scientific reports

OPEN

Lightweight deep learning model for crime pattern recognition based on transformer with simulated annealing sparsity and CNN

HongYuan Lu, ChengXin Chen, YuQi Ma & YanMing Ma

This study addresses the pressing need for high efficiency and low resource consumption in crime pattern recognition within public safety governance by proposing a lightweight deep learning model known as the lightweight crime recognition network (LCRNet). Designed to provide intelligent support for forecasting and case classification, LCRNet integrates a Transformer encoder and convolutional neural network. To optimize performance, the model introduces simulated annealing sparsity (SAS) into the multi-head self-attention of the Transformer architecture, thus effectively reducing computational overhead while maintaining accuracy. Experimental results indicate that LCRNet achieves an accuracy of 97.76% on real-world crime data from Los Angeles and demonstrates strong generalizability in cross-dataset testing. Additionally, ablation studies and visualizations of the sparsity process confirm the effectiveness of SAS. This research provides a practical solution for efficient crime pattern recognition and edge device deployment in public safety, and our future work will focus on enhancing model interpretability and validating adaptability in resource-constrained environments.

Keywords Crime recognition, Lightweight model, Simulated annealing sparsity, Transformer, CNN

The rapid development of technology has brought unprecedented transformations, with the rise of AI[1] and data technologies[2] significantly boosting the efficiency of social functioning and governance[3]. However, this development has also intensified social tensions, increased the complexity of criminal activities, and led to the frequent emergence of new forms of crime. Consequently, the ability to swiftly and accurately identify crime patterns has become a critical challenge for public safety governance.

The application of machine learning and deep learning in crime analysis has proven to be an effective approach[4–6]. In fact, machine learning is a branch of AI that enables computers to learn from experience, allowing them to make predictions or decisions on unseen data[7]. Deep learning, as a subset of machine learning, utilizes backpropagation to optimize feature learning, delivering outstanding performance and enabling efficient modeling of complex problems[8].

In recent years, the Transformer architecture has achieved significant success in natural language processing (NLP), primarily due to its ability to efficiently capture long-term temporal dependencies and sequential patterns. When applied to crime data, the Transformer's multi-head attention enables parallel processing of crime event sequences[9], making it particularly effective in identifying seasonal and periodic trends. In contrast, convolutional neural networks (CNNs) are well-suited for extracting local features—studies have shown that CNNs can effectively learn patterns in localized areas with high crime incidence[10]. However, CNNs alone may fall short of capturing global features.

Based on this, researchers have proposed combining convolutional neural networks (CNNs) with the Transformer architecture to further enhance feature extraction[11]. This technique has been widely applied to models of crime prediction, accurately predicting both long-term and short-term crime patterns[12].

However, traditional Transformer models come with many parameters, making them difficult to deploy directly on resource-constrained devices[13,14]. In practical applications, these models are also characterized by large model sizes, high computational complexity, and difficulties in deployment when resources are constrained. Nevertheless, optimizing Transformers through sparsification is an effective approach. Current sparsification strategies often focus on sparsifying information content, such as optimizing model efficiency through sliding

School of National Security, People's Public Security University of China, Beijing 100038, China. email: lhy20000201@163.com

window attention[15,16] or dynamic sparse attention[17]. Furthermore, some researchers have demonstrated that using reactivatable pruning of redundant connections can facilitate obtaining lightweight CNNs[18]. Inspired by these findings and incorporating the idea of extracting crime clues from global to local perspectives used by law enforcement personnel during crime analysis[19], we design a new mechanism that achieves simulated annealing sparsity (SAS) in the attention of Transformer.

This study aims to balance model accuracy and lightweight design in crime pattern recognition. Traditional models often struggle to reconcile accuracy with the constraint of computational resources. Based on the Transformer-CNN architecture, this study incorporates the SAS mechanism to design an efficient model specifically tailored for crime pattern recognition. Unlike existing sparsity mechanisms, our model employs annealing during the training of Transformer and enables progressive sparsification while maintaining randomness. This approach is then combined with parallel shallow CNN to classify various crime patterns. The specific contributions are as follows:

(1) We designed a lightweight crime recognition network (LCRNet) model, which combines the global modeling of Transformer with the local feature abstraction of shallow CNN. This integration enables robust feature extraction using fewer computational resources and parameters and offers a feasible approach for crime pattern recognition in resource-limited conditions within public safety governance.

(2) We investigated the application of TSAS in crime pattern recognition and used the number of epochs as an adjustment strategy for sparsifying connections in the multi-head self-attention (MHSA) mechanism. Experimental results demonstrated a significant reduction in the focus window of attention matrix within 20 epochs. The model optimized Transformer's self-attention connections in a short time. Moreover, the ablation study shows that the SAS sparsity mechanism reduces the model's FLOPs by 36.24% while maintaining an accuracy of 97.75%. In other words, it achieves a 36.24% reduction in computational overhead with only a marginal 0.5% decrease in accuracy. This method achieves, for the first time in crime recognition, an effective balance between lightweight architecture and high accuracy.

(3) Comprehensive testing was constructed, and our results indicated that LCRNet achieved 97.76% accuracy and a 97.75% F1 score on the crime data of Los Angeles. In tests using three other crime datasets, it attained accuracies of 98.14%, 93.52%, and 84.75%. Additionally, introducing SAS reduced the model's FLOPs (floating point operations) by 36.24%. These results demonstrated the accuracy, generalizability, and efficiency of this model, making a significant contribution to the application of deep learning in crime analysis.

Methods

In Fig. 1, we briefly illustrate the basic structure of LCRNet proposed in this study. The model consists of the following components: a shallow small-kernel CNN focusing on local feature extraction, a Transformer encoder using a SAS mechanism, the component for concatenating global and local features, and a classifier for outputting classification results.

Overall, LCRNet comprises three components: dual-branch parallel feature extraction, feature concatenation, and classification output.

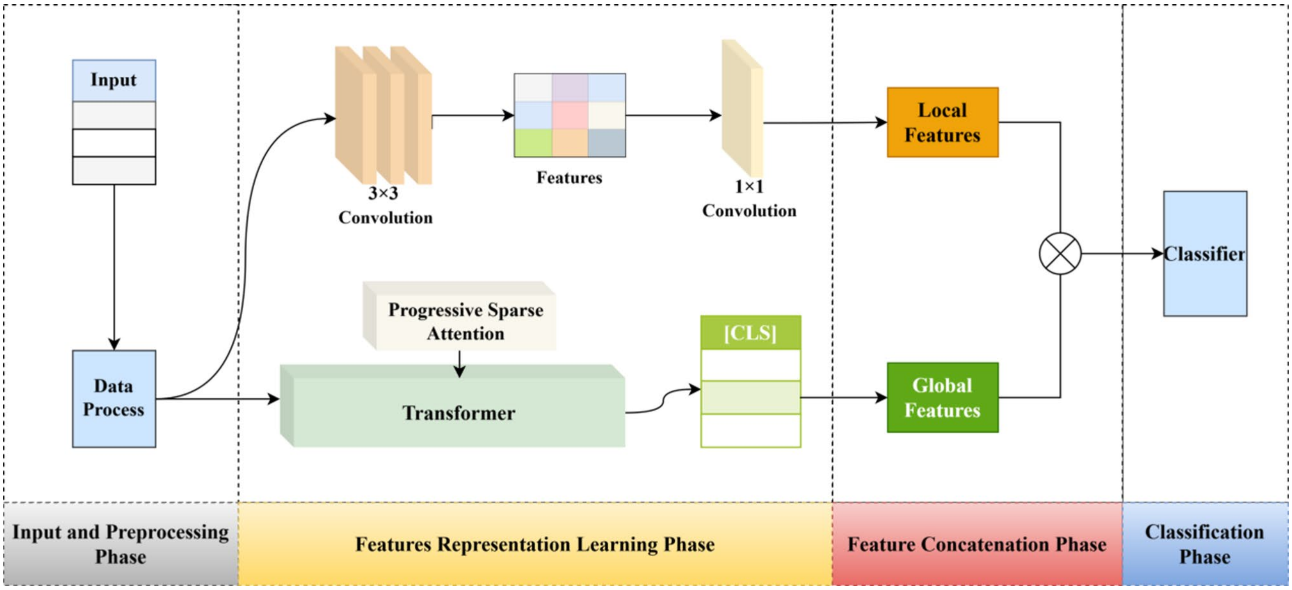

Fig. 1. Overview of the proposed model structure.

Data preprocessing

For data containing large amounts of crime information as a dataset, the first step is data preprocessing. There are mainly three types of features that need to be processed: numerical, free text, and categorical features. Through these steps, the data is preprocessed into numerical features so that it can be smoothly input into models.

For numerical fields in the raw data, we performed L2 normalization to ensure that numerical features are on the same scale. This is mathematically expressed as:

$$\mathrm{X} = \frac{\mathrm{X}}{\|\mathrm{X}\|_2}$$

Where X represents the raw data (x1, x2, x3, x4, … , Xn), and $\|\mathrm{X}\|_2$ represents the square root of the sum of squares of this data. Through L2 normalization, the numerical direction remains unchanged while its length is scaled to 1, avoiding interference in subsequent model training due to scale differences between different features.

For preprocessing categorical data such as gender, region, and occupation, as well as label columns in the dataset, we use embedding encoding to map them to a dense vector space, i.e.,

$$\mathrm{C_i} = \mathrm{Embedding}\left(\mathrm{C_i}\right), \;\; \mathrm{C_i} \in \mathbb{R}^{\mathrm{d_{embedding}}}$$

For free-text data $\mathrm{T} = \{\mathrm{t_1}, \mathrm{t_2}, \mathrm{t_3}, \ldots, \mathrm{t_n}\}$, normalization, data cleaning, and noise removal are conducted first. This includes methods such as case conversion, spelling correction, lemmatization, and stopword removal. Then, a Tokenizer is used to convert the normalized text into a tokenized sequence:

$$\mathrm{X_i} = \mathrm{Tokenizer}\left(\mathrm{t_i^{'}}\right), \quad \mathrm{X_i} \in \mathbb{R}^{\mathrm{L_i} \times \mathrm{d_{token}}}$$

Where $\mathrm{L_i}$ represents the length of the tokenized sequence, and $\mathrm{d_{token}}$ represents the dimensionality of each word vector. Finally, a pre-trained BERT model is used to convert the obtained tokenized sequence into embedding vectors. This step is expressed as:

$$\mathrm{E_i} = \mathrm{BERT}\left(\mathrm{X_i}\right), \mathrm{X_i} \in \mathbb{R}^{\mathrm{L} \times \mathrm{d_{embedding}}}$$

During data preprocessing, addressing outliers and missing values is crucial. For anomalies and missing data in input features, we employed tailored strategies based on the type of feature sequence. First, the case number, which uniquely identifies each case, is used to filter out any records with missing values. If duplicate case numbers are detected, they are considered anomalies requiring verification, followed by deletion or consolidation of the corresponding records. Second, for temporal information, missing values are imputed using the respective time-related feature within the same crime type. Lastly, for text features, missing entries are uniformly replaced with the label “Unknown” to maintain consistency. Additionally, the feature of victim age, which requires special handling, is imputed using the median and then truncated between the 0th and 100th percentiles to minimize the impact of outliers. This hierarchical processing approach takes into account both the specific characteristics of each type of data and ensures consistency and standardization throughout the dataset.

Data input into the Transformer typically requires no processing and can be directly input as embeddings. However, data input into the shallow small-kernel convolutional layers requires further format alignment. To achieve this step more efficiently and quickly, we perform format conversion through dimension transposition and dimensionality expansion. This is expressed as:

$$\mathrm{E}^{'} = \left(\mathrm{Permute}\left(\mathrm{E_i}, \mathrm{dim}\left(0, 2, 1\right)\right), \mathrm{dim} = -1\right)$$

Shallow small-kernel CNN

This study employs shallow small-kernel CNN to design an efficient local feature extraction network comprising three 1D convolutional layers (each followed by a ReLU activation function): First layer (3 × 3 kernel, 64 output channels), Second layer (3 × 3 kernel, 128 output channels), and Third layer (3 × 3 kernel, 256 output channels). Small kernels enhance the receptive field while progressively increasing channel depth enables the extraction of abstract and complex features, with padding = 1 ensuring consistent output sequence lengths across layers.

Transformer with SAS mechanism

This component uses a standard Transformer encoder as the feature extractor for global dependency. We introduce a simulated annealing sparsity mechanism. In essence, the model begins with full attention connectivity and gradually narrows the attention window as training progresses. This approach mirrors the principles of simulated annealing: during the “high-temperature” phase, the model conducts a comprehensive exploration, while in the “low-temperature” phase, it shifts focus to the most relevant ones. To avoid premature convergence to suboptimal solutions, a “reheating process” is incorporated, thus reintroducing controlled randomness and helping the model escape local optima.

During the initial training stage, it employs full self-attention and gradually reduces attention window as model performance improves. Specifically, the model initially adopts a full attention mechanism to ensure global feature extraction, then progressively sparsifies attention connections while allowing the model to stochastically focus on less critical regions. This approach achieves computational efficiency by reducing overhead while maintaining accuracy.

As shown in Fig. 2, this component consists of a standard 6-layer Transformer encoder, where each layer includes MHSA, relative position encoding, feed-forward neural networks, residual connections, and layer normalization. The SAS mechanism is applied to each layer. Specifically, we modify the MHSA in each layer of the Transformer encoder. Through Top-K selection and probabilistic sampling, this design mimics the progressive strategy and stochastic effects of annealing, enabling self-attention to preserve critical information while achieving time-step-dependent sparsity during training.

Figure 3 illustrates the detailed process of constructing the sparse mask and performing matrix fusion within the SAS mechanism. In brief, the approach begins with the self-attention matrix, where a Top-K operation is applied to retain the most important connections at each time step, thereby forming a Top-K set. Specifically, for each query position (i.e., each row in the attention matrix), the K positions with the highest weights are selected. In addition, a probability distribution is constructed from the remaining (non-Top-K) attention weights, from which random sampling is performed. This step mirrors the "random perturbation" in simulated annealing to help the model avoid focusing exclusively on high-weight connections. Based on these selections, a sparse mask matrix is constructed and applied to the original attention matrix (via element-wise multiplication), thus resulting in a sparsified attention matrix. Finally, both the dense and sparse matrices are multiplied by a sparsity coefficient, which progressively strengthens the sparsity effect from the initial to the later stages of training. The specific formulation is as follows:

First, we define a hyperparameter β (aligned with the learning rate) to represent the decay rate. The deterministic retention of Top-K attention weights is expressed as:

$$\mathrm{k}\,(\mathrm{t}) = \mathrm{k_{max}} - (\mathrm{k_{max}} - \mathrm{k_{min}}) \cdot \left(1 - \mathrm{e}^{-\beta\,\mathrm{t}}\right)$$

Then, based on the probability distribution of remaining positions, random sampling is performed to explore connections with low-weight yet impactful associations, thus preventing the model from converging to local optima. The mathematical expression is as follows:

$$M_{ij}^{(l)}\,(t) = \mathbb{I}\left(j \in \mathrm{Top} - (k\,(t) - r)\left(A_{i:}^{(l)}\right)\right) + \mathbb{I}\left(j \in \mathrm{Sample}_r\left(P_{i:}^{(l)}\right)\right)$$

For the probability distribution of the remaining positions, a temperature coefficient τ is used to control the variability of the distribution. Additionally, the sum of the important connections identified by Top-K is calculated using the positions not selected by Top-K, with an emphasis on the less important regions. The mathematical expression is as follows:

$$P_{ij}^{(l)} = \frac{\exp\left(A_{ij}^{(l)}/\tau\right)}{\sum_{m \notin \mathrm{Top}-(k(t)-r)}\left(A_{im}^{(l)}/\tau\right)}$$

Define the sparsity hyperparameter α (t) as the complement of the sparsity decay factor k(t). As training progresses, α (t) increases linearly, blending the sparse mask M and original attention A to produce the final sparsified attention matrix as follows:

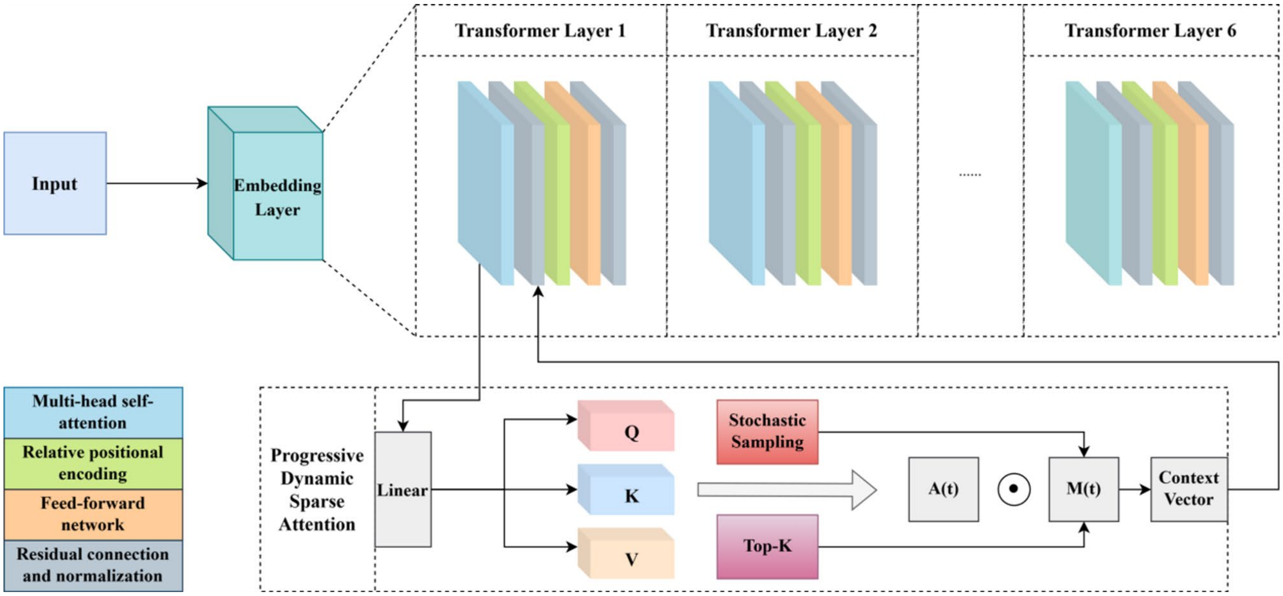

Fig. 2. Schematic diagram of the SAS mechanism.

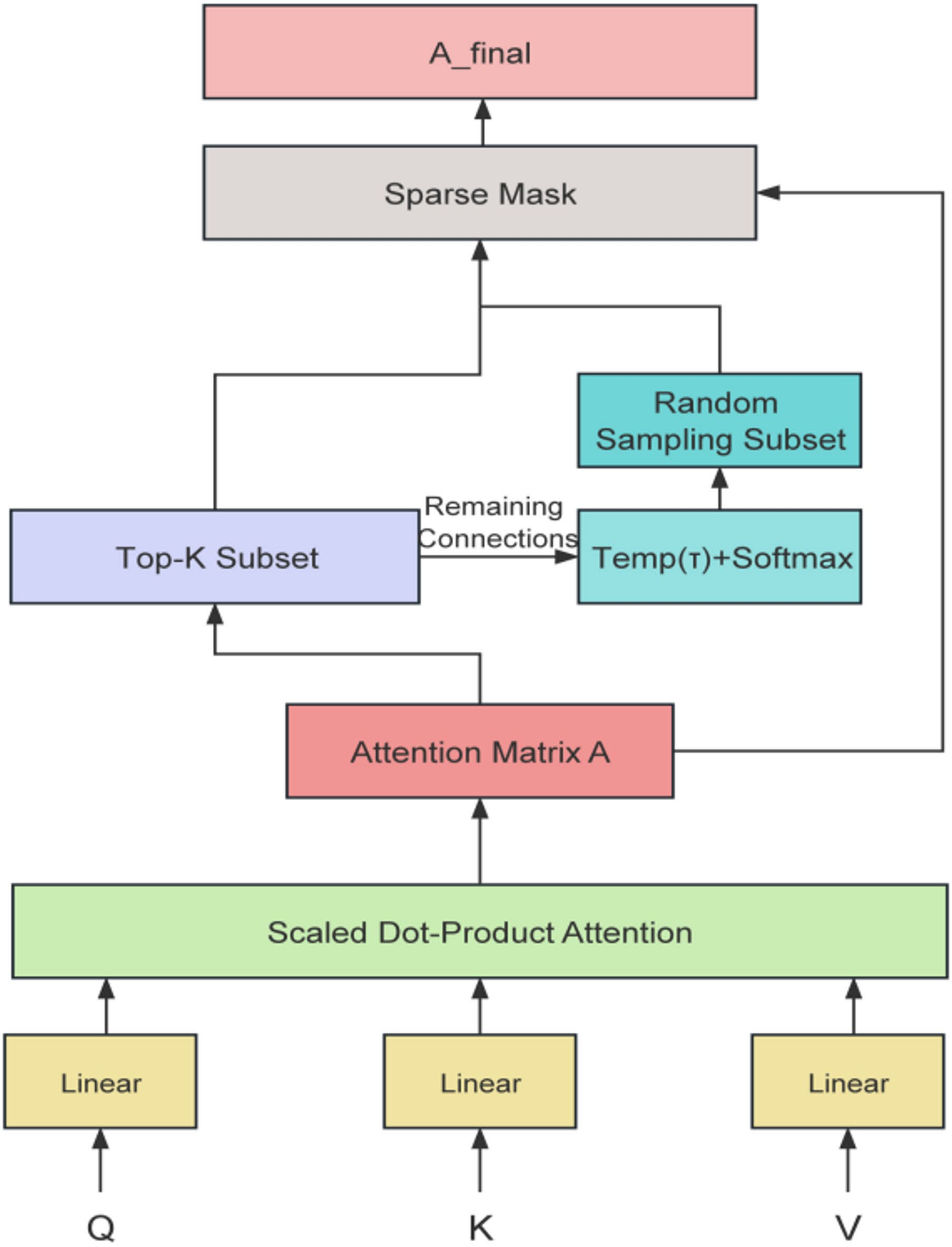

Fig. 3. SAS mechanism.

$$A_{\text{final}}^{(l)}(t) = \alpha(t) \cdot \left(M^{(l)}(t) \odot A^{(l)}\right) + (1 - \alpha(t)) \cdot A^{(l)}$$

Results

Dataset description

The crime data in Los Angeles was used in this study, comprising 185,716 samples. The dataset was randomly divided into training, testing, and validation sets in a ratio of 7:1.5:1.5. The original dataset contains 12 attributes:

	Epoch	Batch_size	Learning_rate	Optimizer	Sparsity decay beta
Hyperparameter	50	128	0.01	Adam	0.01

Table 1. Configuration of model hyperparameters.

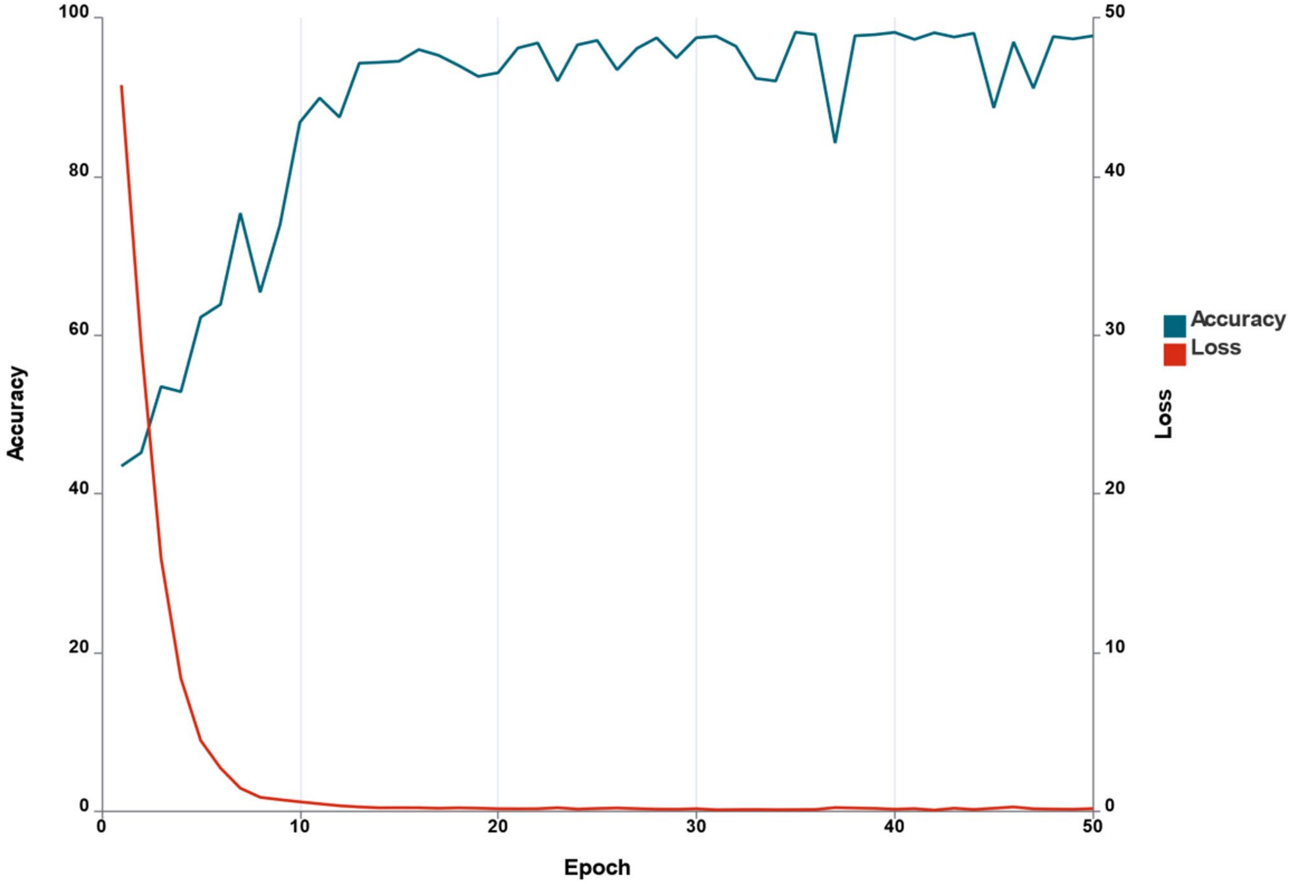

Fig. 4. Training loss and validation accuracy.

case number, report date, date of occurrence, time of occurrence, district/area, crime description, victim's age, gender, and race, the weapon used, case status, and specific location. To enhance the noise resistance of the model, we retained all attributes as feature columns for model inputs.

Experimental setup and hyperparameter configurations

The model architecture described in this paper was built using the PyTorch framework and a virtual environment is created via Conda. The experiments were conducted under the following conditions: Microsoft Windows 11 (version 10.0.22631) as the OS, Python 3.9.21, PyTorch 2.5.1, and CUDA 12.6. The hardware used for training the model included a 13th Gen Intel® Core™ i9-13900HX @ 2.20 GHz CPU, 32GB RAM, and an NVIDIA GeForce RTX 4080 Laptop GPU. All experimental data presented in this paper were obtained using this configuration.

The model described in this paper did not use any pre-existing weights from models trained on other datasets or tasks. Based on this, we explored various hyperparameter configurations and identified the optimal parameters. The detailed hyperparameter configuration is shown in Table 1. Specifically, we used the Adam optimizer during training, set the learning rate to 0.01, used a batch size of 128, and set the decay rate for SAS introduced in the Transformer architecture to match the learning rate at 0.01. The total training time was 50 epochs. This parameter setting was applied to all experiments in this study.

Loss curve

Figure 4 shows the loss and validation accuracy over the 50 epochs of training, where the training performance of the model is effectively monitored. The model's performance gradually improved as training progressed. At around the 15th epoch, the model began to converge, where loss and accuracy no longer exhibited significant changes and ultimately reached optimal performance. Furthermore, the model exhibited faster convergence during training, which implies a significant reduction in computation time and cost, thus further aligning with the goals of lightweight design.

Model evaluation

Misclassification is a critical issue for the model, particularly in highly sensitive law enforcement tasks. For instance, if the model frequently misclassifies one crime type as another, it may indicate significant difficulties in distinguishing between these crime patterns. This study employs a confusion matrix to illustrate misclassification scenarios of this model. In the matrix below, rows represent actual classifications, columns represent model predictions, diagonal elements indicate samples where actual and predicted classifications match, and off-diagonal elements represent samples where predictions deviate from reality. The specific confusion matrix is shown in Fig. 5.

Overall, the model demonstrates excellent performance in crime pattern recognition and classification tasks, and it accurately identifies all categories to a high degree. However, minor misjudgments persist in some categories, primarily between grand theft and vandalism and between vandalism and violent crime. These misclassifications are primarily due to similar attributes and contextual descriptions among these crime patterns. For instance, major theft and vandalism may occur under similar circumstances, and some instances of deliberate vandalism involving confrontational behavior can also include elements of violence. Nonetheless, these misclassifications are relatively rare, which still demonstrates the overall robustness and strong performance of the model.

In addition, the confusion matrix directly shows the model performance in each category. With the sample data of actual classifications versus predicted classifications, we can calculate four evaluation metrics: accuracy, precision, recall, and F1 score to assess the model. The evaluation results are presented in Table 2.

Accuracy: Accuracy is an important metric used to measure the reliability of the model directly. It illustrates the overall accuracy by calculating the percentage of correctly predicted samples among all samples.

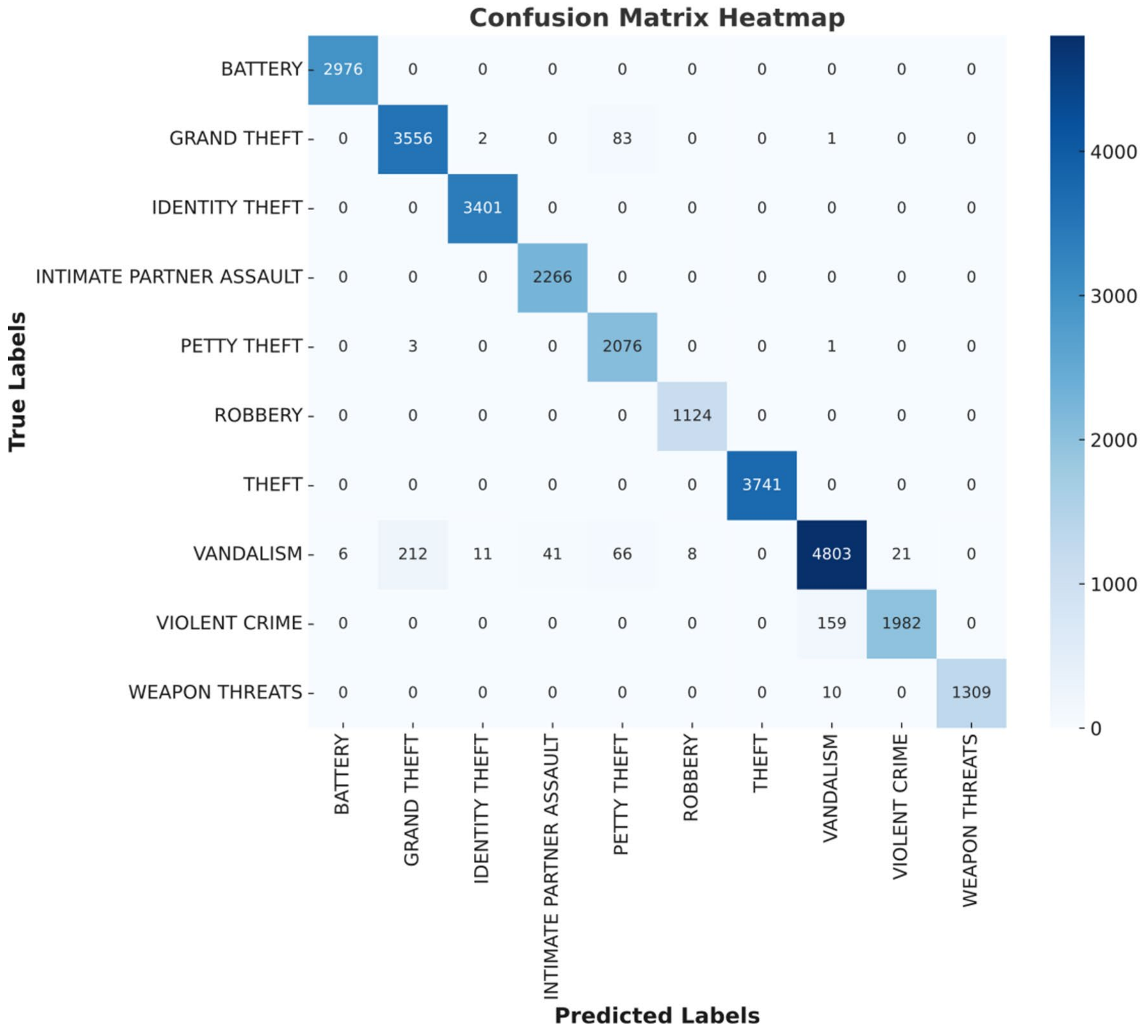

Fig. 5. Confusion matrix of the model.

Method	Accuracy	Precision	Recall	F1 score
LCRNet	97.76	97.76	97.76	97.75

Table 2. Results of model evaluation.

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

Precision: This metric represents the proportion of actual positive samples among all samples predicted as positive by the model. In other words, it determines whether the model incorrectly predicts negative samples as positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: This metric indicates the ratio of samples predicted as positive by the model among all actual positive samples.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score: This metric is essentially the harmonic mean of precision and recall. It effectively balances the conflict between true positives and false positives, providing a relatively stable measure for model evaluation.

$$\text{F1 Score} = 2 \times \frac{Precision \times\ Recall}{Precision + Recall}$$

Sparsified attention

The SAS mechanism described in this paper operates during each epoch of training by selecting important attention connections using Top-K based on attention weights while allowing some “ignored” connections to recover through random distribution sampling. The visualization of sparsity annealing introduced in this paper is shown in Fig. 6. We can observe that α(t) shows a monotonically increasing trend as the epoch of training increases. This indicates that the model effectively adjusts the mixture weights of sparse masks and model attention, with model attention gradually becoming sparser. This aligns with the intention of exploration in the early stages of training and convergence in the later stages, thus contributing to improved computational efficiency and generalization ability of the model.

Simulated annealing allows the model to reconnect previously “abandoned” parts after sparsifying attention connections. We verified the specific effects of simulated annealing by comparing fully connected, fixed attention connection, and Top-K sparse attention. The verification results are shown in Table 3 below. Experimental results indicate that when resources are constrained, SAS can effectively balance computational efficiency while achieving optimal performance.

Impact of temperature coefficient on the model

Hyperparameters significantly affect model training. The SAS mechanism introduces a hyperparameter, the annealing temperature τ, which controls randomness. The impact of the annealing temperature impact is illustrated in the bar chart of Fig. 7 below. It can be seen that a learning rate of 0.01 and an annealing temperature of 1.5 enable the model to achieve optimal performance.

Ablation study

To verify that each component of the model introduced in this paper serves its intended purpose, we designed an ablation study. This part of the study uses LCRNet as the baseline model and networks with various components removed as ablation models. By comparing the potential performance decline after removing components, we generated the data shown in Table 4. For clarity and convenience, we named the component without the SAS mechanism as “TCNN,” the component without shallow CNN as “TSAS,” and the model retaining only the Transformer encoder structure as “Transformer.”

The Transformer encoder structure with SAS mechanism and shallow CNN used in this study demonstrated strong potential for both model compression and accuracy improvement. Shallow CNNs effectively enhance the model’s ability to learn local features, with TCNN improving accuracy by 7.2% compared to standard Transformer architectures. Removing only the CNN component from LCRNet results in a 7.4% drop in accuracy, which underscores the critical role of shallow convolutional layers in processing input sequences. Many crime types share highly similar constitutive elements and differ only in specific descriptions or details. Consequently, the shallow convolution plays a critical role in distinguishing between crime patterns with fine-grained differences. For example, both Petty Theft and Grand Theft fall under the broader category of Theft but are often distinguished by particular keywords or phrase combinations. Shallow convolutions are well-suited to capturing these localized patterns, thereby enhancing the model’s sensitivity to fine-grained differences.

Moreover, TSAS reduced FLOPs by 38.45% compared to standard Transformer architectures. Compared to TCNN (the model without the SAS mechanism), LCRNet reduces FLOPs by 36.24%, indicating that the SAS

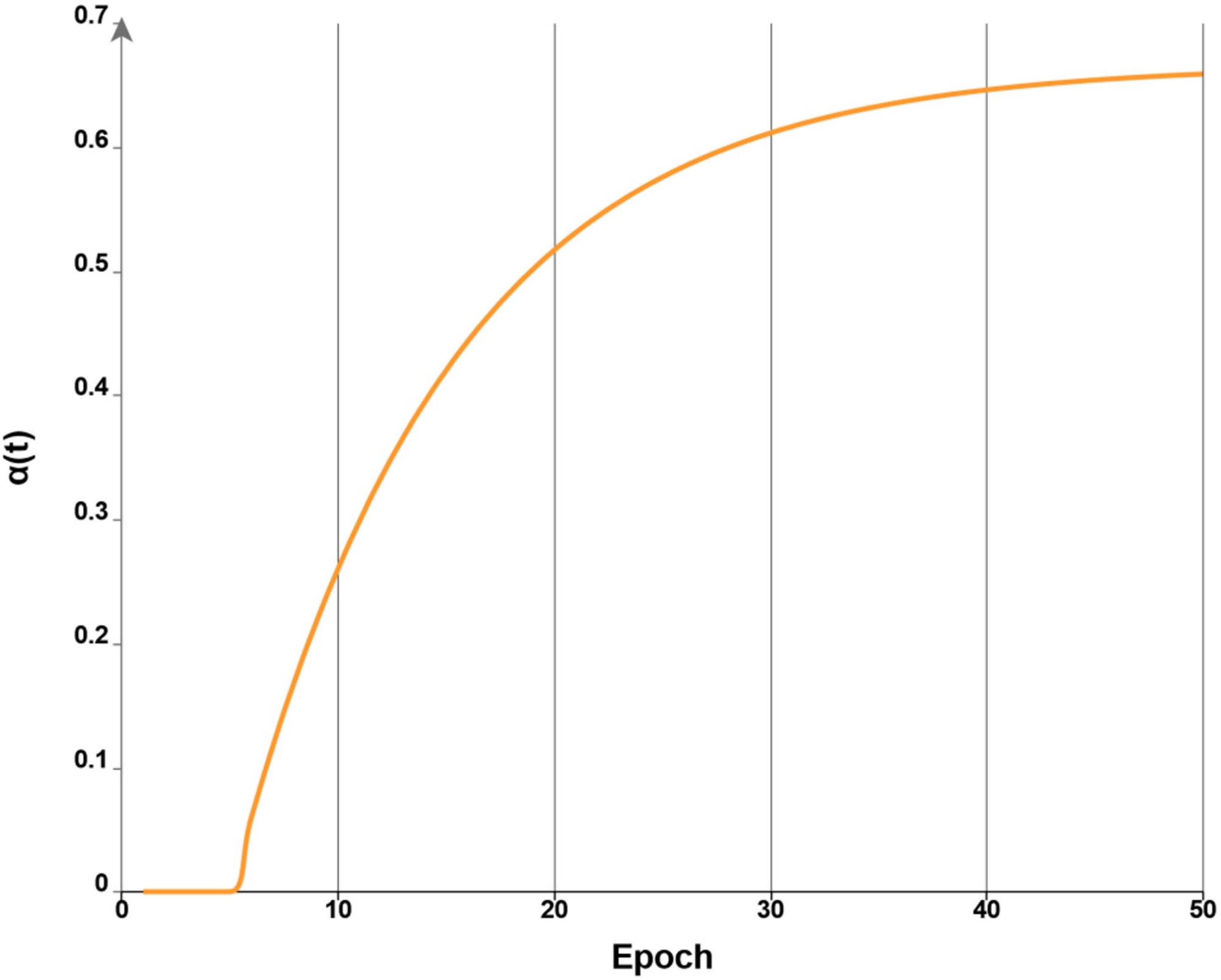

Fig. 6. SAS for attention connections.

Method	Accuracy	F1 score	FLOPs
Fully connected	91	90.8	511.51
Fixed attention connection (k = 6)	88.9	88.1	286.74
Progressive Top-K	87.4	87.2	307.98
TSAS	90.3	90.2	314.83

Table 3. Comparison of models with different attention connections.

mechanism effectively lowers the computational complexity. Although eliminating SAS results in a 0.5% drop in accuracy, the substantial reduction in computational complexity demonstrates that LCRNet achieves an effective balance between performance and lightweight design.

In summary, the components exhibit a clear complementary relationship: the shallow CNN effectively extracts local features from crime data, while the SAS mechanism optimizes the Transformer's self-attention, thereby reducing computational complexity. Therefore, integrating these two components proves to be an effective strategy, as it accurately identifies crime patterns while maintaining a lightweight architecture suitable for practical deployment.

Generalizability test

In this section, we directly transfer the model trained on the LAPD Crime data for testing on the CrimeCast data, Homicide Reports, and Crimes in Chicago data to validate the model's generalizability. The accuracy across these datasets are shown in Fig. 8. The results clearly demonstrate that LCRNet exhibits outstanding performance

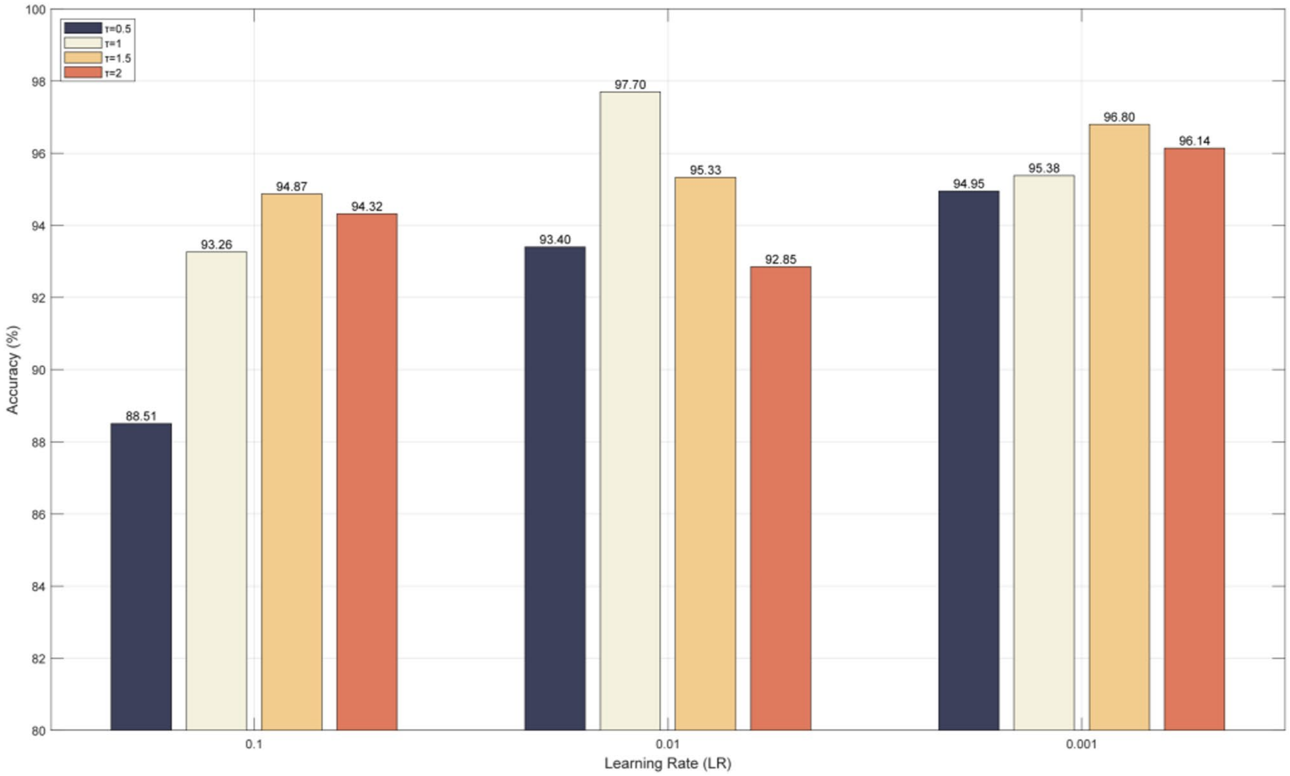

Fig. 7. Impact of the annealing temperature under different learning rates.

Method	Accuracy (%)	F1 score (%)	Parameter (M)	FLOPs (M)
LCRNet	97.8	97.8	2.16	348.94
TCNN	98.3	98.3	2.16	547.28
TSAS	90.4	90.3	1.97	314.83
Transformer	91.1	90.8	1.97	511.51

Table 4. Impact of different components on model performance.

across all datasets, specifically, it achieves 98.14% accuracy on the Crimes in Chicago data, 93.52% accuracy on the CrimeCast data, and 84.75% accuracy on the Homicide Reports.

However, we also observe significant variations in the accuracy across different datasets. For instance, there is a 13.39% accuracy difference between the Homicide Reports and Crimes in Chicago data, an 8.77% difference compared to the CrimeCast data, and a 13.01% difference relative to the LAPD Crime data.

We observed that the test results on the Homicide Reports dataset differ significantly from those on the other three datasets. Two possible factors may explain this discrepancy: (1) differences in feature distributions across datasets and (2) variations in dataset scale and distribution characteristics. Specifically, the Homicide Reports dataset is narrowly focused on violent crimes—namely homicides—which inherently limits the diversity of crime patterns. As a result, there is a significant mismatch between the subcategories of homicide cases and the broader class structure used during model traininininherently lacks a variety of crime patterns and is entirely focused on violent crimes of the homicide type, resulting in significant differences between the subcategories of homicides and the original class structure used for model training. g. Moreover, due to the distinct nature of homicide cases, the feature distribution in this dataset differs significantly from that of the other three datasets.

For example, in the Homicide Reports dataset, almost every entry in the "Weapon" feature column involves the use of weapons, and the proportion of homicides without weapon usage is extremely low. In contrast, in the other datasets, due to differences in crime patterns (such as theft or fraud), the use of weapons may not be a necessary condition. Additionally, when comparing homicide cases with more general crime data, the age distribution is an important indicator. To further investigate the model's performance on the Homicide Reports dataset, we plotted boxplots comparing victim age distributions between the Homicide Reports and the LAPD Crime Data. The Fig. 9. results indicate that the LAPD Crime Data exhibit a wider age distribution with a larger interquartile range, a wider variety of case types, and social backgrounds. In contrast, victim ages in the Homicide Reports are more concentrated, predominantly within the young to middle-aged demographic. In

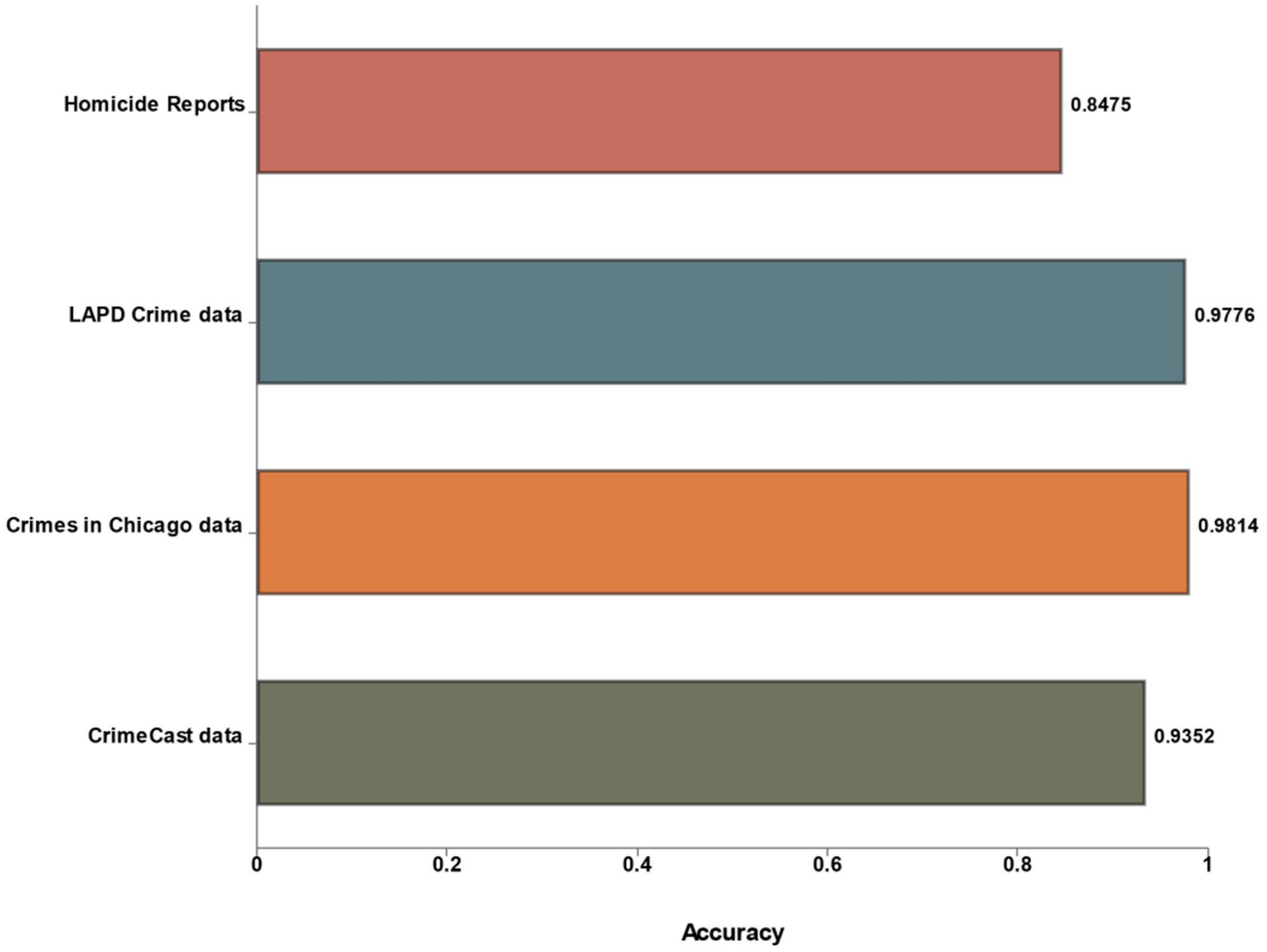

Fig. 8. Comparison of model accuracy across different datasets.

summary, the differences in feature distributions—particularly due to the specialized nature of homicide cases—are crucial in influencing the model's ability to recognize crime patterns accurately across datasets.

Comparison with lightweight techniques

We compared the performance of LCRNet with MobileVIT, DistilBERT, and Mobile-Former, which are all advanced lightweight variants based on the Transformer architecture. Furthermore, Fig. 10 illustrates the accuracy curves of different models during the training process. Results demonstrate that these three models achieved accuracies of 97.16%, 95.96%, and 96.29%, respectively on LAPD Crime data, further validating that LCRNet achieved optimal performance among different lightweight models.

LCRNet demonstrates distinct advantages when compared with the three models mentioned above. MobileVIT employs convolutional operations to extract feature maps and then applies a Transformer to further reduce spatial dimensions. However, this design may fail to capture certain fine-grained features. Moreover, MobileVIT is primarily designed for visual tasks and lacks optimization for processing long textual sequences. DistilBERT, on the other hand, is a lightweight Transformer-based model that reduces the parameter count compared to BERT (approximately 66 M parameters) but still remains significantly larger than LCRNet, which contains only 2.16 M parameters. Despite its smaller size, DistilBERT does not feature any specialized mechanisms for computational optimization. Mobile-Former, which parallels the structure of LCRNet by combining a mobile CNN with a Transformer, achieves an accuracy of 96.29% on the LAPD Crime dataset. However, it is evident that its convergence speed is substantially slower than that of LCRNet, thereby showcasing the performance benefits of the unique SAS design.

Comparison with other crime recognition models

This section compares LCRNet with other deep-learning models used for crime pattern recognition in existing studies. The results are shown in Table 5. Under current research conditions, LCRNet achieves relatively high accuracy in crime pattern recognition. Although its accuracy is 1.74% lower compared to the deeper network HO-ResNet-MA, LCRNet reduces parameter count by approximately 55.28%, indicating that LCRNet reduces complexity and enhances deployability.

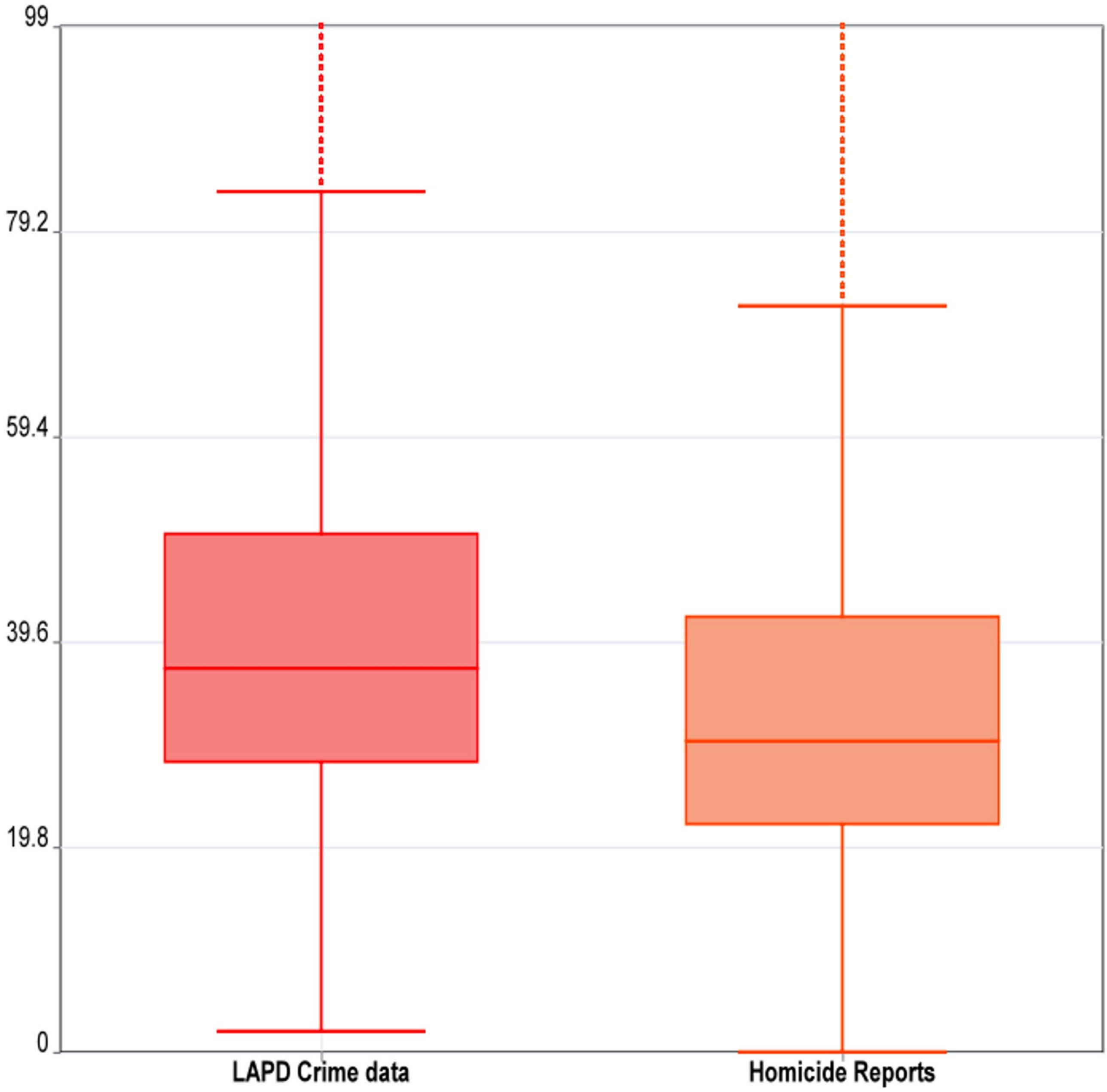

Fig. 9. Boxplot of the age distribution in LAPD Crime versus Homicide Reports.

It is important to note that limited research resources make it difficult to ensure all models were tested on identical datasets or under fully consistent experimental conditions. Also, most studies on crime recognition do not provide specific model parameters or FLOPs. Therefore, these comparisons should be treated as reference material rather than definitive evaluations.

Discussion

The method proposed in this study is designed to address the problems of computational complexity and deployment difficulties in traditional deep learning networks. It is a model specifically designed for crime pattern analysis and recognition in textual data.

To enhance inference speed and reduce deployment costs, we drew inspiration from the routine activity theory in criminology and incorporated the randomness of simulated annealing to avoid local optima. This led to the design of the SAS mechanism, which allows attention to focus on key clues from global information, similar to how law enforcement personnel analyze cases. Specifically, the model uses α(t) to adjust the mixture weights of sparse masks and attention matrices, gradually retaining important attention connections during the training while implementing model "annealing" for less important parts through temperature parameters

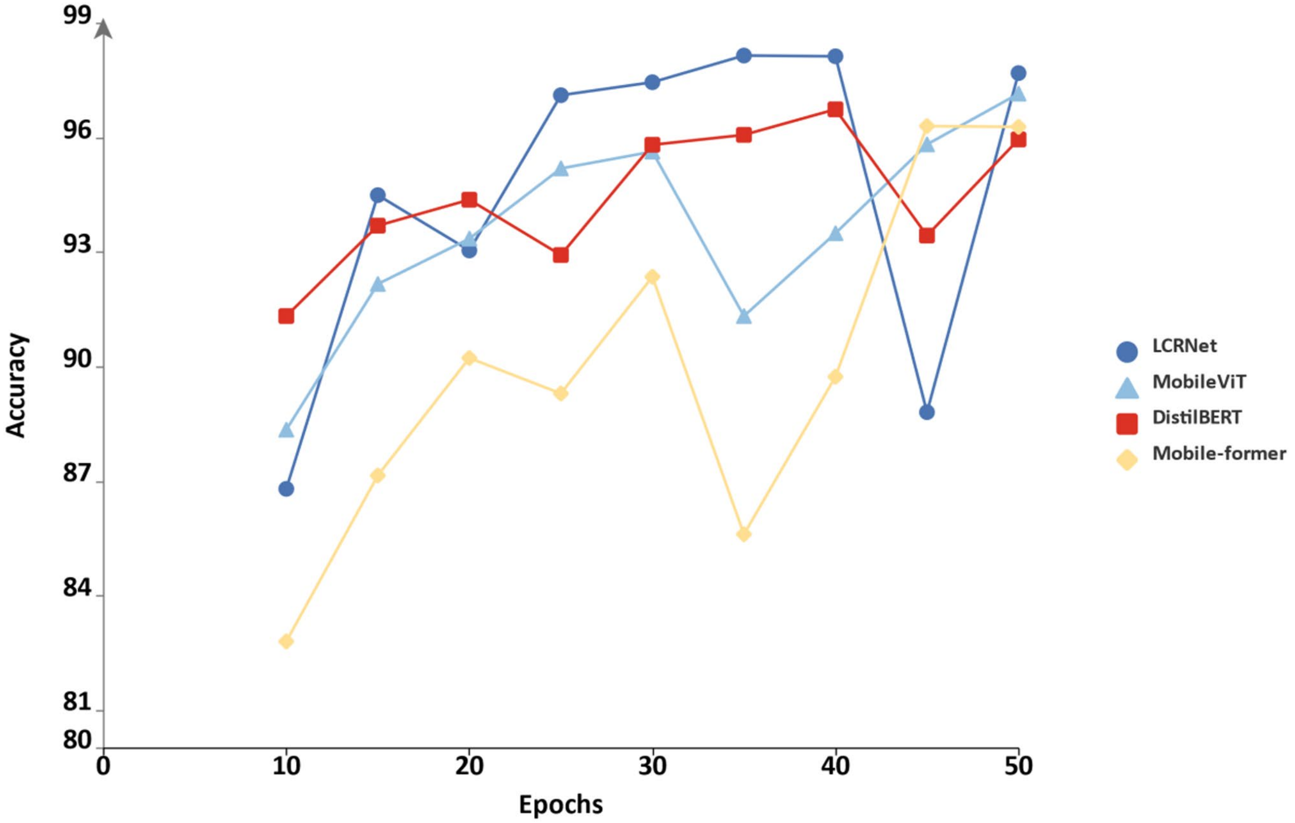

Fig. 10. Comparison with state-of-the-art lightweight models.

Literature	Method	Accuracy (%)	Parameter (M)
	LCRNet	97.76	2.16
Dawei Qiu et al.[20]	HO-ResNet-MA	99.5	4.83
	ContNet	99.4	9.42
	ResNet	99.5	4.03
Myung-Sun Baek et al.[21]	CNN-based prediction model	91	–
George Karystianis et al.[22]	LSTM-MLP	69	–
Wajiha Safa et al.[26]	ARIMA	94	–
Hyeon-Woo Kang et al.[23]	DNN	84.25	–

Table 5. Comparison with other crime recognition models.

to reduce model "oversights." Results confirm that this sparsity mechanism significantly reduces computational overhead in crime pattern recognition while maintaining accuracy.

The results of this study demonstrate the potential of lightweight models in public safety governance, achieving 97.76% accuracy in crime pattern recognition. This high accuracy not only represents the ability of lightweight models to effectively handle complex tasks but also demonstrates their potential in environments with limited real-time inference and deployment resources. It is noteworthy that this design enables LCRNet to be directly deployed on edge devices without relying on additional pruning or quantization. The model is inherently lightweight and adaptable, requiring only format conversion (e.g., to ONNX or TorchScript) and compliance with the target device's runtime environment. This approach not only streamlines deployment but also avoids the accuracy loss typically associated with pruning or quantization.

However, our experiments have only verified theoretical performance with the high-performance GPU and have not yet tested inference speed, memory usage, and related energy consumption on law enforcement terminals equipped with edge computing capabilities. Therefore, future work will focus on model deployment and optimization to validate its real-world performance.

In this study, we observed that a major source of model confusion is the similarity in attributes among certain crime incidents. Therefore, we emphasize incorporating discriminative features into crime datasets. Adding features that help differentiate categories—such as indicating whether bodily harm was involved or if a weapon was used during the crime—can be crucial for distinguishing between violent and non-violent cases.Furthermore, while the model performs well in generalizability tests, there are still significant differences in accuracy across different datasets. These differences may be amplified in practical applications due to data characteristics specific to certain regions or scenarios, making the model's performance unstable in the real world.

As technology continues to advance, lightweight deep learning will play an increasingly important role in public safety governance, especially in crime prediction, recognition, and analysis. Although most studies can confirm the contribution of deep learning to public safety, the "black box" nature of deep learning has always led decision-makers of criminal justice to maintain a skeptical attitude towards it[24,25]. Moreover, in practical applications where law enforcement personnel cannot fully trust the model, it may have negative effects[26]. Therefore, future research should focus on improving and enhancing the interpretability of models to increase trust among both the public and law enforcement personnel. Layer-wise relevance propagation (LRP) offers a practical method for interpreting model predictions. Specifically, when applied to LCRNet, LRP can identify the features that most significantly influence a given classification. For instance, in cases where a crime pattern is classified as fraud, LRP may reveal that specific descriptive phrases (e.g., "credit card") and the victim's age may be particularly relevant to the model's decision.

Conclusion

The ability to accurately and efficiently identify crime patterns is crucial in public safety governance. Precise identification of specific crime patterns can provide reliable data analysis in law enforcement while the lightweight nature and deployability of the model allow law enforcement personnel to dynamically accomplish this complex task in a short time. This not only reduces the occurrence of criminal incidents but also optimizes the allocation of police resources, avoiding waste caused by excessive investment of resources.

This paper introduces a lightweight deep-learning method for crime pattern recognition. We successfully integrated the SAS mechanism into the Transformer architecture, and through parallel shallow small-kernel CNN, we significantly reduced computational complexity while ensuring effective extraction of both local and global features by the model, thereby achieving cost reduction and efficiency improvement over traditional deep learning models. This model demonstrates powerful performance in crime pattern classification, achieving 97.76% accuracy while reducing FLOPs by 36.24%. Furthermore, test results on different datasets also demonstrate good generalizability of the model. It alleviates the challenges posed by poor generalization and limited deployment resources in practical applications of crime pattern recognition, making an important contribution to effective public safety governance.

Despite these advancements, the model still lacks interpretability during crime pattern recognition. To enhance trust in deep learning models among law enforcement agencies and the public, we plan to focus on model interpretability in the next phase of our work. We have noted that layer-wise relevance propagation can effectively interpret deep learning models, so we are expected to explore how to apply this mechanism to analyze the role of deep learning models in crime pattern recognition in future work. In addition, the next stage of research plans to deploy the model on resource-limited devices, such as the Jetson series and Google Coral series. This will aim to provide reliable support tools for law enforcement personnel in real-world applications, helping them make accurate and swift decisions and execute tasks efficiently.

Data availability

All datasets used in this study are openly accessible on Kaggle or GitHub: Homicide Reports: https://www.kaggle.com/datasets/murderaccountability/homicide-reportsCrimeCast data: https://kaggle.com/competitions/crime-cast-forecasting-crime-categoriesLAPD Crime data: https://github.com/mondher0/Analysing-crimes-in-Los-AngelesCrimes in Chicago data: https://www.kaggle.com/datasets/currie32/crimes-in-chicagoTo protect intellectual property rights and ensure research continuity, the model code is currently under restricted access. Interested researchers must submit a formal request to the corresponding author, which will be reviewed by the People's Public Security University of China. Approved requestors will be required to sign a non-disclosure agreement binding them to use the code exclusively for non-commercial academic research purposes.

Received: 14 March 2025; Accepted: 13 June 2025
Published online: 22 September 2025

References

1. Ezzeddine, Y., Bayerl, P. S. & Gibson, H. Safety, privacy, or both: Evaluating citizens' perspectives around artificial intelligence use by Police forces. *Polic. Soc.* **33**, 861–876. https://doi.org/10.1080/10439463.2023.2211813 (2023).
2. Sandhu, A. & Fussey, P. Surveillance arbitration in the era of digital policing. *Theoretical Criminol.* https://doi.org/10.1177/1362480620967020 (2020).
3. Vinuesa, R. et al. The role of artificial intelligence in achieving the sustainable development goals. *Nat. Commun.* **11**, 233. https://doi.org/10.1038/s41467-019-14108-y (2020).
4. Mandalapu, V., Elluri, L., Vyas, P. & Roy, N. Crime prediction using machine learning and deep learning: A systematic review and future directions. *IEEE Access* **11**, 60153–60170. https://doi.org/10.1109/ACCESS.2023.3286344 (2023).
5. Safat, W., Asghar, S. & Gillani, S. A. Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques. *IEEE Access* **9**, 70080–70094. https://doi.org/10.1109/ACCESS.2021.3078117 (2021).
6. Du, Y. & Ding, N. A. Systematic review of multi-scale spatio-temporal crime prediction methods. *ISPRS Int. J. Geo-Inf.* **12**, 209. https://doi.org/10.3390/ijgi12060209 (2023).
7. Bini, S. A. Artificial intelligence, machine learning, deep learning, and cognitive computing: What do these terms mean and how will they impact health care? *J. Arthroplasty*. **33**, 2358–2361. https://doi.org/10.1016/j.arth.2018.02.067 (2018).
8. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444. https://doi.org/10.1038/nature14539 (2015).
9. Qin, Z., Wei, B. & Gao, C. Innovative LSGTime Model for Crime Spatiotemporal Prediction Baseck (No. arxiv:2503.20136). arXiv. (2025). https://doi.org/10.48550/arXiv.2503.20136

10. Lv, X., Jing, C., Wang, Y. & Jin, S. A deep neural network for spatiotemporal prediction of theft crimes. *Int. Arch. Photogram. Remote Sens. Spat. Inform. Sci.* **XLVIII-3/W2-2022**, 35–41. https://doi.org/10.5194/isprs-archives-XLVIII-3-W2-2022-35-2022 (2022).
11. Xu, G. et al. Lightweight real-time semantic segmentation network with efficient transformer and CNN. *IEEE Trans. Intell. Transp. Syst.* **24**, 15897–15906. https://doi.org/10.1109/TITS.2023.3248089 (2023).
12. Huang, B. Predicting Future Incidences of Crime Based on the CNN-Transformer Model. In *5th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*. 766–769 (2024).
13. Zheng, W., Lu, S., Yang, Y., Yin, Z. & Yin, L. Lightweight transformer image feature extraction network. *PeerJ Comput. Sci.* **10**, e1755. https://doi.org/10.7717/peerj-cs.1755 (2024).
14. Meng, L. et al. Enhancing dynamic ECG heartbeat classification with lightweight transformer model. *Artif. Intell. Med.* **124**, 102236. https://doi.org/10.1016/j.artmed.2022.102236 (2022).
15. Liu, Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 9992–10002 (2021).
16. Fang, J. et al. Msg-transformer: Exchanging local spatial information by manipulating messenger tokens. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 12053–12062 (2022).
17. Roy, A., Saffar, M., Vaswani, A. & Grangier, D. Efficient content-based sparse attention with routing transformers. Trans. Assoc. Comput. Linguist.. **9**, 53–68. https://doi.org/10.1162/tacl_a_00353 (2021).
18. Yang, L. et al. Condensenet v2: Sparse feature reactivation for deep networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 3568–3577 (2021).
19. Groff, E. R. Simulation for theory testing and experimentation: An example using routine activity theory and street robbery. *J. Quant. Criminol.* **23**, 75–103. https://doi.org/10.1007/s10940-006-9021-z (2007).
20. Qiu, D., Liu, C., Shang, Y., Zhao, Z. & Shi, J. Crime type identification using high-order deep residual network with multiple attention algorithm. *Appl. Artif. Intell.* **38**, 2428552. https://doi.org/10.1080/08839514.2024.2428552 (2024).
21. Baek, M. S., Park, W., Park, J., Jang, K. H. & Lee, Y. T. Smart policing technique with crime type and risk score prediction based on machine learning for early awareness of risk situation. *IEEE Access* **9**, 131906–131915. https://doi.org/10.1109/ACCESS.2021.3112682 (2021).
22. Karystianis, G., Cabral, R. C., Han, S. C., Poon, J. & Butler, T. Utilizing text mining, data linkage and deep learning in Police and health records to predict future offenses in family and domestic violence. *Front. Digit. Health*. **3**, 602683. https://doi.org/10.3389/fdgth.2021.602683 (2021).
23. Kang, H. W. & Kang, H. B. Prediction of crime occurrence from multi-modal data using deep learning. *PLoS ONE*. **12**, e0176244. https://doi.org/10.1371/journal.pone.0176244 (2017).
24. Berk, R. A. Artificial intelligence, Predictive policing, and risk assessment for law enforcement. *Annual Rev. Criminol.* **4**, 209–237. https://doi.org/10.1146/annurev-criminol-051520-012342 (2021).
25. Hälterlein, J. Epistemologies of predictive policing: Mathematical social science, social physics and machine learning. *Big Data Soc.* **8**, 1–13. https://doi.org/10.1177/20539517211003118 (2021).
26. Sarzaeim, P., Mahmoud, Q. H. & Azim, A. A framework for LLM-assisted smart policing system. *IEEE Access* **12**, 74915–74929. https://doi.org/10.1109/ACCESS.2024.3404862 (2024).

Acknowledgements

Not Applicable.

Author contributions

H.Y.L and C.X.C conceived and wrote the manuscript and provided the methodology. C.X.C contributed to the manuscript's design, writing, and review of certain sections. H.Y.L, Y.Q.M, and Y.M.M processed the datasets. Y.Q.M and Y.M.M supported portions of the study and performed the formal analysis. All authors reviewed the manuscript.

Funding

Not applicable.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.