x)|$ $xj \epsilon |N_{i(x)} |Ni(x)|$

The Relief algorithm is used to change the weight of feature i by calculating the averagedifference in feature values between positive and negative sets and scaling it by the $|P_i(x)|$ and $|(x)|$cardinalities of the sets.

Steps 2-4 are repeated for each instance to update the weights of features for all instances inthe dataset. After updating the feature weights, features are ranked, and the top features are selected as thefinal feature subset.

The Jaccard distance and Kernel smoothing-based Relief algorithm is an improvement over theoriginal Relief algorithm by incorporating Jaccard distance for categorical data and kernel smoothing for better local structure capture. This leads to more accurate feature weighting andselection.

3.5 Crime Anamoly Detection

A novel hybrid deep learning approach for crime anomaly detection is proposed, whichcombines a transformer model and hybrid machine learning techniques such as support vectormachine and artificial neural network. The proposed approach aims to improve the accuracy ofcrime anomaly detection and can potentially enhance law enforcement efforts.

3.5.1 "Support Vector Machine"

Support Vector Machine is a popular supervised learning model that is useful for analyzing data in "classification, regression, and outlier detection". It is particularly well-suitedto handling non-linear data, making it a powerful tool for many real-world applications. To calculate the weight vector in SVM, we use a specific mathematical equation that involves finding the optimal hyperplane that separates data points into different classes, while also minimizing classification errors. The LaGrange multipliers α are adjusted to control the trade-off between maximizing the margin between the hyperplane and the data points and minimizingclassification errors.

Given an input p, the weight vector Θ can be represented by a class or label d and LaGrange multipliers α is determined using the Eq. (25)

$$\Theta = \sum_{a=1}^{n} \alpha_a d_a p_a \qquad (25)$$

The SVM's objective is to improve the following equation.

$$Maximize_{\alpha_a} \sum_{a=1}^{n}$$

In Eq. (26), $< p_a p_b >$ is a vector that can be constructed using various kernels such as thepolynomial kernel, the Radial Basis Function kernel, and the Sigmoid Kernel". $\alpha_a - \sum$

$$\sum_{b=1}^{\text{Cjg}n}$$

$$\alpha_a \alpha_b d_a d_b < p_a \; p_b > \qquad (26)$$

3.4.1 ANN

Artificial Neural Networks (ANNs) are a type of machine learning technology that serves as the foundation for many deep learning techniques. We can train the ANN model usingraw data. It has a large amount of tuning parameters as compared to other classifiers, making it a complex structure. It also takes longer to optimize faults than other strategies. As a result, CUDA programming serves to train Neural Network algorithm instances in the GPU.

Each ANN Neuron Node is being trained thru a feature set $P = P_1, P_2, P_3, \ldots, P_m$

(where $P1 - P_m$ = Distinct features). The features are multiplied by random weights, $V =$

$V_1, V_2, V_3 \ldots \ldots V_m$, then augmented with bias values, $j = j_1, j_2, \ldots, j_m$ The values are then fed into a non-linear activation function. There are various kinds of activation functions. Some activation functions are shown in Eq. (27)-(30). In the equations, (a) denotes a single sample.

$$Sigmoid\ Function: (r) r\ i(r) = \frac{1}{1+e^{-r}} \qquad (27)$$

$$Tanh\ Function: i(r) = \frac{e^{r}-e^{-r}}{e^{r}+e^{-r}} \qquad (28)$$

$$Rectified\ Linear\ Un(RELU): i(r) = \max(0, r) \qquad (29)$$

$$Leaky\ RELU: (r) = \max(0.0001 * r, r) \qquad (30)$$

Following the implementation of the Non-Linear function, a softmax function is used to derivethe initial predicted value, as illustrated in Eq. (31).

$$Predicted\ Value: \hat{q}^{(r)} = \sigma(V^{transpose}\, p^{(r)} + j) \qquad (31)$$

Finally, the true and predicted values are used to generate the loss function, and the weights ofthe complete neural network design are altered using the backpropagation technique, gradient descent, and the loss function error. The subsequent equation describes the loss function.

$$LF(\hat{q}^{(r)}, q^{(r)}) = -(q^{(r)}\ log(\hat{q}^{(r)}) + (1 - q^{(r)})\log(1 - \hat{q}^{(r)})) \qquad (32)$$

2. Result and Discussion

The proposed model has been implemented using PYTHON. The proposed model hasbeen analyzed in terms of accuracy, precision, f-measure, sensitivity, specificity, NPV, FPR, FNR, and MCC.

4.1 Overall analysis of the performance metrics

Table 2 presents the results of evaluating five different models, CNN, Bi-LSTM, ANN,SVM, and Proposed, with a learning rate of 0. The models were evaluated using various metricsto assess their performance. The Proposed model achieved the highest Accuracy score at 0.912516, indicating that it was able to correctly classify a large proportion of the data. Similarly, the Precision score was also highest for the Proposed model at 0.810237, indicatingthat it had the highest proportion of true positives among the predicted positive values. The Sensitivity score, which measures the proportion of actual positives that were correctlyidentified, was highest for the Proposed model at 0.810237. Additionally, the Specificity score,which measures the proportion of actual negatives that were correctly identified, was highest for the Proposed model at 0.946609. The F-Measure, which is the harmonic mean of Precisionand Sensitivity, was highest for the Proposed model at 0.810237. The Matthews Correlation Coefficient (MCC), which is a quality measure for binary classifications, was highest for the Proposed model at 0.742051. The Negative Predictive Value (NPV), which measures the proportion of actual negatives among the predicted negative values, was highest for the Proposed model at 0.946609. The False Positive Rate (FPR), which measures the proportion of actual negatives that were incorrectly identified, was lowest for the Proposed model at 0.068186. The False Negative Rate (FNR), which measures the proportion of actual positives that were incorrectly identified, was also the lowest for the Proposed model at 0.204558. Overall, the Proposed model outperformed the other models in terms of most of the evaluatedmetrics, indicating its effectiveness in accurately classifying the data.

Table 2: Testing Metrices-Learn Rate--0

	CNN	Bi-LSTM	ANN	SVM	Proposed
Accuracy	0.859211	0.861210	0.884197	0.870871	0.912516
Precision	0.703627	0.707624	0.753600	0.726947	0.810237
Sensitivity	0.703627	0.707624	0.753600	0.726947	0.810237
Specificity	0.911072	0.912405	0.927730	0.918846	0.946609
F-Measure	0.703627	0.707624	0.753600	0.726947	0.810237
MCC	0.599904	0.605234	0.666535	0.630998	0.742051
NPV	0.911072	0.912405	0.927730	0.918846	0.946609
FPR	0.103723	0.102390	0.087065	0.095949	0.068186
FNR	0.311168	0.307170	0.261195	0.287847	0.204558

Table 3 shows the testing metrics for different machine learning models using a learningrate of 1. The models tested include CNN, Bi-LSTM, ANN, SVM, and Proposed. The accuracyof the models ranges from 0.771590 for SVM to 0.940501 for Proposed. The precision scoresvary from 0.528386 for SVM to 0.866207 for Proposed. The sensitivity scores range from 0.528386 for SVM to 0.866207 for Proposed, and the specificity scores range from 0.852659 for SVM to 0.965265 for Proposed. The F-measure for all models is similar, ranging from 0.707624 to 0.866207. The Matthews Correlation Coefficient (MCC) ranges from 0.366250 for SVM to 0.816678 for Proposed. The negative predictive value (NPV) scores for all modelsrange from 0.912405 to 0.965265. The false positive rate (FPR) scores range from 0.049529 for Proposed to 0.162136 for SVM, while the false negative rate (FNR) scores range from 0.148588 for Proposed to 0.486408 for SVM. Overall, the Proposed model shows the highest accuracy and precision scores, along with the highest MCC score and lowest FPR and FNR scores,

indicating better performance compared to the other models tested.

Table 3: Testing Metrics-Learn Rate—1

	CNN	Bi-LSTM	ANN	SVM	Proposed
Accuracy	0.863542	0.861210	0.871537	0.771590	0.940501
Precision	0.712289	0.707624	0.728280	0.528386	0.866207
Sensitivity	0.712289	0.707624	0.728280	0.528386	0.866207
Specificity	0.913959	0.912405	0.919290	0.852659	0.965265
F-Measure	0.712289	0.707624	0.728280	0.528386	0.866207
MCC	0.611453	0.605234	0.632775	0.366250	0.816678
NPV	0.913959	0.912405	0.919290	0.852659	0.965265
FPR	0.100835	0.102390	0.095505	0.162136	0.049529
FNR	0.302506	0.307170	0.286515	0.486408	0.148588

In Table 4, the testing metrics for different machine learning models with a learning rate of 2 are shown. The models include CNN, Bi-LSTM, ANN, SVM, and Proposed. The accuracy of the models ranges from 0.882198 for CNN to 0.964155 for Proposed. Bi-LSTM has the highest accuracy of 0.923176 among all the models. The precision of the models rangesfrom 0.749602 for CNN to 0.913515 for Proposed. Bi-LSTM has the highest precision of 0.831558 among all the models. The sensitivity and F-measure of the models are the same as their precision values. The specificity of the models ranges from 0.926397 for CNN to 0.981035 for Proposed. The proposed has the highest specificity among all the models. The Matthews correlation coefficient (MCC) of the models ranges from 0.661204 for CNN to 0.879755 for Proposed. Bi-LSTM has the highest MCC of 0.770480 among all the models. Thenegative predictive value (NPV) of the models ranges from 0.926397 for CNN to 0.981035 forProposed. The false positive rate (FPR) of the models ranges from 0.033760 for Proposed to 0.088398 for CNN, while the false negative rate (FNR) of the models ranges from 0.101280 for Proposed to 0.265193 for CNN. Overall, the Proposed model has the best performance in terms of accuracy, precision, specificity, MCC, NPV, FPR, and FNR among all the models.

Table 4: Testing Metrics-Learn Rate—2

	CNN	Bi-LSTM	ANN	SVM	Proposed
Accuracy	0.882198	0.923176	0.889195	0.902521	0.964155
Precision	0.749602	0.831558	0.763595	0.790247	0.913515
Sensitivity	0.749602	0.831558	0.763595	0.790247	0.913515
Specificity	0.926397	0.953716	0.931061	0.939945	0.981035
F-Measure	0.749602	0.831558	0.763595	0.790247	0.913515
MCC	0.661204	0.770480	0.679861	0.715398	0.879755
NPV	0.926397	0.953716	0.931061	0.715398	0.981035
FPR	0.088398	0.061079	0.083733	0.074849	0.033760
FNR	0.265193	0.183236	0.251200	0.224547	0.101280

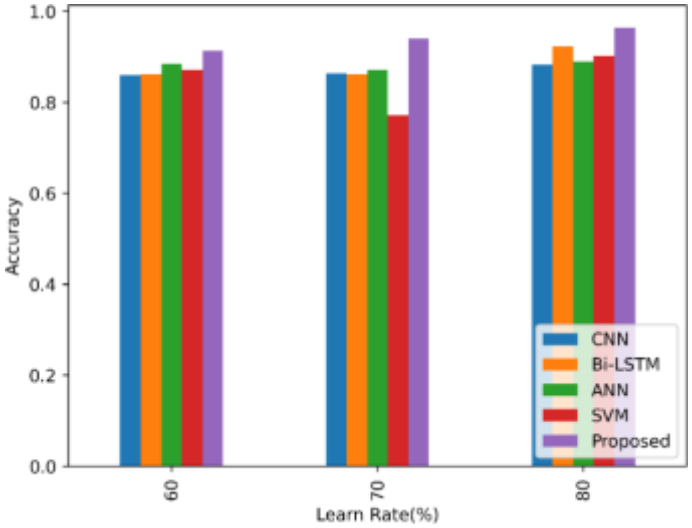


Figure 3: Graphical representation of performance metrics-Accuracy

In Figure 3, accuracy is visually represented as a performance metric. It is a measure of the model's ability to correctly predict the outcome, expressed as a percentage. This graphical representation helps in evaluating the effectiveness of the model's predictions.

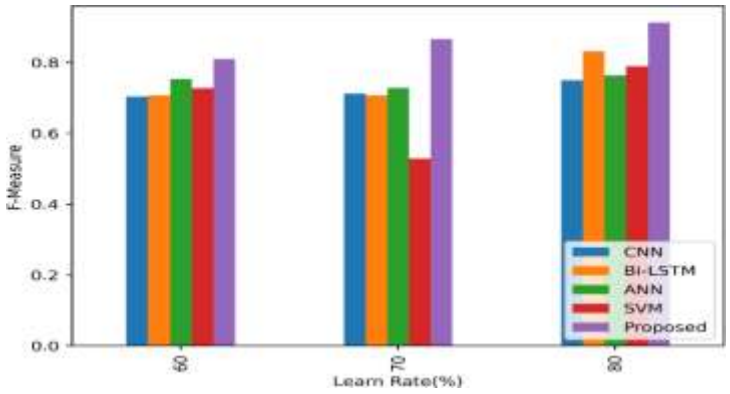


Figure 4: Graphical representation of performance metrics-F-Measure

Figure 4 displays F-Measure as a performance metric. It is a measure of the model's precisionand recall, where the higher the F-Measure, the better the model's ability to accurately predictthe positive outcome. This graphical representation can help in determining the effectiveness of the model's predictions and in making improvements to the model.

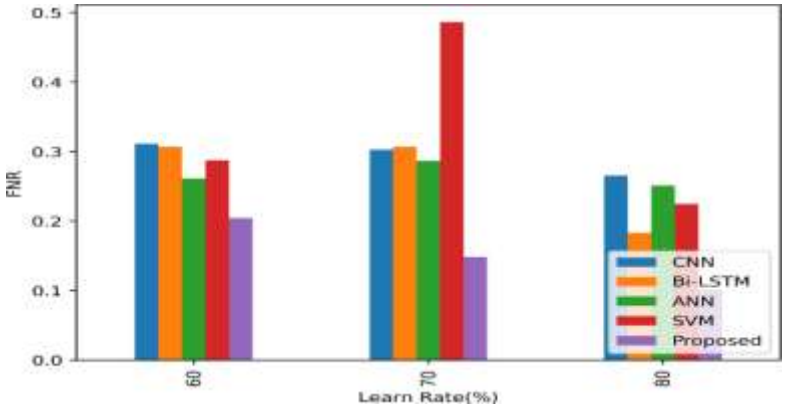


Figure 5: Graphical representation of performance metrics-FNR

Figure 5 represents FNR (False Negative Rate) as a performance metric. It measures the percentage of actual positive cases that the model predicted as negative. A lower FNR indicatesbetter performance of the model in identifying positive cases. This graphical representation can help in evaluating and improving the model's performance in detecting positive cases.

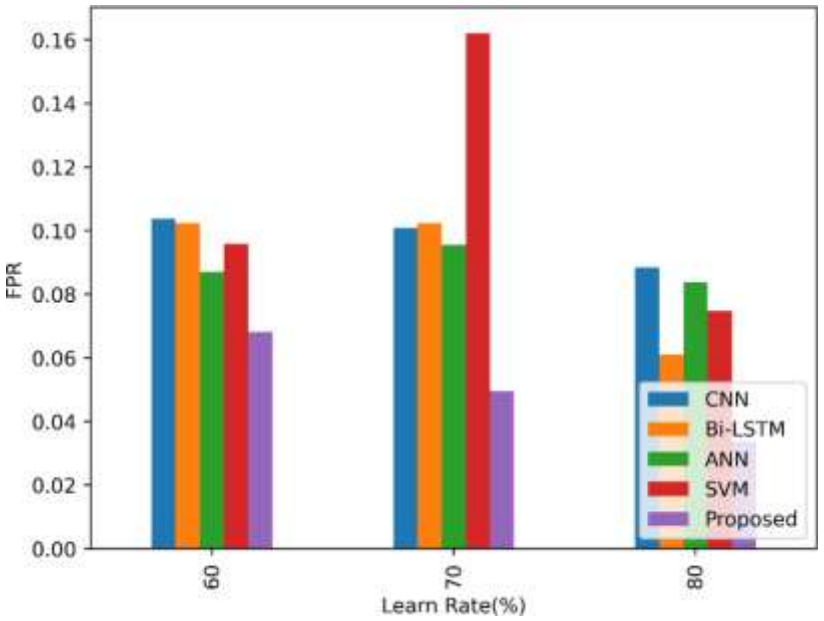


Figure 6: Graphical representation of performance metrics-FPR

Figure 6 shows FPR (False Positive Rate) as a performance metric. It measures the percentageof actual negative cases that the model predicted as positive. A lower FPR indicates better performance of the model in avoiding false positive predictions

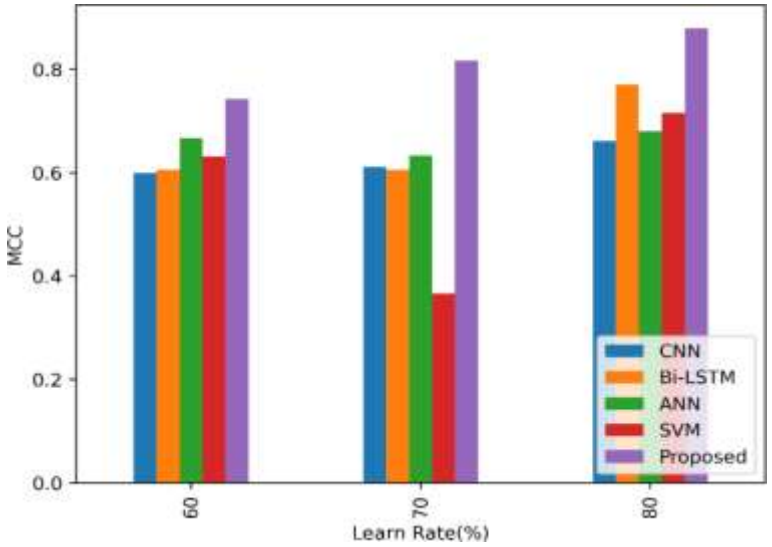


Figure 7: Graphical representation of performance metrics-MCC

Figure 7 illustrates MCC (Matthews Correlation Coefficient) as a performance metric. It is a measure of the correlation between the predicted and actual values, ranging from -1 to 1, withhigher values indicating better performance. MCC takes into account true positive, true negative, false positive, and false negative predictions, making it a more comprehensive metric than others. This graphical representation can help in evaluating and improving the overallperformance of the model.

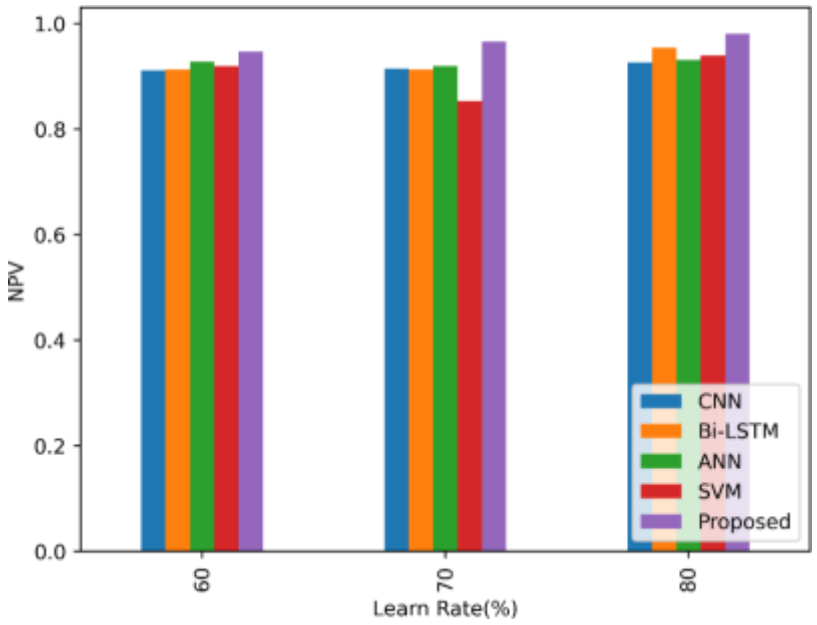


Figure 8: Graphical representation of performance metrics-NPV

In Figure 8, NPV (Negative Predictive Value) is displayed as a performance metric. It measures the percentage of actual negative cases that the model correctly predicted as negative.A higher NPV indicates better performance of the model in correctly identifying negative cases. This graphical representation can help in evaluating and improving the model's performance in avoiding false negative predictions. NPV is particularly important in medical applications, where a false negative prediction can have severe consequences.

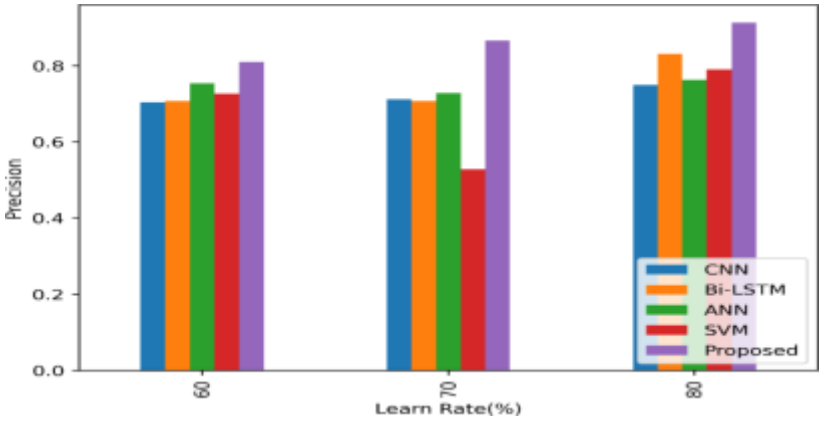


Figure 9: Graphical representation of performance metrics-Precision

Figure 9 illustrates precision as a performance metric. It is a measure of the model's ability to correctly predict positive cases out of all the positive predictions. A higher precision indicatesbetter performance of the model in avoiding false positive predictions.

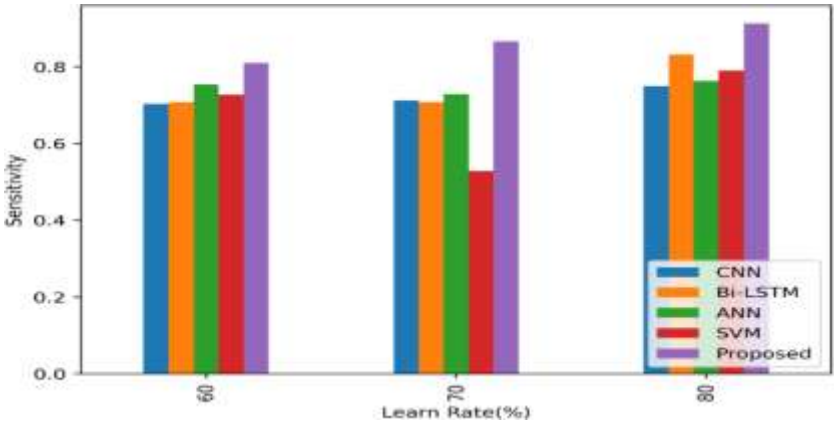


Figure 10: Graphical representation of performance metrics-Sensitivity

In Figure 10, sensitivity is visually represented as a performance metric. It measures the percentage of actual positive cases that the model correctly predicted as positive

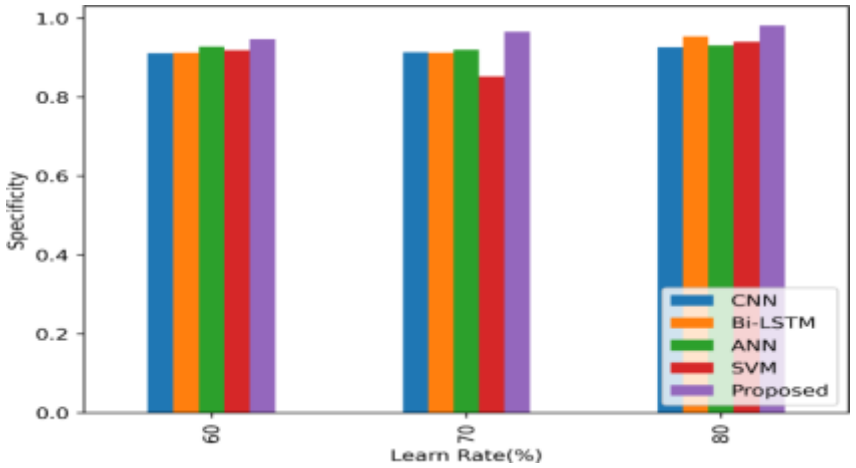


Figure 11: Graphical representation of performance metrics-Specificity

Figure 11 displays specificity as a performance metric. It measures the percentage of actual negative cases that the model correctly predicted as negative. A higher specificity indicates better performance of the model in avoiding false positive predictions.

3. Conclusion

In conclusion, the proposed solution offers an innovative approach to enhancingsecurity in urban areas by effectively analyzing video stream data with quick and accurate identification of criminal activity. The solution addresses the challenging task of monitoring alarge volume of video data by pre-processing it with techniques such as Video-to-Frame Conversion, Resizing, and Normalization, followed by segmentation of the frames using an optimized Semantic Segmentation-Optimized FCN algorithm. Features were then extracted from the segmented regions using techniques such as SIFT and the proposed Improved Histogram of Oriented Gradients algorithm. The extracted features are refined using the new improved Relief Algorithm for feature selection, which results in a reduced number of featureswith improved accuracy. The hybrid machine learning approach, which combines the transformer model, SVM, and ANN, was expected to provide improved accuracy in crime anomaly detection. The implementation of the proposed solution using the Python programming language can potentially facilitate its adoption by law enforcement agencies for practical crime prevention and detection. Overall, the proposed solution offers a promisingapproach to improving security and reducing crime rates in urban areas.

References

[1] K. Guo, Y. Lu, H. Gao, and R. Cao, “Artificial intelligence-based semantic internet of things in a user-centric smart city,” Sensors (Switzerland), vol. 18, no. 5, 2018, doi: 10.3390/s18051341.

[2] M. Islam, A. S. Dukyil, S. Alyahya, and S. Habib, “An IoT Enable Anomaly Detection System for Smart City Surveillance,” Sensors, vol. 23, no. 4, 2023, doi: 10.3390/s23042358.

[3] Y. Guo, T. Ji, Q. Wang, L. Yu, G. Min, and P. Li, “Unsupervised Anomaly Detection in IoT Systems for Smart Cities,” IEEE Trans. Netw. Sci. Eng., vol. 7, no. 4, pp. 2231–2242, 2020, doi: 10.1109/TNSE.2020.3027543.

[4] F. Cauteruccio et al., “A framework for anomaly detection and classification in Multiple IoT scenarios,” Futur. Gener. Comput. Syst., vol. 114, pp. 322–335, 2021, doi: 10.1016/j.future.2020.08.010.

[5] Y. M. Tukur, D. Thakker, and I. U. Awan, “Edge-based blockchain enabled anomaly detection for insider attack prevention in Internet of Things,” Trans. Emerg. Telecommun. Technol., vol. 32, no. 6, pp. 1–23, 2021, doi: 10.1002/ett.4158.

[6] M. A. Rahman, A. T. Asyhari, L. S. Leong, G. B. Satrya, M. Hai Tao, and M. F. Zolkipli, “Scalable machine learning-based intrusion detection system for IoT-enabled smart cities,” Sustain. Cities Soc., vol. 61, p. 102324, 2020, doi: 10.1016/j.scs.2020.102324.

[7] W. Ullah, A. Ullah, I. U. Haq, K. Muhammad, M. Sajjad, and S. W. Baik, “CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks,” Multimed. Tools Appl., vol. 80, no. 11, pp. 16979–16995, 2021, doi: 10.1007/s11042-020-09406-3.

[8] S. Singh, P. K. Sharma, B. Yoon, M. Shojafar, G. H. Cho, and I. H. Ra, “Convergence of blockchain and artificial intelligence in IoT network for the sustainable smart city,” Sustain. Cities Soc., vol. 63, 2020, doi: 10.1016/j.scs.2020.102364.

[9] S. Manimurugan, “IoT-Fog-Cloud model for anomaly detection using improved Naïve Bayes and principal

component analysis," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2021, doi: 10.1007/s12652-020-02723-3.

[10] P. Kumar et al., "PPSF: A Privacy-Preserving and Secure Framework Using Blockchain-Based Machine-Learning for IoT-Driven Smart Cities," IEEE Trans. Netw. Sci. Eng., vol. 8, no. 3, pp. 2326–2341, 2021, doi: 10.1109/TNSE.2021.3089435.

[11] R. Xu, Y. Cheng, Z. Liu, Y. Xie, and Y. Yang, "Improved Long Short-Term Memory based anomaly detection with concept drift adaptive method for supporting IoT services," Futur. Gener. Comput. Syst., vol. 112, pp. 228–242, 2020, doi: 10.1016/j.future.2020.05.035.

[12] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," Internet of Things (Netherlands), vol. 7, p. 100059, 2019, doi: 10.1016/j.iot.2019.100059.

[13] D. Alahakoon, R. Nawaratne, Y. Xu, D. De Silva, U. Sivarajah, and B. Gupta, "Self-Building Artificial Intelligence and Machine Learning to Empower Big Data Analytics in Smart Cities," Inf. Syst. Front., vol. 25, no. 1, pp. 221–240, 2023, doi: 10.1007/s10796-020-10056-x.

[14] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "FogFlow: Easy Programming of IoT Services Over Cloud and Edges for Smart Cities," IEEE Internet Things J., vol. 5, no. 2, pp. 696–707, 2018, doi: 10.1109/JIOT.2017.2747214.

[15] D. J. Mandala, P. Akhila, and V. S. Reddy, "An Integrated Reinforcement DQNN Algorithm to Detect Crime Anomaly Objects in Smart Cities," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 12, pp. 348–352, 2021, doi: 10.14569/IJACSA.2021.0121246.

[16] T. Ahmad and D. Zhang, "Using the internet of things in smart energy systems and networks," Sustain. Cities Soc., vol. 68, no. February, p. 102783, 2021, doi: 10.1016/j.scs.2021.102783.

[17] A. Garcia-Serrano, "Anomaly Detection for malware identification using Hardware Performance Counters," pp. 1–12, 2015, [Online]. Available: http://arxiv.org/abs/1508.07482.

[18] M. Mohammadi and A. Al-Fuqaha, "Enabling Cognitive Smart Cities Using Big Data and Machine Learning: Approaches and Challenges," IEEE Commun. Mag., vol. 56, no. 2, pp. 94–101, 2018, doi: 10.1109/MCOM.2018.1700298.

[19] I. H. Sarker, Smart City Data Science: Towards data-driven smart cities with open research issues, vol. 19, no. April. 2022.

[20] A. Al-Qarafi et al., "Optimal Machine Learning Based Privacy Preserving Blockchain Assisted Internet of Things with Smart Cities Environment," Appl. Sci., vol. 12, no. 12, 2022, doi: 10.3390/app12125893.

[21] L. Yang, N. Elisa, and N. Eliot, Privacy and security aspects of E-government in smart cities. Elsevier Inc., 2018.

[22] D. K. Sharma et al., "Anomaly detection framework to prevent DDoS attack in fog empowered IoT networks," Ad Hoc Networks, vol. 121, no. April, p. 102603, 2021, doi: 10.1016/j.adhoc.2021.102603.

[23] I. V. D. Srihith, I. V. S. Kumar, R. Varaprasad, Y. R. Mohan, T. A. S. Srinivas, and Y. Sravanthi, "Future of Smart Cities: The Role of Machine Learning and Artificial Intelligence," South Asian Res. J. Eng. Technol., vol. 4, no. 5, pp. 110–119, 2022, doi: 10.36346/sarjet.2022.v04i05.005.