UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Crime Forecasting using Neural Networks and Transfer Learning

Diploma Thesis

Nikolaos Nikolaidis

Supervisor: Rafailidis Dimitrios

June 2023

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Crime Forecasting using Neural Networks and Transfer Learning

Diploma Thesis

Nikolaos Nikolaidis

Supervisor: Rafailidis Dimitrios

June 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Πρόβλεψη Εγκλημάτων με την Χρήση Νευρωνικών Δικτύων και Μεταφορά Μάθησης

Διπλωματική Εργασία

Νικόλαος Νικολαΐδης

Επιβλέπων: Ραφαηλίδης Δημήτριος

Ιούνιος 2023

Approved by the Examination Committee:

Supervisor **Rafailidis Dimitrios**
Associate professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Thanos Georgios**
Laboratory Stuff, Department of Electrical and Computer Engineering, University of Thessaly

Member **Katsaros Dimitrios**
Associate professor, Department of Electrical and Computer Engineering, University of Thessaly

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Nikolaos Nikolaidis

Diploma Thesis

Crime Forecasting using Neural Networks and Transfer Learning

Nikolaos Nikolaidis

Abstract

Safety is one of the most critical foundations of modern societies. The treatment so far is limited to the afterward investigation of crimes, and the prevention solely focuses on socioeconomic factors like poverty, income inequality, and material or mental conditions. In data science, we want to reinforce the part that focuses on preventing crimes by developing a tool that will effectively predict the approximate location, timeframe, and type of crime that will occur. Extensive research papers have examined the issue of Crime Forecasting, providing each time with state-of-the-art methods. In order to enhance our current work, we begin by experimenting with fundamental time series forecasting techniques that are conventionally employed to analyze data with temporal relationships. For the proposed model, we train a HAGEN model for each category, which is currently the most advanced method available. Then, we use our crime similarity algorithm to analyze the distribution of crimes and identify any similarities between them. In order to find the best model, we must do hyper-parameter tuning and assess the performance of two major cities for a period of five months across all categories.

Keywords:

Crime Forecasting, Transfer Learning, Graph Neural Networks

Διπλωματική Εργασία

Πρόβλεψη Εγκλημάτων με την Χρήση Νευρωνικών Δικτύων και Μεταφορά Μάθησης

Νικόλαος Νικολαΐδης

Περίληψη

Η ασφάλεια είναι ένα από τα πιο κρίσιμα θεμέλια των σύγχρονων κοινωνιών. Η αντιμετώπιση μέχρι στιγμής περιορίζεται στη μετέπειτα διερεύνηση εγκλημάτων και η πρόληψη επικεντρώνεται αποκλειστικά σε κοινωνικοοικονομικούς παράγοντες όπως η οικονιμική κατάσταση και η κοινωνικές ανισότητες. Στην επιστήμη δεδομένων επιδιώκουμε την ανάπτυξη ενός εργαλείου που θα προβλέπει αποτελεσματικά την κατά προσέγγιση της τοποθεσία, την χρονική στιγμή και τον τύπο του εγκλήματος. Εκτενείς ερευνητικές εργασίες έχουν εξετάσει το θέμα της Πρόβλεψης Εγκλήματος, παρέχοντας κάθε φορά σύγχρονες μεθόδους. Προκειμένου να βελτιώσουμε την τρέχουσα δουλειά πάνω στην πρόβλεψη εγκλημάτων, ξεκινάμε με τη μελέτη θεμελιωδών τεχνικών πρόβλεψης χρονοσειρών που χρησιμοποιούνται συμβατικά για την ανάλυση δεδομένων με χρονικές σχέσεις. Στην συνέχεια για την δική μας μέθοδο, εκπαιδεύουμε ένα μοντέλο HAGEN για κάθε κατηγορία, το οποίο είναι αυτή τη στιγμή το πιο προηγμένο μοντέλο. Στη συνέχεια, χρησιμοποιούμε έναν αλγόριθμο ομοιότητας εγκλημάτων για να αναλύσουμε την κατανομή τους και να εντοπίσουμε τυχόν ομοιότητες μεταξύ τους. Για να βρούμε το καλύτερο μοντέλο, πρέπει να κάνουμε συντονισμό υπερ-παραμέτρων και για την αξιολόγηση της απόδοσης χρησιμοποιούμε τα δεδομένα δύο μεγάλων πόλεων, στα οποία ελέγχουμε τους πέντε τελευταίους μήνες και την κάθε κατηγορία ξεχωριστά.

Λέξεις-κλειδιά:

Πρόβλεψη Εγκλημάτων, Μεταφορά Μάθησης, Νευρωνικά Δίκτυα Γράφων

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Modern urban communities are currently suffering from waves of criminal behaviors. On the one hand, understanding crime trends can assist individuals in making informed decisions on safeguarding themselves and their possessions [4, 5]. On the other hand, it can equip law enforcement with the necessary tools to address these issues effectively and build a crime prevention program since its approach focuses on dealing with the socioeconomic factors that create these anti-social behaviors [4, 5].

According to statistics from the United Nations, urbanization is happening rapidly. By 2050, 64 % of the developing world and 86% of the developed world will be urbanized [6, 7]. As urbanization continues, social inequality will also increase. Analysts currently examine demographic data to address this issue, including wealth disparity [8, 9], educational attainment, and ethnic and religious factors [10, 11, 12]. Although, collecting demographic data can be difficult because it only sometimes accurately represents the geographical dynamics of communities.

All of the above is the core of specific interpretations of criminal behavior. For instance, social criminology and culture conflicts state that criminal behavior results from a clash of different socialized groups based on values. These values come from what is considered acceptable behavior and are highly correlated with religious values and economic classes [13].

On the other hand, environmental criminology, specifically rational choice theory [14] and routine activity theory [15], suggest that the distribution of crime is influenced by time and space, with human mobility being a significant factor. Recently, researchers have turned their attention to environmental criminology, a facet of criminology, as data mining and machine learning continue to advance. This thesis aims to contribute to the development of

environmental criminology by employing machine learning to forecast crime, including its location, time, and type. This approach, Crime Forecasting, utilizes past criminal records to anticipate potential occurrences across all categories and regions. For this thesis, it is essential to define criminal records and Crime Forecasting as follows:

Definition 1 ***Criminal Record.*** Given a region r_i, the criminal records of the region for all timeslots $Y_i = (y_{i,1}^1, \ldots, y_{i,C}^K) \in \mathbb{R}^{C \times K}$. C denotes the number of categories, and K the sequence length for the timeslots $T = (t_1, \ldots, t_k)$ [2].

Definition 2 ***Crime Forecasting.*** For the criminal records $\{Y^1, \ldots, Y^K\}$, the goal is to find a function $p(.)$ that predict the criminal record Y^{K+1}. For the models that learns the adaptive graph the problem is modified to $\{Y^1, \ldots, Y^K, G\} \xrightarrow{p(.)} \{Y^{K+1}, G'\}$ [2].

To approach the problem, we employ several preliminary techniques such as Multilayer-Perceptron (MLP), Long-Short-Term-Memory (LSTM), and Gated-Recurrent-Unit (GRU). After reviewing past work, we consider utilizing a region graph to enhance our understanding of the areas where criminal activities occur.

Definition 3 ***Geographical Region.*** The geographical regions are the result of the partitioning of a city in a grid-based map segmentation, with I rows and J columns [1].

Definition 4 ***Region Graph.*** A region can be defined as the set $G = (V, \in, A_r)$ where V is the region nodes, ϵ is the directional edges between the nodes , A_r is the weight matrix that represents the weights between the nodes [2].

Definition 5 ***Weight Matrix.*** The weight matrix $A_r \in \mathbb{R}^{N \times N}$ is a representation of weighted graph G with $A_r(i,j) = w > 0$ if $(u_i, u_j) \in \mathcal{E}$, else $A_r(i,j) = 0$. In this implementation the matrix is uni-directional, meaning that if $A_r(i,j) > 0$ then $A_r(j,i) = 0$ to increase performance [2].

The first approach by Huang et al. [1] introduces the MiST framework, which uses a Recurrent-Neural-Network (RNN) based model that captures regional dependencies using an attention mechanism. However, the nature of the problem (see Definition 2), with the addition of the region graph, compels researchers to introduce Graph Neural Networks to the

problem. Using DCRNN [16] is the most optimal approach as it expertly combines the diffusion convolution function for spatial properties with the RNN for temporal properties. Sun et al. [3] introduce the above method with the CrimeForecaster framework using the DCGRU method, which uses GRU for the temporal properties. Then Wang et al. [2] introduce the HAGEN model that expand the previous implementation by adding several features to the encoder and using the homophily ratio as a constraint to regularize the optimization, using the hypothesis that regions with similar socioeconomic situation have similar criminal behavior [17, 18].

Definition 6 ***Homophily Ratio.*** Given the graph $G = (V, \in, A_r)$ and node label Y the homophily ratio is defined as: $H(G, V) = \frac{1}{|V|} \sum_{u \in V} \frac{|u:u \in N_v \wedge y_u = y_v|}{|N_v|}$, which denotes the probability that neighboring nodes (N_v) share the same label (y_u) [19].

To expand the previous work, we want to explore how the distribution of different categories has similarities to the use of transfer learning, reinforcing the model's performance. Using the HAGEN [2] framework, we process each category separately and apply four different transfer learning techniques.

The contribution of the thesis diploma is concluded as follows:

1. Analyse traditional time series forecasting models and spatiotemporal models for crime forecasting.
2. Train a separate HAGEN model for each category.
3. Examine the correlation between the categories distribution.
4. Create different variations to weight the similarity of distributions.
5. Examine different hyper-parameters.
6. Evaluate the models with the established evaluation protocol.

The thesis has three different chapters. Chapter 1 covers related work on crime forecasting, dividing the discussion into traditional models for time series forecasting and spatiotemporal graph networks. Chapter 2 analyzes the proposed transfer learning model and

its variations. Lastly, Chapter 3 focuses on the dataset used, the evaluation protocol, and the results of the experiments.

Chapter 2

Related Work

In order to address Crime forecasting, which is defined as predicting future criminal activity, we must review past approaches and consider other frameworks used for similar issues, such as traffic forecasting. The early approaches only rely on temporal dependencies to forecast sequential data. As a result, the models used were standard frameworks designed to address the problem. Long-Short-Term-Memory [20, 21] and Gated-Recurrent-Unit [22] are recurrent models commonly used for time series forecasting. Additionally, Multilayer Perceptron [23] can approximate data, making it a valuable tool for solving forecasting problems.

Then, the following two approaches from Huang et al. [1, 24] use the spatial dependencies for the first time to improve the performance. There are two implementations: DeepCrime [24], which uses a CNN-based model, and MiST, which uses an RNN-based [1] model. Neither implementation utilizes GNNs to handle regional dependencies, instead relying on regional embeddings.

Sun et al. [3] introduce the CrimeForecaster, which is the first approach to use GNN. They use DCRNN [16], specifically DCGRU, to capture spatial and temporal dependencies. This implementation uses DCGRU for data encoding and a Multilayer Perceptron for result decoding. Then, the state-of-the-art HAGEN [2] model, similar to the CrimeForecaster, follows the same Encoder-Decoder [25] architecture and still uses the DCGRU method. In addition, they use the homophily ratio to measure the similarity between regions based on crime rates. They also include Point-Of-Interests using embeddings and compressed criminal records to create crime embeddings. The compression method is Principal-Component-Analysis, defined as an orthogonal linear transformation that transforms the data to a new coordinate system such

that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on [26].

In the field of traffic forecasting, the most advanced models that utilize spatial-temporal methods are STGCN [27], Graph-WaveNet [28], and GMAN [29]. AGCRN [30], and MTGNN [31] also employ adaptive graph learning to address traffic forecasting. The above methods have fundamental problems of GNNs, such as node similarity, sparsity, and symmetry. STGCN stands for Spatio-Temporal Graph Convolution Network. It comprises two gated sequential convolution layers and a spatial convolution layer. Graph-WaveNet is a model that combines spatial and temporal aspects. It uses a node embeddings matrix to capture hidden spatial dependencies and a 1D convolution component capable of handling long sequences. GMAN follows an Encoder-Decoder architecture that consists of multiple spatiotemporal attention blocks. Between Encoder and Decoder, an attention transform layer transforms the data as sequence representations for the Decoder to predict the output sequence.

The term AGCRN refers to an Adaptive Graph Convolutional Recurrent Network. This network includes the NAPL-GCN, which replaces MLP layers in the GRU to handle node-specific patterns, as well as the DAGG and GRU units. The MTGNN (Multivariate Graph Neural Networks) is a cutting-edge model that identifies the one-way connections between nodes and trains end-to-end graph learning models using graph convolution for spatial data and temporal convolution for convolution data. To summarize, traffic prediction and crime forecasting problems share a similar format. Therefore, the methods above can also apply to crime forecasting. However, these methods may need to perform better in crime forecasting due to their inability to handle sparsity and node similarity.

Chapter 3

Examined Models

3.1 Input/Output

Given the public records, we need to convert them to mathematical matrices that the model can process. The problem is that each crime is happening at a specific address, at a particular timestamp, and with an accurate crime description, creating many different possibilities. We aim to narrow these possibilities by dividing the city into R regions, daytime into K timeslots, and grouping various crimes into C big categories. Since we don't care about the number of crimes happening in a region at a specific timeslot, just the existence of crimes, in the end, the result of the matrix is a binary one, with "0" indicating that no crime occurred and "1" indicating that a crime did occur. This matrix can be descried mathematical as: $X \in \mathbb{R}^{N \times K \times R \times C}$,where the variable N represents the total number of days available for analysis. From the previous definition, the output matrix can be defined as: $y \in \mathbb{R}^{N \times 1 \times R \times C}$.

3.2 Preliminary Models

As preliminary methods we describe the simplest methods which are used for time series forecasting. Traditionally RNNs are suitable due to their ability to store information from previous inputs. Another alternative is the Multi-layer Perceptron with the ability to approximate data.

3.2.1 Multi-layer Perceptron

The mathematical form for the Multi-layer Perceptron goes as follows:

$\psi_1 = ReLU(W_1 * x_1 + b_1)$

$\dots$

$\psi_n = ReLU(W_n * \psi_{n-1} + b_{n-1})$

$y = \sigma(W'\psi_n + b')$

Note: The first layer uses 1024 hidden units and 512 for the second. The goal is to minimize the loss function: $L_{crime} = -\sum_{i\in\{1,\dots,R\},l\in\{1,\dots C\}}(y_{i,l}\log y'_{i,l} + (1-y_{i,l})\log(1-y'_{i,l}))$

3.2.2 Recurrent Neural Networks

For RNN, we present LSTM and GRU, both fundamental and similar models. Exactly like the Multilayer Perceptron the objective function is Binary Cross Entropy.

3.2.3 Gated Recurrent Unit

The equations for the Gated Recurrent Unit(GRU) are:

$z_t = \sigma(W_z * [h_{t-1}; x_t])$

$r_t = \sigma(W_r * [h_{t-1}; x = t])$

$h_t^{\sim} = tanh(W * [r_t * h_{t-1}; x_t])$

$h_t = (1 - z_t) * h_{t-1} + z_t * h_t^{\sim}$

3.2.4 Long Short-Term Memory

$f_t = \sigma(W_f * x_t + U_f * h_{t-1} + b_f)$

$i_t = \sigma(W_i * x_t + U_i * h_{t-1} + b_i)$

$o_t = \sigma(W_o * x_t + U - o * h_{t-1} + b_o)$

$c_t^{\sim} = \sigma(W_c * x_t + U_c * x_{t-1} + b_c)$

$c_t = f_t \odot c_{t-1} + i_t \odot c_t^{\sim}$

$h_t = o_t \odot \sigma(c_t)$

3.3 MiST

The MiST model uses an LSTM encoder and a Multilayer Perceptron as the decoder. The MiST framework also adds attention mechanism defined by Huang et al. [1]:

$\eta^k_{i,j,l} = tanh(W^k(h^k_{i,j,l}, e_{r_{i,j}, e_{c_j}}) + b_k)$

$\alpha^k_{i,j,l} = \frac{exp(\eta^k_{i,j,l})}{\sum_{i,j \in G} \sum_{l=1}^{L} exp(\eta^k_{i,j,l})}$

Then the actual output is calculated:

$q^k = \sum_{i,j \in G} \sum_{l=1}^{L} \alpha^k_{i,j,l} \eta^k_{i,j,l}$

The input h to the attention layer is the output of the LSTM. Also the e_r and e_c are region and crime embeddings respectively. Lastly W and b are the weights and biases.

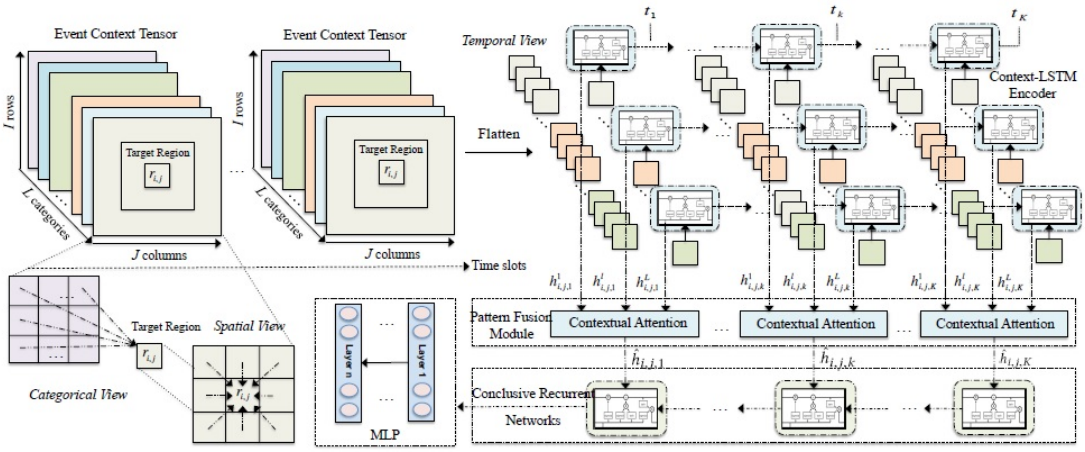


Figure 3.1: MiST Framework [1]

3.4 HAGEN model

Generally HAGEN follows the Encoder-Decoder proposed by Cho et al. [25] The following implementation is an extension of that model.

3.4.1 Region Graph

The processed graph is an adaptive weight matrix uni-directional graph (i.e if $A_r(i,j) > 0$ and $A_r(j,i) = 0$):

$Z_s = tanh(\alpha E_s \Theta_1)$

$Z_t = tanh(\alpha E_t \Theta_2)$

$A_r = ReLU(tanh(\alpha(Z_s Z_t^T - Z_t Z_s^T)))$

3.4.2 Encoding input

The encoding of data to use as input uses the criminal records y^k and the geographical dependencies with the form of embedding matrices. The first embedding matrix is, $E_c \in \mathbb{R}^{D\times C}$, which is a result of the Principal Component Analysis(PCA) [26] of the criminal records used for training. The geographical dependencies are calculated by the Node2Vec [32] based on the distances and Point-Of-Interest creating and embedding matrix $E_{pre} \in \mathbb{R}^{R\times R}$. To encode the input we first calculate the element-wise product of both embeddings creating the matrix $W_{inter} \in \mathbb{R}^{R\times C}$. The final input is calculated as $X = W_{inter} \odot y^k \in \mathbb{R}^{R\times C}$.

3.4.3 Temporal Module

The definition of DCGRU is:

$r^t = \sigma(f_*([x^t, h^{t-1}]; G, \Theta_r, D_w) + b_r)$

$u^t = \sigma(f_*([x^t, h^{t-1}]; G, \Theta_u, D_w) + b_u)$

$c^t = tanh(f_*([x^t, r^t \odot h^{t-1}]; G, \Theta_c, D_w) + b_w)$

$h^t = u^t \odot h^t + (1 - u^t) \odot c^t$

The explanations of the symbols are:

- $\sigma(.)$ is the sigmoid function calculated as $\sigma(z) = \frac{1}{1+e^{-z}}$.
- $tanh(.)$ is the hyperbolic tangent function calculated as $tanh(z) = \frac{e^z-e^{-z}}{e^z+e^{-z}}$.
- $f_*(.)$ is the diffusion convolution function where:

  $f_*(X; G, \Theta, D_w) = \sum_{m=0}^{M}(S_m^O X\Theta_{:,:,m,1} + \cdots + D_w S_m^l X\Theta_{:,:,m,2})$

  $S_m^O = (D_O^{-1}A_r)^m$, $S_m^I = (D_I^{-1}A_r)^m$.
- x^t, h^t are input and output respectively. r^t, u^t, c^t are reset gates. b_r, b_u, b_w are biases. $\Theta_r, \Theta_u, \Theta_c$ are filter parameters.

The goal of the HAGEN model is to minimize the following loss function using Adam optimization [33]:

$L_{HAGEN} = L_{crime} + \lambda L_{homo}$ where:

$L_{homo} = \sum_{k=1}^{K}\sum_{l=1}^{C}[H(A_r, y_l^k, l) - 1]^2$, the $H(.)$ function denotes the extension of the basic homophily-ratio. The extension by Zhu et al. [19] is calculated as:

$H(A_r, y_l^k, l) = \frac{1}{|V|}\sum_{v\in V}\frac{\sum_{u\in N(u), y_{u,l}^k = y_{v,l}^k} A_r(u,v)}{\sum_{u\in N(u)} A_r(u,v}$

$L_{crime} = -\sum_{i\in\{1,\ldots,R\}, l\in\{1,\ldots C\}}(y_{i,l}\log y'_{i,l} + (1 - y_{i,l})\log(1 - y'_{i,l}))$

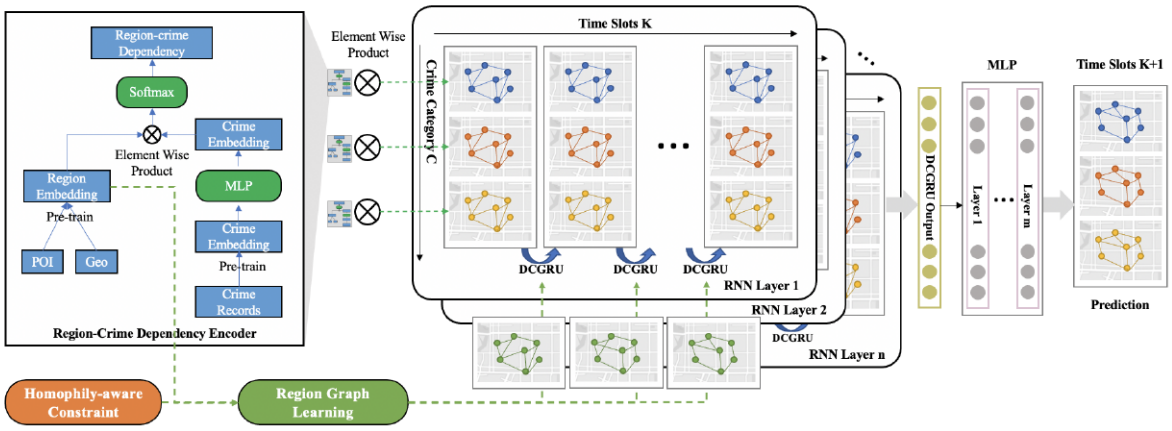


Figure 3.2: HAGEN Framework [2]

3.5 Transfer Learning Model

Given the previous definition of the HAGEN, our modification is to change the number of crime categories processed to one. Hence creating eight new models that are pre-trained, and their output will use for transferring the learning.

3.5.1 Category Similarities

The following algorithm 1 shows how to extract the similarities of crime distributions. The algorithm's input is the criminal records, presented with the variable Y, the number of regions is the variable R, and the number of categories is the variable C. Then it proceeds to initialize the final matrix "TotalSimilarities" to a $C \times C$ matrix with zeros. For the calculation, we create three nested for-loops, with the first showing the iteration of all regions and the other two iterations leading through all pairs of categories. During each iteration, we determine the frequency of criminal activity during a specific timeslot and then calculate the proportion by dividing it by the highest number of criminal incidents for each category. To obtain the mean for each similarity, we must divide each by the number of regions.

The following Figure 3.3, captures the resulted matrix for the previous algorithm, for the month of December in Los Angeles. The categories have similarities ranging from 0.12 to 0.41 and the average being to 0.3, indicating significant resemblances.

Algorithm 1 Categories Similarities

procedure Crime-Similarity(Y, R, C)
    TotalSimilarities ← Zeros(C,C)
    for $r \leftarrow 0$ to R **do**
        for Category1 $\leftarrow 0$ to C **do**
            for Category2 $\leftarrow 0$ to C **do**
                NonZeros1 = Length($Y^r_{Category1} \neq 0$)
                NonZeros2 = Length($Y^r_{Category2} \neq 0$)
                MaxNonZeros = max(NonZeros1,NonZeros2)
                C = Length($Y^r_{Category1} \cap Y^r_{Category2}$ and $Y^r_{Category1} = Y^r_{Category2} = 1$)
                TotalSimilarities(Category1,Categpry2) += $\frac{C}{MaxNonZeros}$
            end for
        end for
    end for
    TotalSimilarities ← $\frac{TotalSimilarities}{R}$
    return TotalSimilarities
end procedure

Figure 3.3: Example of the output for categories similarities algorithm.

3.5.2 Final Output

The output for the i^{th} model is symbolized as H_i^t. Combining the previous models we can calculate the final output of the i^{th} model as $H_i^t \leftarrow H_i^t + \lambda \sum w_{i,j} H_j^t$, where $w_{i,j}$ is the similarity of the crime categories i and j.

The first variation of the previous model is by using the output defined prior and we use it as an input to a attention layer. The attention layer used is the Additive Attention Layer or Bahdanau Attention, calculated as: $f_{att}([h_i; s_j]) = u_a^T tanh(W_a[h_i; s_j])$ [34, 35], where u_a, W_a are attention parameter and H, s are both use the output of HAGEN models. For the training of this variation is happening by freezing the HAGEN layers, and only optimizing the attention layer.

The other variations discard the attention layer as it shows no improvement to the model. So the next step is to use just the information from the pre-trained model. The next variation sets λ value to zero, to examine the performance without models transferring learning between them. Then we examine just the outputs of the pre-trained models and transfer learning between them. Finally the last model is identical to the previous but uses the threshold of the original pre-trained model.

Chapter 4

Experiments

4.1 Dataset

In the experiments we use datasets from two major cities, Chicago [36, 37] and Los Angeles [38, 39]. To create the data-sets from the original crime records, it is important to group the categories of the crimes to smaller subset. The chosen categorization by Sun et al. [3] is shown at the Table 4.1, also the counts of each category are shown at Figures 4.1 and 4.2:

Table 4.1: Categories for both cities

Los Angeles	**Chicago**
Theft	Theft
Vehicle Theft	Battery
Burglary	Burglary
Fraud	Deceptive Practices
Assault	Assault
Vandalism	Criminal Damage
Robbery	Robbery
Sexual Offenses	Narcotics

Category	Theft	Vehicle theft	Burglary	Fraud	Assault	Sexual offenses	Robbery	Vandalism
Counts	66,697	17,123	14,517	15,578	32,372	6,161	8,864	17,123

Figure 4.1: Total instances in Los Angeles Dataset [3]

Category	Theft	Criminal damage	Narcotics	Robbery	Assault	Deceptive practices	Burglary	Battery
Counts	56,695	28,589	21,607	9,632	16,692	14,085	13,103	48,824

Figure 4.2: Total instances in Chicago Dataset [3]

4.2 Evaluation Protocol

To properly present the evaluation metrics, we must first explain the process of converting output matrices from continuous values to binary values. The first step is to calculate the threshold from training and validation datasets. The threshold is calculated as: $thr = 1 - \overline{A}$,where $A = [y_{tr}; y_{val}]$. Secondly, the output of the model is scaled as: $h_{scaled} = \frac{h - h_{min}}{h_{max} - h_{min}}$. Lastly at each batch it is calculated the threshold by calculating the quantile of the first threshold to the scaled output. Then for the values greater than threshold we assign 1 and for less 0.

4.2.1 Evaluation Metrics

Crime forecasting can be described as multi-class classification problem. The metrics used to evaluate the model are used to multi-class classification problems.

- **Macro-F1** : Macro-F1 = $\frac{1}{C} \sum_{l=1}^{C} \frac{2TP_l}{2TP_l + FN_l + TP_l}$ [40]
- **Micro-F1** : $\frac{2 \sum_{l=1}^{C} TP_l}{2 \sum_{l=1}^{C} TP_l + \sum_{l=1}^{C} FN_l + \sum_{l=1}^{C} FP_l}$ [40]
- **AUC** : Sensitivity = Recall = $\frac{TP}{TP+FN}$ Specificity = $\frac{TN}{FP+TN}$
  The value is calculated from $Sensitivity$(TPR)-$(1 - Specificity)$(FPR) [41]

Since we have 8 different processes we need to calculate a summarized form of the individual tasks. Since that the dataset is unbalanced,we need to calculate the contribution of each category to the complete dataset. The contribution is described in the following vector:

$$c = \begin{bmatrix} \frac{N_1}{S} \\ \frac{N_2}{S} \\ \vdots \\ \frac{N_C}{S} \end{bmatrix}$$

where, S is the total number of crime instances in the labels of dataset. N_i is the total number crime instances for category i.

Then by calculating the above metrics for each task we create the following vectors:

$$MicroF1s = \begin{bmatrix} MicroF1^1 \\ MicroF1^2 \\ \vdots \\ MicroF1^C \end{bmatrix} \; MacroF1s = \begin{bmatrix} MacroF1^1 \\ MacroF1^2 \\ \vdots \\ MacroF1^C \end{bmatrix} \; AUCs = \begin{bmatrix} AUC^1 \\ AUC^2 \\ \vdots \\ AUC^C \end{bmatrix} \tag{4.1}$$

The final metric for the whole dataset is calculated as the dot product of the metrics vectors described above and the contributions:

$MicroF1 = c \cdot MicroF1s$

$MacroF1 = c \cdot MacroF1s$

$AUC = c \cdot AUCs$

4.2.2 Train-Test-Validation Split

The train-test-validation follows the model by Sun et al. [3]. Test utilizes the last month, validation utilize the 0.5 month before that and the rest as the train (i.e 6.5 months for August etc.)

4.3 Models

- **MLP**: Multi-layer Perceptron, where records of a timeslot is processed.
- **LSTM**: Long-Short-term Memory, a supervised neural network model that examines the records of a timeslot.
- **GRU**: Gated Recurrent Unit, a recurrent supervised neural network that learns from the records of a timeslot.

- **MIST**: To encode and decode information, we use LSTM and MLP, along with region embeddings. [1]

- **HAGEN**: A Graph Neural Network is used to analyze geographical dependencies. The encoder and decoder utilize DCGRU cells, while an MLP is used for the decoding process. [2]

- **Attn** : Transfer learning between the HAGEN models using category similarities and an Additive Attention Layer.

- $\boldsymbol{\lambda = 0}$: Single-task with non transferable learning.

- **NoAttn**: Transfer learning between the HAGEN models without attention layer.

- **NewThr**: Similar to NoAttn but changes the evaluation threshold by using single task HAGEN's old threshold.

4.4 Performance Evaluation

Table 4.2: Los Angeles Micro-F1 Score per Category

Category/Model	MLP	LSTM	GRU	MIST	HAGEN	Attn	$\lambda = 0$	NoAttn	NewThr
Theft	0.398	0.701	**0.747**	0.653	0.743	0.728	0.730	0.727	0.731
Vehicle theft	0.472	0.602	0.601	0.420	0.727	0.730	**0.731**	0.730	0.730
Burglary	0.513	0.445	0.499	0.341	0.701	**0.713**	0.702	0.708	0.704
Fraud	0.427	0.501	0.466	0.370	**0.734**	0.726	0.726	0.723	0.727
Assault	0.547	0.568	0.632	0.479	**0.736**	0.727	0.727	0.724	0.724
Vandalism	0.506	0.533	0.634	0.420	0.716	0.716	0.715	0.716	**0.718**
Robbery	0.520	0.465	0.446	0.575	**0.832**	0.809	0.811	0.808	0.809
Sexual offenses	0.513	0.519	0.439	0.598	**0.865**	0.831	0.830	0.832	0.830

Table 4.3: Chicago Micro-F1 Score per Category

Category/Model	MLP	LSTM	GRU	MIST	**HAGEN**	Attn	$\lambda = 0$	NoAttn	NewThr
Robbery	0.558	0.424	0.532	0.551	**0.757**	0.741	0.728	0.7467	0.741
Battery	0.490	0.506	0.503	0.509	**0.703**	0.675	0.676	0.677	0.664
Deceptive	0.560	0.514	0.493	0.613	**0.725**	0.698	0.690	0.705	0.695
Burglary	0.492	0.555	0.478	0.514	**0.716**	0.693	0.684	0.695	0.690
Assault	0.504	0.673	0.596	0.582	**0.775**	0.750	0.747	0.746	0.733
Theft	0.527	0.524	0.565	0.588	**0.700**	0.678	0.670	0.681	0.679
Damage	0.442	0.516	0.398	0.588	**0.782**	0.757	0.744	0.744	0.741
Narcotics	0.521	0.518	0.462	0.706	**0.752**	0.735	0.739	0.736	0.739

Firstly, we must evaluate the performance, from Figures 4.2 and 4.3, of each category since our model is designed to split the categories and has a separate model for each. The top-performing model for each class in the Los Angeles dataset tends to vary. In the initial models, MLP has the poorest performance compared to LSTM and GRU, which perform slightly better or, in some cases, worse than the spatiotemporal model, MiST. However, the HAGEN model and the four proposed models outperform the rest and alternate the best performance, with the differences varying from 0.01 to 0.001. In contrast, the Chicago dataset consistently produces more steady results, with the HAGEN model performing better than the others by at least 0.2 to 0.3.

Table 4.4: Los Angeles AUC Score per Category

Category/Model	MLP	LSTM	GRU	MIST	HAGEN	Attn	$\lambda = 0$	NoAttn	NewThr
Theft	0.49	0.508	0.504	0.519	**0.723**	0.707	0.709	0.705	0.709
Vehicle Theft	0.460	0.564	0.522	0.581	0.674	0.674	**0.675**	0.674	0.673
Burglary	0.490	0.407	0.451	0.538	**0.635**	0.633	0.620	0.628	0.622
Fraud	0.483	0.492	0.425	0.564	**0.656**	0.648	0.648	0.641	0.649
Assault	0.516	0.537	0.480	0.557	**0.727**	0.7202	0.719	0.717	0.716
Vandalism	0.527	0.477	0.477	0.558	0.683	0.6846	0.684	0.684	**0.686**
Robbery	0.501	0.444	0.472	0.666	**0.685**	0.6608	0.664	0.658	0.661
Sexual Offenses	0.528	0.533	0.503	0.639	**0.656**	0.614	0.613	0.609	0.612

Table 4.5: Chicago AUC Score per Category

Category/Model	MLP	LSTM	GRU	MIST	**HAGEN**	Attn	$\lambda = 0$	NoAttn	NewThr
Robbery	0.533	0.489	0.519	0.636	**0.737**	0.715	0.701	0.720	0.716
Battery	0.503	0.495	0.585	0.596	**0.657**	0.638	0.639	0.641	0.623
Deceptive	0.573	0.509	0.499	0.632	**0.699**	0.654	0.646	0.663	0.652
Burglary	0.519	0.515	0.473	0.585	**0.700**	0.679	0.669	0.681	0.675
Assault	0.508	0.491	0.484	0.660	**0.735**	0.705	0.703	0.701	0.701
Theft	0.558	0.561	0.505	0.540	**0.697**	0.676	0.669	0.680	0.678
Damage	0.442	0.516	0.356	0.662	**0.753**	0.738	0.723	0.723	0.720
Narcotics	0.498	0.526	0.528	0.673	**0.692**	0.648	0.653	0.6498	0.652

To further examine the performance of individual categories, we also look at Tables 4.4 and 4.5, which show the performance of tasks for the AUC metric. Generally, the AUC table shows the same ambiguous results as the AUC. But it is worth noticing that the models outperform at specific categories at the Micro-F1 tables and didn't outperform the same at the AUC table. The only steady performances from the four proposed models are the single-task model for "Vehicle Theft" and the transfer learning with the new threshold for "Vandalism." The differences between the models increase, with the maximum being 0.2, and the preliminary models are significantly worse than the rest. On the other hand, results from the Chicago dataset follow the same distribution as Table 4.3.

Table 4.6: Los Angeles per Month

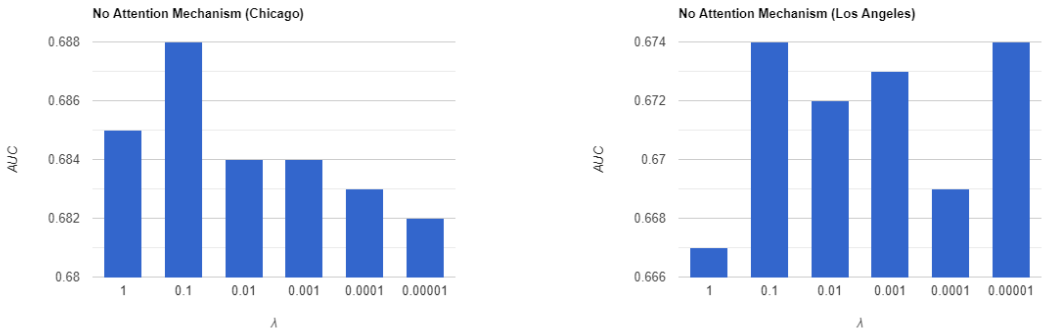
Month/Model	Metric	MLP	LSTM	GRU	MIST	HAGEN	Attn	$\lambda = 0$	NoAttn	NewThr
August	Micro-F1	0.410	0.460	0.455	0.539	0.612	0.738	**0.739**	0.731	0.735
August	Macro-F1	0.353	0.440	0.435	0.504	0.546	0.684	0.686	0.673	**0.680**
August	AUC	0.602	0.513	0.516	0.616	0.673	0.684	**0.686**	0.674	0.681
September	Micro-F1	0.415	0.457	0.439	0.540	0.619	0.738	**0.739**	0.737	0.738
September	Macro-F1	0.351	0.435	0.419	0.500	0.537	**0.683**	0.683	0.680	0.681
September	AUC	0.607	0.588	0.495	0.619	0.667	0.684	**0.685**	0.681	0.683
October	Micro-F1	0.448	0.448	0.468	0.539	0.612	0.731	0.732	0.733	**0.734**
October	Macro-F1	0.372	0.433	0.444	0.501	0.536	0.675	0.676	0.678	**0.678**
October	AUC	0.625	0.511	0.536	0.622	0.667	0.675	0.677	0.679	**0.679**
November	Micro-F1	0.436	0.460	0.451	0.541	0.611	0.738	0.737	0.738	**0.739**
November	Macro-F1	0.343	0.437	0.431	0.501	0.539	0.682	0.681	0.681	**0.683**
November	AUC	0.615	0.518	0.512	0.625	0.670	0.683	0.682	0.682	**0.684**
December	Micro-F1	0.373	0.457	0.471	0.539	0.612	**0.734**	0.729	0.731	0.730
December	Macro-F1	0.332	0.431	0.442	0.497	0.529	**0.676**	0.670	0.673	0.672
December	AUC	0.498	0.520	0.541	0.625	0.625	**0.677**	0.672	0.674	0.673

Table 4.7: Chicago per Month

Month/Model	Metric	MLP	LSTM	GRU	MIST	HAGEN	Attn	$\lambda = 0$	NoAttn	NewThr
August	Micro-F1	0.492	0.512	0.535	0.592	0.712	**0.722**	0.710	0.728	0.716
August	Macro-F1	0.430	0.435	0.442	0.581	0.669	0.691	0.677	**0.697**	0.684
August	AUC	0.505	0.500	0.498	0.581	**0.702**	0.692	0.679	0.698	0.686
September	Micro-F1	0.476	0.524	0.521	0.599	0.709	**0.722**	0.719	0.718	0.718
September	Macro-F1	0.420	0.440	0.432	0.585	0.660	**0.691**	0.688	0.687	0.686
September	AUC	0.501	0.502	0.501	0.589	**0.693**	0.692	0.689	0.688	0.687
October	Micro-F1	0.474	0.518	0.521	0.592	0.711	0.718	0.714	0.716	**0.721**
October	Macro-F1	0.422	0.432	0.425	0.579	0.665	0.686	0.683	0.683	**0.689**
October	AUC	0.503	0.504	0.497	0.579	**0.695**	0.686	0.683	0.684	0.689
November	Micro-F1	0.509	0.477	0.549	0.593	0.700	0.713	0.709	**0.718**	0.714
November	Macro-F1	0.426	0.403	0.444	0.581	0.657	0.682	0.677	**0.688**	0.684
November	AUC	0.501	0.501	0.506	0.595	**0.690**	0.682	0.677	0.687	0.684
December	Micro-F1	0.464	0.461	0.498	0.594	0.696	**0.721**	0.710	0.718	0.707
December	Macro-F1	0.405	0.391	0.401	0.582	0.650	**0.690**	0.678	0.688	0.675
December	AUC	0.499	0.496	0.500	0.602	0.687	**0.691**	0.679	0.688	0.676

Finally, we need to assess the monthly performance, shown in Tables 4.6 and 4.7, since previous frameworks primarily focus on it. The preliminary models, alongside MiST, follow the same distribution of results as before. However, HAGEN and the four proposed models differ from the individual categories' results. In the Los Angeles dataset, we see the model that uses transfer learning and the individual task's threshold to have more instances of better performance. The model with the attention layer comes second, then the single-task model. In this instance, not only the HAGEN model fail to outperform the other models, but there was also a significant difference between them, with a range of 0.1. In contrast to the Chicago dataset, it has an entirely different distribution of results, particularly regarding the performance of HAGEN, which appears to have specific instances of better performance, all in the AUC metric. Yet, the model with the attention layer performs better overall, but only in some months, and then follow both transfer learning models.

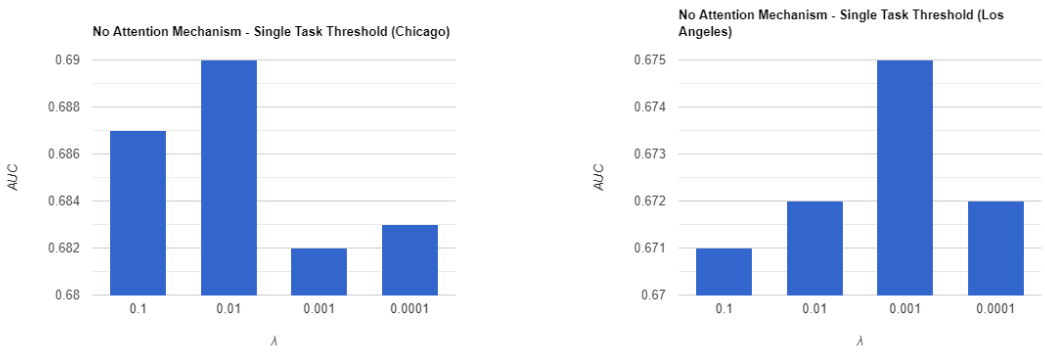
4.5 Hyper-Parameter Tuning



(α′) Chicago (β′) Los Angeles

Figure 4.3: The figures show the tuning for the model of transfer learning without the attention mechanism. Figure 4.3α′ shows the tuning for the λ value for Chicago for the month of December. Figure 4.3β′ shows the tuning for Los Angeles for the month of December.

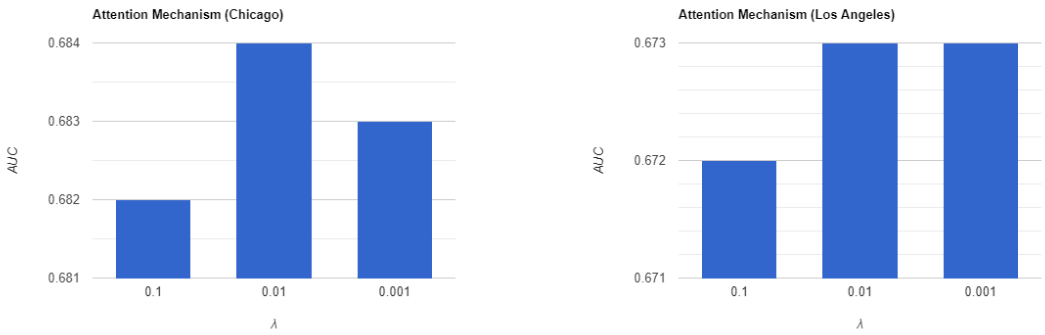
In the Figure 4.3 we present the results for the tuning of the value $\lambda \in \{1, 0.1, 0.01, 0.001, 0.0001\}$. We use AUC as the comparison metric. Although the differences are marginal and range from 0.001 to 0.004, the chosen value is 0.1 for Chicago and Los Angeles. It is also worth noting that the AUC scores of 0.1 and 0.0001 are the same in Los Angeles.



(α′) Chicago (β′) Los Angeles

Figure 4.4: The figures show the tuning for the model of transfer learning without the attention mechanism and the use of single task model's threshold. Figure 4.4α′ shows the tuning for the λ value for Chicago for the month of December. Figure 4.4β′ shows the tuning for Los Angeles for the month of December.

For the tuning of the model with no attention mechanism and single task's threshold, shown in the Figure 4.4, we discard the cases of $\lambda = 1$ and $\lambda = 0.0001$. Again the differences are slim and the chosen values are 0.01 and 0.001 for Chicago and Los Angeles respectively.



(α′) Chicago (β′) Los Angeles

Figure 4.5: The figures show the tuning for the model of transfer learning with the attention mechanism. Figure 4.5α′ shows the tuning for the λ value for Chicago for the month of December. Figure 4.5β′ shows the tuning for Los Angeles for the month of December.

Lastly, the model with the attention mechanism, shown in Figure 4.5, is tested with three different $\lambda \in \{0.1, 0.01, 0.001\}$ values, and tested for the AUC metric. The chosen value in both cases is 0.01.

Chapter 5

Conclusion

From the examination of Crime Forecasting arise the following conclusions. Firstly, it is clear from the results in the section that the spatiotemporal models with the GNNs offer the best performance of all models. The most outstanding model is HAGEN, as expected, since it is a state-of-the-art solution. We explore the correlation of the crime categories' distributions to expand this solution, yet it is still being determined whether or not the transfer learning models improve the performance. From the results in Section 4, we can deduce that the performance is improving in some cases without showing any transparent outstanding model. That leads us to question our hypothesis and the actual distribution of the original dataset since it did not bring down the model's performance to the point that we view it as noise. The grouping of the categories led to groups where certain occurrences did not match the group, implying that the original dataset has inconsistencies. The behavior of the metrics could be better when it comes to the transition from the per-category table to the monthly. Since the figures show an imbalanced dataset, our evaluation protocol can handle imbalanced datasets, but previous works may not consider that factor. Lastly, since Crime forecasting is a very studied subject in machine learning, it may be tricky to expand it. Future work may further examine the homophily ratio since it was the turning point for HAGEN. Specifically, a framework that can transfer learning not from crime categories but from the regions of the city with the homophily ratio showing the similarities between the regions. The tricky part is to host such a solution as it may need to learn and optimize hundreds of models simultaneously.

Bibliography

[1] Chao Huang, Chuxu Zhang, Jiashu Zhao, Xian Wu, Dawei Yin, and Nitesh Chawla. Mist: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting. In *The world wide web conference*, pages 717–728, 2019.

[2] Chenyu Wang, Zongyu Lin, Xiaochen Yang, Jiao Sun, Mingxuan Yue, and Cyrus Shahabi. Hagen: Homophily-aware graph convolutional recurrent network for crime forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4193–4200, 2022.

[3] Jiao Sun, Mingxuan Yue, Zongyu Lin, Xiaochen Yang, Luciano Nocera, Gabriel Kahn, and Cyrus Shahabi. Crimeforecaster: crime prediction by exploiting the geographical neighborhoods' spatiotemporal dependencies. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V*, pages 52–67. Springer, 2021.

[4] Xiangyu Zhao and Jiliang Tang. Crime in urban areas: A data mining perspective. *Acm Sigkdd Explorations Newsletter*, 20(1):1–12, 2018.

[5] Chris Couch and Annekatrin Dennemann. Urban regeneration and sustainable development in britain: The example of the liverpool ropewalks partnership. *Cities*, 17(2):137–147, 2000.

[6] Cem Oyvat. How urbanization affects the inequality in developing countries: A critique of kuznets curve. In *University of Massachusetts-New School University Economics Graduate Student Workshop*, pages 23–24, 2010.

[7] Kristian Behrens and Frédéric Robert-Nicoud. Survival of the fittest in cities: Urbanisation and inequality. *The Economic Journal*, 124(581):1371–1400, 2014.

[8] E Britt Patterson. Poverty, income inequality, and community crime rates. *Criminology*, 29(4):755–776, 1991.

[9] Bruce P Kennedy, Ichiro Kawachi, Deborah Prothrow-Stith, Kimberly Lochner, and Vanita Gupta. Social capital, income inequality, and firearm violent crime. *Social science & medicine*, 47(1):7–17, 1998.

[10] John Hagan. *Crime and inequality*. Stanford University Press, 1995.

[11] John Braithwaite. *Inequality, crime and public policy (Routledge revivals)*. Routledge, 2013.

[12] Adam Crawford and Karen Evans. Crime prevention and community safety. 2017.

[13] Clifford Robe Shaw and Henry Donald McKay. Juvenile delinquency and urban areas. 1942.

[14] Lawrence E Cohen and Marcus Felson. Social change and crime rate trends: A routine activity approach. *American sociological review*, pages 588–608, 1979.

[15] Ronald V Clarke. Introduction to the transaction edition. *The reasoning criminal: Rational choice perspectives on offending*, pages ix–xvi, 2014.

[16] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[17] Hongjian Wang, Daniel Kifer, Corina Graif, and Zhenhui Li. Crime rate inference with big data. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 635–644, 2016.

[18] Dennis M Gorman, Paul W Speer, Paul J Gruenewald, and Erich W Labouvie. Spatial dynamics of alcohol availability, neighborhood structure and violent crime. *Journal of studies on alcohol*, 62(5):628–636, 2001.

[19] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

[20] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[21] Yuting Mei and Fa Li. Predictability comparison of three kinds of robbery crime events using lstm. In *Proceedings of the 2019 2nd international conference on data storage and data engineering*, pages 22–26, 2019.

[22] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[23] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[24] Chao Huang, Junbo Zhang, Yu Zheng, and Nitesh V Chawla. Deepcrime: Attentive hierarchical recurrent networks for crime prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1423–1432, 2018.

[25] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[26] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[27] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[28] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

[29] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1234–1241, 2020.

[30] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

[31] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020.

[32] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[36] Crime data of 2015 in the city of chicago. `https://data.cityofchicago.org/Public-Safety/Crimes-2015/vwwp-7yr9`.

[37] Boundary information of chicago (2020). `https://data.cityofchicago.org/widgets/bbvz-uum9`.

[38] Census bureau (2020). `https://www.census.gov/`.

[39] Crosstown los angeles (2020). `https://xtown.la/`.

[40] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):1–28, 2015.

[41] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.