An Optimization of GCN-GRU Crime Prediction Network using Lyrebird Optimization Algorithm

Sundari Palanisamy [1], Malathi Arunachalam [2], Raniyaharini Rajendran [3]

[1]PG and Research Department of Information Technology, Government Arts College (Autonomous), Coimbatore, Tamil Nadu, India.
[2]PG and Research Department of Computer Science, Government Arts College (Autonomous), Coimbatore, Tamil Nadu, India.
[3]Master Graduate, Freie Universität Berlin, Germany

[1]sundaripalanisamyphd@yahoo.com

---***---

Abstract - Crime detection is an important aspect of helping law enforcement agencies to prevent new crimes by finding patterns. Nonetheless, the dynamic aspect of criminality and the swiftness of crime occasion creates immense challenges in precise classification of crimes in the future. To overcome this problem, different Deep Learning (DL) models have been examined in crime prediction activities. The Graph Convolutional Network with Gated Recurrent Unit (GCN-GRU) has been promising among them and can capture the spatial and temporal characteristics of crime. GCN models the local and global spatial dependencies by dynamically refining graph topologies, providing resistance to noisy or incomplete data and providing overfitting resistance. Concurrently, GRU which has lightweight architecture, learns effectively both short as well as long-term temporal dependencies in sequential crime information. The classification performance of the model is further enhanced with the Cross-Entropy Loss, which gives greater confidence to the correct crime types. Despite these advantages, the GCN-GRU model suffers from high computational complexity and strong dependence on data quality. Additionally, hyperparameters can be manually tuned and this is time consuming and can lead to suboptimal performance. In order to address these shortcomings, this paper presents an Optimization-based GCN-GRU (O-GCN-GRU) model that could be used to optimize the hyperparameters of the GCN-GRU network to maximize crime prediction results. This model uses a new metaheuristic optimization algorithm called Lyrebird Optimization Algorithm (LOA), inspired by the manner in which lyrebirds in the wild react to threats in their environment. The LOA has two phases, of exploration and exploitation whereby lyrebirds replicate an escape strategy to search a large number of possible solutions and a hiding strategy to search intensively within promising areas. The best hyperparameters of GCN-GRU model are determined based on these processes to make crime prediction with crime data effective. Through the application of LOA, this model is effective in fine-tuning the properties of the GCN-GRU network, which leads to both high predictive ability and less computation costs. Lastly, the experimental findings indicate that the O-GCN-GRU model has an accuracy rate of 96.8%, which is higher than the other models of crime predictions in terms of performance and reliability.

Key Words: Crime Prediction, Deep Learning, GCN-GRU, Hyperparameter tuning and Lyrebird Optimization Algorithm

1. INTRODUCTION

Criminal actions continue to be a problem as societies evolve. An increase in criminal activity has a negative effect on people's standard of living and impedes societal and economic development [1]. Improving public safety and decreasing government expenses are two outcomes of effective crime prevention. The development of better geographic information gathering tools has made it possible to accurately capture crime data across areas in this era of big data. Machine learning models have the potential to revolutionize crime prevention in many different fields.

Numerous academics are currently investigating the feasibility of crime spatiotemporal forecasting [2,3]. In their early iterations, spatiotemporal crime prediction systems primarily used location data and the idea of neighborhood repetition effects to model and predict future criminal acts [4]. These methods use analysis of crime statistics to create algorithmic models that may foretell where crimes will occur within predetermined time and space constraints. A popular approach is the nonparametric Kernel Density Estimation (KDE) approach, which enables the display of crime distributions. The appropriate authorities might identify high-incidence locations for focused surveillance using KDE. A KDE-based model for assessing the probability of crime occurring in a region was proposed by Bowers et al. [5]. To develop a spatiotemporal similarity-based crime prediction method, Xu et al. [6] expanded the KDE model with a time component and demonstrated its efficacy. Additionally, some academics have noted that the geographical and temporal distribution of criminal activity resembles that of aftershock occurrences after earthquakes. Separating the spatial and temporal aspects of criminal occurrences and applying kernel functions to evaluate the probability of crime is one way that Mohler et al. [7] applied the aftershock model from seismology to forecast home invasions. There are a number of drawbacks to these approaches, despite their apparent promise in enhancing the accuracy of crime predictions: (1) When applied to huge geographic regions, these techniques reliably estimate the crime risk. (2) Statistical approaches that are oversimplified fail to capture the nuances and variations in crime data across both short as well as extended time periods.

Now, with the advent of machine learning, more flexible and precise predictions can be generated, particularly where complex crime data is concerned. Random Forest (RF) [8], XGBoost [9], and autoregressive integrated moving average model models have become popular in crime spatiotemporal prediction. Liu et al. [10] compared RF with spatiotemporal kernel density mapping to predict crime hotspots, and Yan et al. [11] suggested a XGBoost-based approach in predicting a range of theft crimes. Nonetheless, such models tend to find it hard to reflect complex time variations in crime data.

On the other hand, DL models perform better in both short- and long-term feature extraction of progressively more complex and diverse data, which results in their extensive application in many fields [12,13]. Yan and Hou [14] provide an example of using Long Short-Term Memory (LSTM) networks to make predictions about the time series of theft crime, and Cortez et al. [15] created a custom LSTM architecture to make predictions related to emergencies. Enhancements have also been made to ensemble models that incorporate predictions of various algorithms in order to increase accuracy. Duan et al. [16] introduced a new spatiotemporal crime network based on deep Convolutional Neural Networks (CNNs) to automatically realize crime-associated features, which were subsequently passed on recurrent neural networks to be predicted. For better crime prediction, Yi et al. [17] used LSTM in combination with an improved continuous conditional random field, and Dong et al. [18] made adjustments to the ST-3DNet model to get finer temporal scales. Unfortunately, a lot of these approaches can't handle data with spatial properties.

Graph Neural Networks (GNNs) have been in the focus recently due to their capacity to simulate intricate spatiotemporal interactions [19], which has led to their application in fields such as crime prediction. For the purpose of future crime prediction, Liang et al. [20] combined a gated recurrent unit with multi-GCNs to capture spatial relationships from many factors. In order to detect latent region-wise dependencies and geographic correlations, Jin et al. [21] suggested a combined model that uses CNNs and adaptive GCNs. There are a number of obstacles that must be overcome in order to properly analyze crime data, despite the fact that these methodologies have shown encouraging results: Issues at the data level (1) There are inherent errors and unpredictable fluctuations in crime data. This leads to noise which may adversely impact the accuracy of prediction. It is important to have effective measures to reduce this noise. (2) Level-of-label issues: It is possible to have a variety of crime types that occur in the same time interval and the levels of different crime types tend to be related. An efficient model that can help in detecting these latent relationships is necessary in order to enhance the effectiveness of prediction.

Shan et al [22] suggested an adaptive GCNLSTM (Ada-GCNLSTM) to solve these problems. The objective of this model is to derive more generalized features and base entirely on the spatiotemporal associations of the various types of crimes to improve prediction ability. In particular, the model utilizes GCN to reveal unobservable spatial aspects of crimes and Maximizes Mean Discrepancy (MMD) to minimize the gap between a Gaussian distribution. This reduces the effects of outliers and enhances generalizability. Then LSTM is used to extract the temporal information and Relational Mechanism Units (RMUs) are introduced to LSTM to extract the latent relationships among various types of crime which enhances further predictive performance. Despite these advantages, the GCN-GRU model suffers from high computational complexity and strong dependence on data quality. In addition, tuning of the hyper parameters manually is time-consuming and might lead to poor performance.

Therefore, this paper presents an Optimization-driven GCN-GRU (O-GCN-GRU) model that can be used to improve the predictive ability of crime and decrease the computational complexity. In this model, a new metaheuristic optimization method is used, the Lyrebird Optimization Algorithm (LOA) is utilized to fine-tune the hyperparameters of the GCN-GRU network. The behavior of lyrebirds in reaction to the threat of the environment is the basis of the LOA, which comprises two key phases, exploration and exploitation. In the exploration, the lyrebirds imitate an escape strategy to search over a large set of possible solutions, whereas in exploitation they imitate a hiding strategy to target the search within the promising areas. After this approach, the best hyperparameters of the GCN-GRU model are chosen with ease in the model training without being overly complex. After tuning, it is trained and validated using GCN-GRU so that it can be able to predict crime intensities. Using LOA, the hyperparameters of the GCN-GRU are set in the best way, which results in increased prediction accuracy and reduced computational power in crime prediction.

Here is the outline of the rest of the paper: The relevant literature is reviewed in Section 2. In Section 3, the O-GCN-GRU model is described in depth. The study's findings and suggestions for future improvements are presented in Section 5, while the experimental results are presented in Section 4.

2. LITERATURE SURVEY

Das et al. [23] proposed a discrete Multi-Objective PSO-based classifier (MOPSO) for crime prediction using historical report data. The system initialized a population of candidate rule sets, evolved over generations using particle swarm operations. To maximize rule accuracy and confidence, two PSO-based classifiers Dominance Relationship based PSO (DRPSO) as well as Non-Dominated Pareto Front based PSO (NPPSO) were used. To enable continuous learning, incremental versions INRDRPSO and INRNPPSO were developed, updating models with incoming data without retraining. Although both incremental variants improved learning efficiency, INRNPPSO achieved superior accuracy and adaptability over INRDRPSO and competing models. However, the system's dependency on high-quality rule formulation poses a challenge in highly dynamic crime environments.

Jeyaboopathiraja & Priscilla [24] proposed a novel crime trend prediction model integrating data preprocessing, optimization, and deep learning. Initially, missing values in the dataset were handled using Predictive Mean Matching to improve data quality. Then, an Improved Bat Optimization (IBAT) algorithm was employed to extract the most relevant features, enhancing prediction performance and reducing computational time. Finally, a Convolutional Neural Network (CNN) was used to forecast future crime trends. While the approach showed promise in combining feature optimization and deep learning for crime forecasting, its overall performance under varying data scales and real-time applicability was not extensively evaluated.

Singuluri et al. [25] devised a Modified Capsule Network with Crisscross Optimization (MCN-CCO) for the cyber-crime prediction. By combining the outputs of both networks to optimize their strengths and boost the detection rate, the approach combines MCN with Multilayer Perceptron (MLP) using a rule-based strategy. Using the CCO method, the capsule network's hyper-parameters were then fine-tuned and improved for crime detection. Unfortunately, F1-Score and training time were both negatively affected by this model.

Mithoo & Kumar [26] presented a Spizella Swarm Optimization based Bidirectional LSTM (SSO-BiLSTM) using twitter data for crime rate detection. The pre-processed and augmented data were inputted to BiLSTM to forecast the crime patterns in related to time period. The hyper-parameter of BiLSTM were fine-tuned by SSO for convergence enhancements and lowering the model's complexity. But the model's performance was hindered due to limited training data which restricts the accuracy and sensitivity rate.

A hybrid DL method for identifying theft incidents in surveillance footage was suggested by Waddenkery and Soma [27]. A Deep Maxout Network is used for event categorization after the Video Summarization (VSUMM) methodology is used to summarize video frames in the method. The newly-introduced Adam-Dingo Optimizer is used to adjust the weights of the network in order to improve performance. The model's capacity to differentiate between instances of theft and everyday behaviors is enhanced by this combination. However, the model's effectiveness under varying weather conditions was not explored, which remains a potential area for future work.

To evaluate human movement patterns, Waddenkery and Soma [28] developed a Loitering-based Human Crime Detection (LHCD) algorithm using video surveillance. This module combines an improved version of Deep Simple Online Real-time Tracking (DSORT) based on geometrical principles with a Segmentation Quality Assessment (SQA) algorithm. To strengthen the model's detection accuracy and system responsiveness, it incorporates a Deep CNN (DCNN) along with the Beluga Whale Adam Dingo Optimizer (BWADO). The approach effectively reduces false alarms and outperforms existing methods across key performance metrics. However, the model's complexity and dependence on accurate loitering behavior detection may impact its adaptability in highly dynamic environments.

An RICCNN-CP-JBOA, or Rotation Invariant Co-ordinate CNN based Crime Prediction, was developed by Devi et al. [29]. Following extraction from the crime dataset, the data was pre-processed with the help of Sub Adaptive Two-stage Unscented Kalman Filtering (STSUKF) to remove any irrelevant or missing values. The six features were selected using Parrot Optimization (PO), and the dynamic features were extracted using the Random Quantum Circuits Transform (RQCT). The RICCNN is fed the results of the feature selection process in order to make predictions about the categories of crimes. Optimizing the RICCNN settings was done using the Jarrett-Butterfly Optimization Algorithm (JBOA). However, this model results in high training time.

Alshahrani [30] introduced a hybrid crime prediction model combining Neural Architecture Search (NAS) and hyperparameter tuning to automate neural network design. The approach utilized NAS to discover optimal architectures and fine-tuned their hyperparameters for binary crime classification. Feature selection was performed using Robust Rank Aggregation (RRA), selecting key predictors like age, location, and month. The model was evaluated on three datasets, including a confidential criminal cases dataset and publicly available Vancouver and Austin crime datasets. NAS+ outperformed traditional methods across all datasets. However, the model's reliance on sensitive data and varying performance across datasets may limit its generalizability.

3. PROPOSED METHODOLOGY

This section explains the proposed O-GCN-GRU model for crime prediction in detail. Fig. 1 shows a schematic of the proposed study's workflow.

1) Model Overview

Initially, the crime-related data are collected from multiple sources, including crime reports, victim surveys, self-report surveys, and police statistics. The collected data are then preprocessed to handle missing values, normalize attributes, and remove redundant information. Next, spatial and temporal features are extracted using a GCN–GRU based feature extraction block. Specifically, the GCN captures spatial correlations and topological dependencies among crime categories and geographical regions, while the GRU models temporal dependencies and dynamic variations in crime occurrences over time.

The structures of GCN and GRU are detailed in the proposed framework. Further, the extracted features are forwarded to the classification layer for crime prediction. To improve prediction accuracy, the model hyperparameters are tuned using the LOA. The hyperparameters of the GCN–GRU model include learning rate, dropout rate, number of hidden units, batch size, number of epochs, weight decay, and activation functions. For reliable and efficient model training, the LOA determines the best values for these parameters. Figure 2 shows the structure of the LOA-optimized GCN-GRU model. This LOA is described in depth in the section that follows:

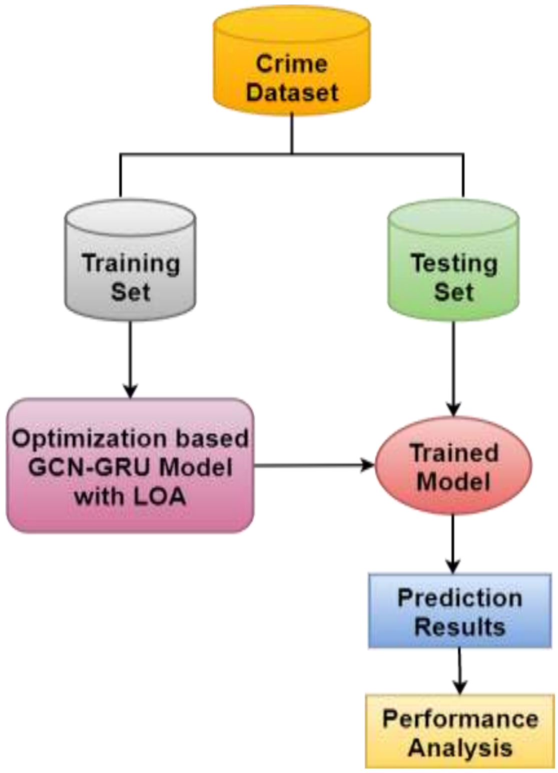

Fig -1: Pipeline of the Proposed Model

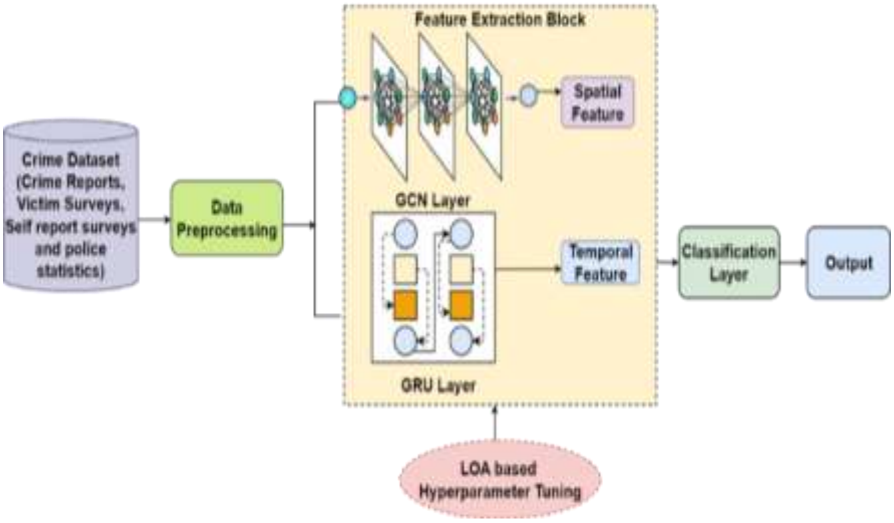

Fig -2: Optimized GCN-GRU model using LOA

2) Lyrebird Optimization Algorithm

One example of an optimization algorithm that may use some inspiration comes from the lyrebird, an Australian songbird that can imitate a wide variety of sounds and change its behavior in response to danger by looking around and deciding whether to flee or hide. Detailed explanations of the LOA's mathematical modeling are provided below.

3.1 Initialization

The suggested LOA is a metaheuristic method that uses lyrebirds as its population model. Through an iterative process that utilizes the search capabilities of its members inside the solution space, it produces the optimum solutions to optimization problems.

For each lyrebird in LOA, the decision factors are weighted according on its location in the solution space. The mathematical representation of a lyrebird is a vector, with each member representing a decision. For each lyrebird in LOA, the decision factors are weighted according on its location in the solution space. The mathematical representation of a lyrebird is a vector, with each member representing a decision variable. All members of the LOA combined make up the algorithm's population, which is represented as a matrix in Equation (1).

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_x \\ \vdots \\ M_N \end{bmatrix}_{N\times z} = \begin{bmatrix} m_{(1,1)} & \cdots & m_{(1,y)} & \cdots & m_{(1,z)} \\ \vdots & \ddots & \vdots & ⋰ & \vdots \\ m_{(x,1)} & \cdots & m_{(x,y)} & \cdots & m_{(x,z)} \\ \vdots & ⋰ & \vdots & \ddots & \vdots \\ m_{(N,1)} & \cdots & m_{(N,y)} & \cdots & m_{(N,z)} \end{bmatrix}_{N\times z} \quad (1)$$

The initial positions of the lyrebirds are randomly generated within the defined bounds of each decision variable using Eq. (2)

$$m_{x,y} = lb_y \times r.\left(ub_y - lb_y\right) \quad (2)$$

In equations (1) and (2), the following symbols are used: M for the population matrix, M_x for the x^{th} member of the LOA (a potential solution), $m_{x,y}$ for the y^{th} dimension (the decision variable), N for the number of lyrebirds, z for the number of decision variables (the total number of hyper parameters of the GCN-GRU approach), r for a random value between 0 and 1, and lb_y for the y^{th}decision variable of lower bound and ub_y for the upper bound.

Every member of the LOA represents a potential solution to the issue; hence it is possible to calculate the objective function for each member. Accordingly, the population size is directly proportional to the number of values for the objective function. The values of the evaluated objective functions can be represented as a vector in Eq. (3).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_x \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(M_1) \\ \vdots \\ F(M_x) \\ \vdots \\ F(M_N) \end{bmatrix}_{N\times 1} \quad (3)$$

The function of fitness values is represented by F in Eq. (3), while F_x is the obtained value for the x^{th} member of the LOA. For possible solution viability evaluations, observed values of the goal function are crucial. When looking at potential solutions, the best value indicates the best option, and the worst value indicates the worst option. Modifying the best potential solution is a necessary step in every search iteration since it updates the lyrebird's location within the search space.

3.2 Lyrebird Behavior Modeling in LOA

The LOA mimics two main lyrebird strategies when sensing danger: escaping (exploration phase) and hiding (exploitation phase). Using Eq. (4), LOA mimics the lyrebird's decision-making process as it chooses between hiding and escaping in a dangerous situation. So, in every cycle, the positions of all LOA members are changed depending on either the first as well as second phase.

.

$$Update\ process\ for$$
$$M_x : \begin{cases} based\ on\ Phase1, & r_p \le 0.5 \\ based\ on\ Phase2, & else \end{cases} \quad (4)$$

In this case, r_p is an integer valued at random from 0 to 1. A lyrebird will choose one of these methods at random with an equal chance in each iteration.

Phase 1: Escaping Strategy (Exploration Phase)

This LOA step involves updating the population's search area location based on lyrebird escape behavior modeling, which involves moving away from dangerous locations and towards safer ones. Moving to a safe place and scanning different parts of the problem-solving arena are examples of the lyrebird's extensive positional adjustments, which indicate the LOA's exploratory capability in a global search. By design, LOA takes into account the whereabouts of other individuals with higher objective function values and treats them as safe zones. Thus, the range of safe zones of all the LOA members can be calculated based on Eq. (5).

$$SA_x = \{M_k, \quad F_k < F_x \; and \; k \in \{1,2,\dots,N\}\}$$

$$where \; i = \{1,2,\dots,N\} \tag{5}$$

In this case, SA_x represents the series of safe zones for the x^{th} lyrebird and M_k stands for the k^{th} row of the M matrix, which has a higher objective function value F_k than the x^{th} member of the LOA ($F_k < F_x$). The LOA design assumes that the lyrebird will randomly make its way to one of these secure locations. Each member of the LOA has their new position determined using Eq. (6) according on the lyrebird displacement framework used in this phase. After that, this new position will take the place of the previous one of the matching members in accordance with Eq. (7) if the objective function's value is enhanced.

$$m_{x,j}^{P1} = m_{x,j} + r_{x,j}.(SSA_{xj} - I_{x,j}.m_{x,j}) \tag{6}$$

$$M_x = \begin{cases} M_x^{P1}, & F_x^{P1} \leq F_x \\ M_x, & else \end{cases} \tag{7}$$

In this context, SSA_x denotes the chosen safe area for the x^{th} lyrebird, $SSA_{x,j}$ denotes its x^{th} dimension, M_x^{P1} denotes the new location determined for the x^{th} lyrebird according to the escape strategy of the proposed LOA, $m_{i,j}^{P1}$ denotes its j^{th} dimension, F_x^{P1} stands for the objective function's value, $r_{x,j}$ are random integers through the interval [0, 1], and $I_{x,j}$ are randomly assigned 1 or 2.

Phase 2: Hiding Strategy (Exploitation Phase)

This part of the LOA involves updating the search space positions of all members of the population based on the lyrebird's plan to hide in a nearby safe region. The lyrebird's capacity to use LOA in local search can be observed in its careful assessment of its surroundings and its tiny steps to seek an appropriate hiding site.

In LOA design, each member of the LOA is given a new position using Eq. (8), which is based on the lyrebird's movement model towards the near-suitable hiding spot. Assuming this new position increases the value of the goal function as per Eq. (9) it will replace the prior position of the relevant member.

$$m_{x,j}^{P2} = m_{x,j} + (1 - 2r_{x,j}).\frac{ub_j - lb_j}{t} \tag{8}$$

$$M_x = \begin{cases} M_x^{P2}, & F_x^{P2} \leq F_x \\ M_x, & else \end{cases} \tag{9}$$

In equations (8) and (9), the variable M_x^{P2} represents the x^{th} lyrebird's new position determined by the suggested LOA's hiding technique, $m_{x,j}^{P2}$. As for its j^{th} dimension, the value of its objective function is denoted by F_x^{P2}, the iteration counter is represented by t, and $r_{x,j}$ are random values from the range [0, 1].

Therefore, the LOA is an iterative method that compares fitness function values, adjusts the best candidate solution based on those comparisons, and updates lyrebird locations in the first iteration. In the final iteration, the algorithm adjusts lyrebird's positions, and after complete implementation, the optimal candidate solution, which includes the hyperparameters that performed the best during the iterations, is considered as the correct solution to the problem. Fig. 3 shows the LOA's overall procedure, while Algorithm 1 describes the LOA's pseudocode for hyperparameter modification.

Table -1: List of Optimal Hyperparameters for GCN-GRU Model

Parameters	Search Space	Optimal Range
Kernel Size	[3x3, 5x5, 7x7]	3x3
No. of GCN layers	[2, 4, 6]	2
Filter Size	[32, 64, 128, 256]	64
Pool Size	[1, 3, 5, 7]	5
No. of GRU layers	[1, 2, 3]	2
GRU Output Size	[16, 32, 64]	32
Dropout	[0.1, 0.2, 0.3, 0.5]	0.2
Activation Operation	[Tanh, Sigmoid, ReLU]	ReLU
Momentum	[0.6, 0.7, 0.8]	0.8
Batch size	[8, 16, 32, 64, 128]	16
Rate of Training	[0.00001, 0.0001, 0.001, 0.01]	0.0001
Optimizer	[Adam, SGD, RMSprop]	Adam
Number of epochs	[50, 100, 200, 300]	150
Loss Function	[Cross-entropy, Mean Squared Error]	Cross Entropy

Algorithm 1: Hyperparameter tuning using LOA
Input: Set of hyperparameters for the GCN-GRU model
Output: Optimal hyperparameters

1. **Begin**
2. //Initialization stage:
3. Set the LOA population size N as well as define the maximum number of iterations as T;
4. Make a randomly generated initial population matrix according to Eqn. (2);
5. Apply Equation (3) to the objective function;
6. Choose the potential solutions into the best
7. $\boldsymbol{for}(t = 1{:}T)$
8. $\boldsymbol{for}(x = 1{:}N)$
9. Analyze Eq. (4) to get the x^{th} lyrebird's defense strategy when faced with a predator;
10. if $r_p \leq$ 0.5 (Choose Escaping Strategy)
11. //**Exploration Stage**
12. Determine candidate safe areas for the x^{th} lyrebird based on Eq. (5);
13. Using Eq. (6), determine the new location of the x^{th} LOA member.
14. Apply Equation (7) to the x^{th} member of the LOA;
15. else (select Hiding Strategy):
16. //**Exploitation Stage:**
17. Determine the updated location of the x^{th} member of LOA by applying Eq. (8);
18. Modify the x^{th} member of the LOA by utilizing Eq. (9);
19. end (if)
20. $\boldsymbol{end\ for}$
21. Store the best solution obtained up to this point
22. $\boldsymbol{end\ for}$
23. Get back the optimal solution (i.e., optimal hyperparameter selection in GCN-GRU)
24. $\boldsymbol{end}$

3) Model Training

The O-GCN-GRU model for crime prediction is trained using the optimal hyperparameters summarized in Table 1. Furthermore, the obtained data can be used to correctly predict crime using the trained model.

4. RESULT AND DISCUSSION

1) Dataset Description

The dataset employed in this framework is 'Crime in India' [31], which provides comprehensive information on various categories of crimes that occurred in India from 2001 onwards. Multiple factors can be analyzed from this dataset, enabling individuals to gain deeper insights into India's crime statistics. It comprises 43 sections covering different types of crimes. Some data include district-level details such as police districts and special police units, which may vary from revenue districts. The majority of the records span the years 2001 to 2010, with certain files extending to 2011 and 2001–2014. For experimental analysis, four major crime classes are considered: 'ST_SCcrime,' 'Childcrime,' 'Womencrime,' and 'Theftcrime.' Additionally, image and video data related to crime terminologies from social media tweets corresponding to these four classes are collected. By associating each crime record with its respective image and video content, a total of 9,794 crime instances are obtained for experimentation.

2) Experimental Design and Evaluation Criteria

This part evaluates the O-GCN-GRU model performance, written in Python 3.11, against current methods including MOPSO [23], DSORT [28], MCN-CCO [25], SSO-BiLSTM [26] and GCN-GRU. The experiments were conducted on a system with Intel® Core™ i5-4210 CPU @ 3GHz, 4GB RAM as well as 1TB HDD with windows 10 (64-bit). The proposed and the baseline models were tested with the help of the datasets described in Section 4.1. After the data processing, 9,794 samples were found, where 7,834 were used as training data and 1,960 as testing data, after an 80:20 split. Fig. 4 presents the confusion matrix of the proposed model.

The following performance indicators are used to evaluate the model's crime prediction capabilities.

Accuracy: It is determined by dividing the number of occurrences for which predictions were accurate by the total number of cases.

$$Accuracy = \frac{TP + TN}{TP+TN+FP + FN} \tag{10}$$

A True Positive (TP) occurs when the model accurately labels an occurrence as a criminal offense. The model mistakenly labels a non-criminal occurrence as criminal, a phenomenon known as a False Positive (FP). When the model accurately labels an occurrence as not involving criminal activity, it is said to be a True Negative (TN). An example of a False Negative (FN) is when a criminal occurrence is misclassified by the model.

Precision: This metric quantifies the proportion of the model's positive predictions that are TP forecasts, or properly predicted crime events.

$$Precision = \frac{TP}{TP+FP} \tag{11}$$

Recall: It finds the fraction of positive cases in the dataset that were really predicted to be positive.

$$Recall = \frac{TP}{TP+FN} \tag{12}$$

F1-score: It strikes a balance between recall and precision.

$$F1 - score = \frac{2\times Precision \times Recall}{Precision+Recall} \tag{13}$$

Fig.5 compares the performance of various crime prediction models using the Crime in India dataset. The precision of O-GCN-GRU model is 6.61%, 5.56%, 4.65%, 3.20%, and 1.68% higher over the MOPSO, DSORT, MCN-CCO, SSO-BiLSTM, and GCN-GRU models, respectively. In terms of recall, O-GCN-GRU shows improvements of 7.21%, 6.15%, 5.11%, 3.87%, and 2.22% over the same models. Additionally, the F1-score of O-GCN-GRU is higher by 6.60%, 5.44%, 4.53%, 3.19%, and 1.57% over the same models, respectively. These enhancements are due to an efficient optimization of hyperparameter in GCN-GRU enabling effective feature extraction and sequential pattern learning in crime prediction.

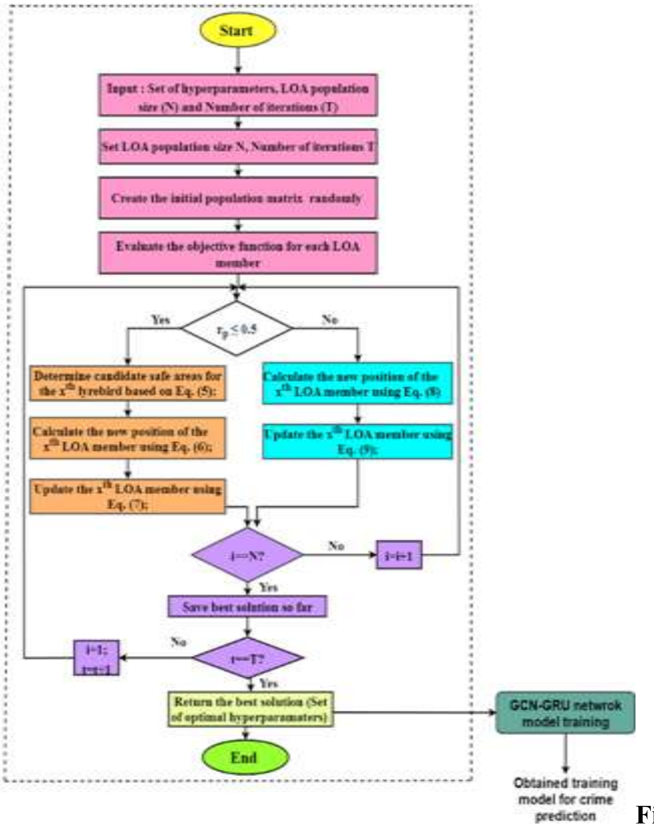

Fig -3: Flow Diagram of O-GCN-GRU Model based on Lyrebird Optimizer

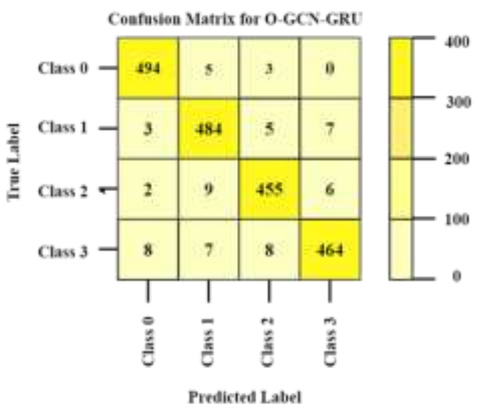

Fig -4: Confusion Matrix for the proposed model

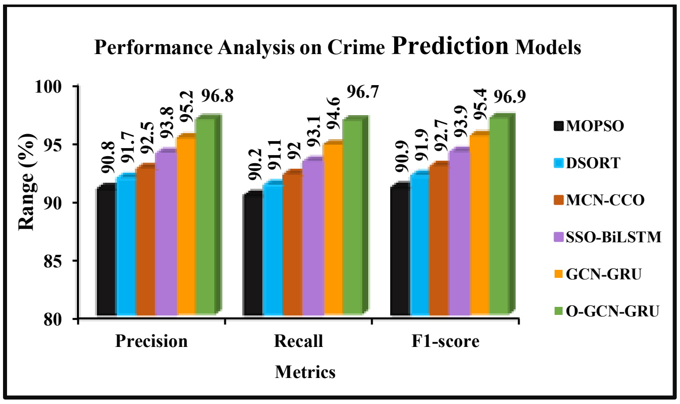

Fig -5: Performance Analysis of Different Crime Prediction Models

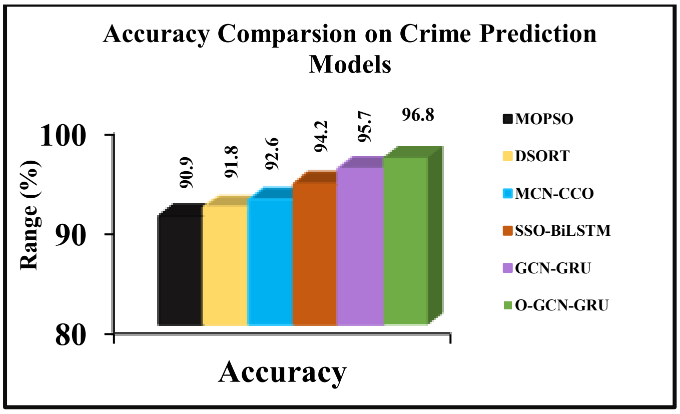

Fig -6: Accuracy Analysis of Different Crime Prediction Models on Collected Dataset

Fig. 6 illustrates the accuracy of various models evaluated on the crime data prediction dataset. The O-GCN-GRU model achieves accuracy that is 6.49% higher than MOPSO, 5.45% higher than DSORT, 4.54% higher than MCN-CCO, 2.76% higher than SSO-BiLSTM, and 1.15% higher than GCN-GRU. This enhancement is due to which the O-GCN-GRU model maximizes the prediction accuracy compared to the other models by optimizing the model hyperparameters to learn more complex features from the historical and social media information to predict crime intensities precisely.

5. CONCLUSION

This paper suggests O-GCN-GRU model to optimize the hyperparameters of the GCN-GRU network to maximize the performance of the crime prediction and minimum computational cost. The model utilizes the LOA, which is based on how the lyrebirds behave in the wild in reaction to threats. The LOA occurs in two phases: exploration and exploitation, where the first stage is the simulation of an escape strategy by the lyrebirds to provide a wide search of potential solutions and the second is the refinement of the search in promising areas, when the exploitation stage occurs. According to these steps, optimal hyperparameters of the GCN-GRU model when predicting crime using crime data are identified successfully. Through the application of LOA, the hyperparameters of the GCN-GRU network are refined thus giving high accuracy and low complexity. Lastly, the experimental findings suggest that the O-GCN-GRU model performs better than other currently existing crime prediction models with accuracy of 96.8% on the crime in Indian dataset.

REFERENCES

1. Chen, H., Chung, W., Xu, J.J., Wang, G., Qin, Y., Chau, M.: Crime data mining: a general framework and some examples. Computer 37(4) (2004) 50–56
2. Polvi, N., Looman, T., Humphries, C., Pease, K.: The time course of repeat burglary victimization. Br. J. Criminol. 31(4) (1991) 411–414
3. Sherman, L.W., Gartin, P.R., Buerger, M.E.: Hot spots of predatory crime: routine activities and the criminology of place. Criminology 27(1) (1989) 27–56
4. Farrell, G., Pease, K.: Once bitten, twice bitten: repeat victimisation and its implications for crime prevention. Home Office Police Research Group, London (1993)

5. Bowers, K.J., Johnson, S.D., Pease, K.: Prospective hot-spotting: the future of crime mapping. Br. J. Criminol. 44(5) (2004) 641–658
6. Xu, C., Liu, L., Zhou, S.H.: The comparison of predictive accuracy of crime hotspot density maps with the consideration of the near similarity: a case study of robberies at DP peninsula. Sci. Geogr. Sin. 36(1) (2016) 55–62
7. Mohler, G.O., Short, M.B., Brantingham, P.J., Schoenberg, F.P., Tita, G.E.: Self-exciting point process modeling of crime. J. Am. Stat. Assoc. 106(493) (2011) 100–108
8. Alves, L.G.A., Ribeiro, H.V., Rodrigues, F.A.: Crime prediction through urban metrics and statistical learning. Physica A 505 (2018) 435–443
9. Wang, B.H., Patel, B., Rudin, C.: In pursuit of interpretable, fair and accurate machine learning for criminal recidivism prediction. J. Quant. Criminol. 39(2) (2022) 519–558
10. Liu, L., et al.: Comparison of random forest algorithm and space-time kernel density mapping for crime hotspot prediction. Prog. Phys. Geogr. 37(6) (2018) 761–771
11. Yan, Z., Chen, H., Dong, X., Zhou, K., Xu, Z.: Research on prediction of multi-class theft crimes by an optimized decomposition and fusion method based on XGBoost. Expert Syst. Appl. 207 (2022) 117943
12. Shi, T., Fu, J., Hu, X.: TSE-Tran: prediction method of telecommunication-network fraud crime based on time series representation and transformer. J. Saf. Sci. Resil. 4(4) (2023) 340–347
13. Lu, X., et al.: The prediction of PM2.5 concentration using transfer learning based on ADGRU. Water Air Soil Pollut. 234(4) (2023) 58
14. Yan, J., Hou, M.: Predicting time series of theft crimes based on LSTM network. Data Anal. Knowl. Discov. 4(11) (2020) 84–91
15. Cortez, B., Carrera, B., Kim, Y.J., Jung, J.Y.: An architecture for emergency event prediction using LSTM recurrent neural networks. Expert Syst. Appl. 97 (2018) 315–325
16. Duan, L., Hu, T., Cheng, E., Zhu, J., Gao, C.: Deep convolutional neural networks for spatiotemporal crime prediction. In: Proc. Int. Conf. Inf. Knowl. Eng. (IKE), WorldComp (2017) 61–67
17. Yi, F., Yu, Z., Zhuang, F., Guo, B.: Neural network based continuous conditional random field for fine-grained crime prediction. In: Proc. Int. Joint Conf. Artif. Intell. (IJCAI) (2019) 4157–4163
18. Dong, Q., Li, Y., Zheng, Z., Wang, X., Li, G.: ST3DNetCrime: improved ST-3DNet model for crime prediction at fine spatial temporal scales. ISPRS Int. J. Geo-Inf. 11(10) (2022) 529
19. Jiang, H., Cao, P., Xu, M., Yang, J., Zaiane, O.: Hi-GCN: a hierarchical graph convolution network for graph embedding learning of brain network and brain disorders prediction. Comput. Biol. Med. 127 (2020) 104096
20. Liang, W., Wang, Y., Tao, H., Gao, J.: Towards hour-level crime prediction: a neural attentive framework with spatial–temporal-categorical fusion. Neurocomputing 486 (2022) 286–297
21. Jin, G., Liu, C., Xi, Z., Sha, H., Liu, Y., Huang, J.: Adaptive dual-view WaveNet for urban spatial–temporal event prediction. Inf. Sci. 588 (2022) 315–330
22. Shan, M., Ye, C., Chen, P., Peng, S.: Ada-GCNLSTM: an adaptive urban crime spatiotemporal prediction model. J. Saf. Sci. Resil. (2025)
23. Das, P., Das, A.K., Nayak, J., Pelusi, D., Ding, W.: Incremental classifier in crime prediction using bi-objective particle swarm optimization. Inf. Sci. 562 (2021) 279–303
24. Jeyaboopathiraja, J., Priscilla, G.M.: Optimized features and deep learning based crime trends prediction. ICTACT J. Soft Comput. 11(2) (2021)
25. Singuluri, P.K., Jayamma, B., Bharathi, B.D., Sree, T.B., Santhoshi, B.L.: Crime prediction using modified capsule network with criss-cross optimization on the sentiment analysis for cyber-security. In: Proc. Int. Conf. Mining Intell. Knowl. Explor. (Springer, Cham) (2023) 105–116
26. Mithoo, P., Kumar, M.: Social network analysis for crime rate detection using Spizella swarm optimization based BiLSTM classifier. Knowl.-Based Syst. 269 (2023) 110450
27. Waddenkery, N., Soma, S.: Adam-Dingo optimized deep maxout network-based video surveillance system for stealing crime detection. Meas. Sensors 29 (2023) 100885