International Core Journal of Engineering Volume 10 Issue 8, 2024
ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0004

Deep Learning-Based Data Completion and Prediction Model for Burglary Cases

Enqi Cao, Fanliang Bu*, and Zhuxuan Han

School of Information Technology and Cyber Security, People's Public Security University of China, Beijing, China

*Corresponding author: 2021211461@stu.ppsuc.edu.cn

Abstract

To address the problem of data completion and prediction for burglary cases, a data completion and prediction model for burglary cases is proposed. The model first constructs a Conv-BiGRU-AE data completion module to effectively fill in the missing data in burglary cases. Then, a Trans-CrimeNet crime prediction module is established to use the completed burglary data for crime prediction. Experimental results show that, under a 20% missing rate, the MAE and RMSE of burglary crime prediction by Trans-CrimeNet after completion by Conv-BiGRU-AE decreased by 17.14% and 11.68%, respectively. Under a 30% missing rate, the MAE and RMSE decreased by 13.77% and 13.17%, respectively.

Keywords

Deep Learning; Data Completion; Crime Prediction.

1. Introduction

Theft crime is a major challenge in social governance, as it involves criminal acts aimed at illegally obtaining others' property and disrupting social order. Burglary not only results in property loss but also poses potential threats to the safety of residents. Under the current concept of social governance, it is essential not only to enhance the investigation and crackdown on theft crimes but also to intensify the efforts in predicting and preventing theft crimes to minimize their occurrence.

Accurate and reliable prediction results depend on high-quality data, with missing data being a key factor affecting data quality [1]. In most data collected across various social domains, there is often some degree of missing data, which is a common and unavoidable issue [2]. Missing data directly leads to reduced data completeness and decreased correlation among different attributes within the data, thereby degrading data quality and subsequently impacting data analysis and prediction tasks [3].

Therefore, this paper constructs a model for data completion and prediction of burglary cases. Specifically, in the data completion phase, the model can use the temporal and spatial distribution characteristics of burglary cases as important labels, while focusing on capturing the sequential correlation of burglary cases over time, thus effectively completing the missing data in burglary cases. After completing the burglary case data, the model can further utilize the completed burglary case data in the crime prediction phase to extract and capture the patterns of burglary crimes, thereby achieving accurate predictions of burglary crimes.

2. Construction of a Data Completion and Prediction Model

This section proposes a data completion and prediction model for burglary cases, consisting of a Conv-BiGRU-AE data completion module and a Trans-CrimeNet crime prediction module. The

ISSN: 2414-1895
DOI: 10.6919/ICJE.202408_10(8).0004

design of this model aims to fill in the missing field values in burglary case data and to perform accurate crime prediction based on the completed data.

2.1 Conv-BiGRU-AE Data Completion Module

The Conv-BiGRU-AE consists of an encoder and a decoder, which can complete the missing data in existing burglary case data containing missing field values, resulting in complete burglary case data. Suppose the input data is X, $X = \{x_i\}_{i=1}^{N}$ where $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,M}]$ represents the i-th burglary case record in the input data X, containing M field values. N represents the number of burglary case records in X. For a certain field value $x_{i,j}$ in a burglary case record x_i, there may be missing data, where $j \in \{1,2,\dots,M\}$.

2.1.1 Encoder Part of Conv-BiGRU-AE

The encoder of Conv-BiGRU-AE consists of three layers of one-dimensional convolution (1D-CNN) [4] and four layers of bidirectional gated recurrent units (BiGRU) [5]. The three layers of 1D-CNN are responsible for extracting local features from burglary case data to capture the spatial distribution characteristics of burglary cases. Each convolutional layer is followed by BatchNorm operation and ReLU activation function to effectively extract local patterns and detailed features from the burglary case data. The specific calculation process for the l-th layer convolution and BatchNorm operation and ReLU activation is as follows:

$$H_{\text{conv}}^{l} = \text{ReLU}\left(\text{BatchNorm}\left(\text{Conv1D}^{l}(H_{\text{conv}}^{l-1})\right)\right) \tag{1}$$

Where $H_{\text{conv}}^{0} = X$, representing the input of the initial convolutional layer. H_{conv}^{l} represents the output features after the l-th layer convolution and BatchNorm operation and ReLU activation. Conv1D^{l} represents the 1D convolution operation of the l-th layer.

Subsequently, the four layers of BiGRU in the encoder process the output of the last convolutional layer to capture the bidirectional dependencies and temporal patterns in the time series of burglary case data. For the l-th layer BiGRU in the encoder, the specific calculation process is as follows:

$$H_{\text{BiGRU}}^{l}, h_t^{l} = \text{BiGRU}^{l}\left(H_{\text{BiGRU}}^{l-1}, h_{t-1}^{l}\right) \tag{2}$$

Where $H_{\text{BiGRU}}^{0} = H_{\text{conv}}^{3}$, i.e., the output of the last convolutional layer is the input of the initial BiGRU layer in the encoder. H_{BiGRU}^{l} represents the output of the l-th layer BiGRU over the entire sequence. h_t^{l} represents the hidden state at the current time step t in the l-th layer, computed based on the hidden state h_{t-1}^{l} of the previous time step and the current input feature H_{BiGRU}^{l}. BiGRU^{l} represents the BiGRU computation operation of the l-th layer.

The high-level feature representation H_{BiGRU}^{4} generated by the last layer BiGRU in the encoder is the final output of the encoder.

2.1.2 Decoder Part of Conv-BiGRU-AE

The decoder of Conv-BiGRU-AE is symmetrical to the encoder and decodes the lowdimensional representation generated by the encoder back to the original data form. The decoder first reconstructs the temporal relationships in the burglary case data using four layers of BiGRU. For the l-th layer BiGRU in the decoder, the specific calculation process is as follows:

$$H_{\text{decoded}}^{l}, h_t^{l} = \text{BiGRU}^{l}\left(H_{\text{decoded}}^{l-1}, h_{t-1}^{l}\right) \tag{3}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0004

Where $H_{\text{decoded}}^{0} = H_{\text{encoded}}$, i.e., the output of the encoder is the input of the initial BiGRU layer in the decoder. H_{decoded}^{l} represents the output of the l-th layer BiGRU over the entire sequence. h_t^l represents the hidden state at the current time step t in the l-th layer, computed based on the hidden state h_{t-1}^{l} of the previous time step and the current input feature H_{decoded}^{l-1}.

Then, the burglary case data is reconstructed using three layers of one-dimensional transposed convolution (1D-ConvTranspose) to ensure that the completed data conforms to both local spatial features and global temporal dependencies. After the first two layers of transposed convolution, BatchNorm operation and ReLU activation are further applied to restore the temporal dimension of the feature maps. The final layer of transposed convolution converts the feature maps to the size of the original input, completing the reconstruction of burglary case data. The specific calculation process for the l-th layer (the first two layers) of transposed convolution and BatchNorm operation and ReLU activation is as follows:

$$H_{\text{deconv}}^{l} = \text{ReLU}\left(\text{BatchNorm}\left(\text{ConvTranspose1D}^{l}\left(H_{\text{deconv}}^{l-1}\right)\right)\right) \tag{4}$$

Where $H_{\text{deconv}}^{0} = H_{\text{decoded}}^{4}$, i.e., the output of the last RNN layer in the decoder is the input of the initial transposed convolution layer. H_{deconv}^{l} represents the output after the l-th layer transposed convolution, BatchNorm operation, and ReLU activation. $\text{ConvTranspose1D}^{l}$ represents the 1D transposed convolution operation of the l-th layer.

In the final layer of transposed convolution, BatchNorm operation and ReLU activation are not used, directly converting the feature maps to the size of the original input to complete the reconstruction of burglary case data, achieving the completion of missing field values in burglary case data. The specific calculation process of the final layer of transposed convolution is as follows:

$$H_{\text{deconv}}^{3} = \text{ConvTranspose1D}^{3}(H_{\text{deconv}}^{2}) \tag{5}$$

Where H_{deconv}^{2} represents the output after two layers of transposed convolution, BatchNorm operation, and ReLU activation. H_{deconv}^{3} represents the output after one layer of transposed convolution, which is the reconstructed burglary case data $\tilde{X}$. $\tilde{X}$ represents the complete burglary case data without missing field values, completed by Conv-BiGRU-AE.

2.2 Trans-CrimeNet crime prediction module

Trans-CrimeNet consists of an input layer, four layers of Transformer encoders [6], and an output layer. The input layer receives burglary crime data X with a time step of T . X is defined as follows:

$$X = \{C(t)\}_{t=1}^{T} \tag{6}$$

$$C(t) = [C_1(t), C_2(t), \dots, C_N(t)]^{\top} \tag{7}$$

Where $C(t)$ represents the number of burglary cases in all areas of the city at time t. $C_i(t)$ represents the number of burglary cases in area R_i at time $t.N$ represents the number of areas in the city. T represents the number of historical data time steps, and $t \in \{1,2,\dots,T\}$ represents the current time point.

The main steps of the input layer include linear transformation and positional encoding. Specifically, it first transforms the input burglary crime data X into fixeddimension feature vectors through a linear layer (fully connected layer). The specific calculation formula for the linear transformation is as follows:

$$X_{\text{linear}} = W_{\text{linear}} X + b_{\text{linear}} \tag{8}$$

Where X_{linear} represents the output after the linear transformation. W_{linear} represents the weight matrix of the linear layer. b_{linear} represents the bias term of the linear layer.

Positional Encoding (PE) adds positional information to the input data X, enabling the module to recognize the relationships between different positions in the burglary crime time series data. The specific calculation formulas for PE are as follows:

$$PE(\text{pos}, 2\text{i}) = \sin\left(\frac{\text{pos}}{10000^{2i/d_{\text{feature}}}}\right), PE(\text{pos}, 2\text{i}+1) = \cos\left(\frac{\text{pos}}{10000^{2i/d_{\text{feature}}}}\right) \tag{9}$$

Where "pos" represents the position in the sequence. d_{feature} represents the dimension of the feature vector. i represents the dimension index in the positional encoding, used to differentiate between odd and even positions.

The final output of the input layer of Trans-CrimeNet is the feature vector with added positional encoding, namely:

$$X_{\text{input}} = X_{\text{linear}} + PE \tag{10}$$

The four layers of the Transformer encoder are used to extract the complex temporal features of burglary crimes layer by layer, focusing on the long- and short-term trends and complex temporal patterns of burglary crimes. Each layer of the Transformer encoder consists of a multi-head self-attention mechanism and a feedforward neural network. For the l-th layer Transformer encoder, the specific calculation process is as follows:

First, the output of the $(l-1)$-th layer Transformer encoder is taken as the input of the current layer, and the query matrix Q^l, key matrix K^l, and value matrix V^l are obtained through linear transformation, and the attention weights are calculated.

$$Q^l = H_{\text{trans}}^{l-1} W_Q^l,\ K^l = H_{\text{trans}}^{l-1} W_K^l,\ V^l = H_{\text{trans}}^{l-1} W_V^l \tag{11}$$

$$\text{Attention}(Q^l, K^l, V^l) = \text{softmax}\left(\frac{Q^l K^{l\top}}{\sqrt{d_k}}\right) V^l \tag{12}$$

Where $H_{\text{trans}}^0 = X_{\text{input}}$, representing the initial input as the input features with positional encoding. W_Q^l, W_K^l, and W_V^l represent the query, key, and value weight matrices of the l-th layer, respectively. d_k represents the dimension of the key vector, and $\sqrt{d_k}$ represents the scaling factor. $\frac{Q^l K^{l\top}}{\sqrt{d_k}}$ represents the dot product of the query and key divided by the square root of the key vector dimension.

Based on self-attention, to capture different features in the burglary crime time series from different perspectives and thus capture the complex temporal features of burglary crimes, a multi-head self-attention mechanism is further adopted.

$$\text{MultiHead}(Q^l, K^l, V^l) = \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{h}) W_O^l \tag{13}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0004

$$\text{head}_i = \text{Attention}\left(Q^l W_Q^{l,i}, K^l W_K^{l,i}, V^l W_V^{l,i}\right) \tag{14}$$

Where head_i represents the i-th head, with a total of h heads. $W_Q^{l,i}, W_K^{l,i}$, and $W_V^{l,i}$ represent the learnable weight matrices corresponding to each head. W_O^l represents the output linear transformation matrix.

To stabilize the training process and ensure the stability of network training and the effectiveness of gradient propagation, residual connections and layer normalization are further added after the multi-head self-attention mechanism and feed-forward neural network in each layer of the Transformer encoder.

$$H_{\text{attn}}^l = \text{LayerNorm}\left(\text{MultiHead}(Q^l, K^l, V^l) + H_{\text{trans}}^{l-1}\right) \tag{15}$$

Where H_{attn}^l represents the output of the l-th layer multi-head self-attention mechanism with residual connection and layer normalization added. H_{trans}^{l-1} represents the output of the $(l-1)$-th layer Transformer encoder. LayerNorm represents the layer normalization operation.

After adding residual connections and layer normalization to the output of the multi-head self-attention mechanism, the feed-forward neural network further processes the output of the multi-head self-attention mechanism layer by layer through two linear transformations and a nonlinear activation function to enhance its feature representation capability.

$$H_{\text{ffn}}^l = \text{LayerNorm}(\text{FFN}(H_{\text{attn}}^l) + H_{\text{attn}}^l) \tag{16}$$

Where FFN represents the feed-forward neural network operation, which consists of two fully connected layers and an activation function (such as ReLU activation function).

The output of the l-th layer Transformer encoder is the output after the multi-head self-attention mechanism and the feed-forward neural network:

$$H_{\text{trans}}^l = H_{\text{ffn}}^l \tag{17}$$

After processing by the four layers of Transformer encoders, the output of the last layer Transformer encoder is passed to the output layer of Trans-CrimeNet. The output layer uses a fully connected layer to convert the feature vector of the last time step into the final burglary crime prediction result. The specific calculation formula is as follows:

$$\hat{C}(T+\tau) = H_{\text{trans}}^4[\text{T}] W_{\text{output}} + b_{\text{output}} \tag{18}$$

Where $H_{\text{trans}}^4[\text{T}]$ represents the feature vector of the last time step output by the last layer Transformer encoder. W_{output} represents the weight matrix of the output layer, and b_{output} represents the bias term of the output layer. $\hat{C}(T+\tau)$ represents the predicted number of burglary cases in each area of the city for the future τ time steps.

3. Experimental Validation and Result Analysis

3.1 Data Preprocessing

To accurately evaluate the model, 8396 complete burglary case records without missing field values from January 1, 2018, to December 31, 2023, from a city in China, were selected as the experimental dataset after anonymization and desensitization processing. Specifically, 5433 burglary case records from January 1, 2018, to December 31, 2021, were used for the training set, and 2963 burglary case records from January 1, 2022, to December 31, 2023, were used for the test set.

In the data completion experiment, four key feature fields were selected for data completion, specifically: "Time of Crime", "Location of Crime", and "Police Substation". For the "Time of Crime" feature field, based on people's daily activities and routines, the year, month, and date were extracted from the timestamp and converted into discrete categorical features. Additionally, the 24-hour time format was divided into six time periods per day: early morning (00:00-04:00), morning (04:00-08:00), forenoon (08:00-12:00), afternoon (12:00-16:00), evening (16:00-20:00), and night (20:00-00:00). After converting the "Time of Crime" feature field, all feature fields were one-hot encoded. The number of categories and examples for each selected feature field in the burglary case data are shown in Table 1.

Table 1. Burglary Case Data Feature Fields, Number of Categories, and Examples

Field Name	Number of Categories	Specific Examples
Time of Crime	9	Year, Month, Date, Early Morning, Morning, etc.
Location of Crime	18	Detached Farmhouse, Shops and Supermarkets, etc.
Police Substation	6	Substation A, Substation B, Substation C, etc.

3.2 Experimental Design

To validate the effectiveness and superiority of the model, the Trans-CrimeNet crime prediction results before and after data completion by Conv-BiGRU-AE were compared. This was done to verify the role of Conv-BiGRU-AE in Trans-CrimeNet and to evaluate the importance of data completion for crime prediction. First, two different data missing rates (20%, 30%) were introduced to simulate the missing data situation in real-world burglary cases. For each missing rate, random missing datasets of burglary cases were generated. Then, Conv-BiGRU-AE was used to complete the data for burglary cases with the two different missing rates. After that, Trans-CrimeNet was used for crime prediction with the completed data for each missing rate.

3.3 Experimental Parameter Settings

Conv-BiGRU-AE uses the cross-entropy loss function, which consists of two parts: missing data completion loss and overall data loss. The missing data completion loss calculates the cross-entropy loss only for the missing data to be completed and assigns it a higher weight (set to 2). The overall data loss calculates the cross-entropy loss over the entire output data to evaluate the overall performance of Conv-BiGRU-AE on all data. The final loss function is the weighted sum of these two parts. Trans-CrimeNet uses the L1 loss function, also known as the Mean Absolute Error (MAE), as its main loss function for optimization.

In the data completion experiment, a continuous sequence of 32 burglary case records was taken as a sample, and data completion was performed for the missing field values in the sample records. In the crime prediction experiment, the number of burglary cases per substation per month was counted, and the number of burglary cases in the next month for each substation was predicted using the number of burglary cases in the previous six months.

3.4 Experimental Evaluation Metrics

In the burglary case data completion and prediction experiments, the specific evaluation metrics used are as follows: In the data completion experiment, accuracy, precision, recall, and F1 score were selected as evaluation metrics. The macro average method was used to calculate precision, recall, and F1 score. In the crime prediction experiment, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were selected as evaluation metrics.

3.5 Experimental Results and Analysis

Table 2 shows the experimental results of Trans-CrimeNet crime prediction before and after data completion by Conv-BiGRU-AE under different missing rates.

Table 2. Comparison of Crime Prediction Experimental Results Before and After Data Completion under Different Missing Rates

Missing Rate	Completion Status	Evaluation Metrics	
		MAE	RMSE
20%	Before Completion	1.05	1.37
	After Completion	0.87	1.21
30%	Before Completion	1.38	1.67
	After Completion	1.19	1.45

The experimental results in Table 2 indicate that data completion plays an important role in improving the accuracy of crime prediction. The completed burglary case data significantly reduced the prediction error of burglary crimes. Specifically, under a 20% data missing rate, the MAE and RMSE of the completed burglary crime prediction decreased by 17.14% and 11.68%, respectively; under a 30% missing rate, the MAE and RMSE decreased by 13.77% and 13.17%, respectively. Although the prediction performance of Trans-CrimeNet decreased under a 30% missing rate, the MAE and RMSE of crime prediction still significantly decreased after data completion by Conv-BiGRU-AE. This indicates that burglary case data completion is crucial for improving the accuracy and stability of burglary crime prediction.

4. Summary

This paper proposes a data completion and prediction model for burglary cases, consisting of a Conv-BiGRU-AE data completion module and a Trans-CrimeNet crime prediction module. Experimental results of crime prediction before and after data completion for burglary cases indicate that the accuracy of burglary crime prediction by Trans-CrimeNet significantly improves after data completion by Conv-BiGRU-AE. Data completion plays an important role in crime prediction.

References

[1] Taleb I, Serhani M A, Dssouli R. Big Data Quality: A Survey. 2018 IEEE International Congress on Big Data (BigData Congress). IEEE, 2018: 166-173.

[2] Dong Y, Peng C Y J. Principled missing data methods for researchers. SpringerPlus, 2013, 2: 222.

[3] Laranjeiro N, Soydemir S N, Bernardino J. A Survey on Data Quality: Classifying Poor Data. 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC). IEEE, 2015: 179-188.

[4] Alzubaidi L, Zhang J, Humaidi A J, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 2021, 8: 53.

[5] Wang S, Shao C, Zhang J, et al. Traffic flow prediction using bi-directional gated recurrent unit method. Urban informatics, 2022, 1(1): 16.

[6] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need. Advances in Neural Information Processing Systems 30 (NIPS 2017). Curran Associates, 2017: 5998-6008.