ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ
http://www.eee.uniwa.gr
http://www.idpe.uniwa.gr
Θηβών 250, Αθήνα-Αιγάλεω 12241
Τηλ: +30 210 538-1614
Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών
Τεχνητή Νοημοσύνη και Βαθιά Μάθηση
https://aidl.uniwa.gr/

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
DEPARTMENT OF INDUSTRIAL DESIGN AND PRODUCTION ENGINEERING
http://www.eee.uniwa.gr
http://www.idpe.uniwa.gr
250, Thivon Str., Athens, GR-12241, Greece
Tel: +30 210 538-1614
Master of Science in
Artificial Intelligence and Deep Learning
https://aidl.uniwa.gr/

Master of Science Thesis

Spatial-temporal crime prediction using Deep Learning

Student: Botsarakos Georgios
Registration Number: AIDL-0010

MSc Thesis Supervisor

Kasnesis Panagiotis
Lecturer

ATHENS-EGALEO, September 2022

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ
http://www.eee.uniwa.gr
http://www.idpe.uniwa.gr
Θηβών 250, Αθήνα-Αιγάλεω 12241
Τηλ: +30 210 538-1614
Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών
Τεχνητή Νοημοσύνη και Βαθιά Μάθηση
https://aidl.uniwa.gr/

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
DEPARTMENT OF INDUSTRIAL DESIGN AND PRODUCTION ENGINEERING
http://www.eee.uniwa.gr
http://www.idpe.uniwa.gr
250, Thivon Str., Athens, GR-12241, Greece
Tel: +30 210 538-1614
Master of Science in
Artificial Intelligence and Deep Learning
https://aidl.uniwa.gr/

Μεταπτυχιακή Διπλωματική Εργασία

Χωρική – χρονική πρόβλεψη του εγκλήματος χρησιμοποιώντας βαθιά μάθηση

Φοιτητής: Μποτσαράκος Γεώργιος
ΑΜ: AIDL-0010

Επιβλέπων Καθηγητής
Κασνέσης Παναγιώτης
Λέκτορας

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Σεπτέμβριος 2022

This MSc Thesis has been accepted, evaluated and graded by the following committee:

Supervisor	Member	Member
Panagiotis Kasnesis Digitally signed by Panagiotis Kasnesis Date: 2022.10.04 14:03:32 +03'00'	Digitally signed by Charalampos Patrikakis Date: 2022.09.29 08:56:51 +03'00'	Periklis Papadopoulos Digitally signed by Periklis Papadopoulos Date: 2022.09.29 12:42:23 +03'00'
Kasnesis Panagiotis	Patrikakis Charalampos	Papadopoulos Perikles
Lecturer	Professor	Professor
Electrical & Electronics Engineering	Electrical & Electronics Engineering	Electrical & Electronics Engineering
University of West Attica	University of West Attica	University of West Attica

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και (Μποτσαράκος Γεώργιος), Σεπτέμβριος, 2022

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον/την συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η Μποτσαράκος Γεώργιος του Δημητρίου. με αριθμό μητρώου 0010 μεταπτυχιακός/ή φοιτητής/ήτρια του ΔΠΜΣ «Τεχνητή Νοημοσύνη και Βαθιά Μάθηση» του Τμήματος Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών και του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Η εργασία δεν έχει κατατεθεί στο πλαίσιο των απαιτήσεων για τη λήψη άλλου τίτλου σπουδών ή επαγγελματικής πιστοποίησης πλην του παρόντος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Ο/Η Δηλών/ούσα
Μποτσαράκος Γεώργιος

Copyright © All rights reserved.

University of West Attica and (Georgios Botsarakos)
September, 2022

You may not copy, reproduce or distribute this work (or any part of it) for commercial purposes. Copying/reprinting, storage and distribution for any non-profit educational or research purposes are allowed under the conditions of referring to the original source and of reproducing the present copyright note. Any inquiries relevant to the use of this thesis for profit/commercial purposes must be addressed to the author.

The opinions and the conclusions included in this document express solely the author and do not express the opinion of the MSc thesis supervisor or the examination committee or the formal position of the Department(s) or the University of West Attica.

Declaration of the author of this MSc thesis

I, Georgios Demetrios Botsarakos with the following student registration number: 0010, postgraduate student of the MSc programme in "Artificial Intelligence and Deep Learning", which is organized by the Department of Electrical and Electronic Engineering and the Department of Industrial Design and Production Engineering of the Faculty of Engineering of the University of West Attica, hereby declare that:

I am the author of this MSc thesis and any help I may have received is clearly mentioned in the thesis. Additionally, all the sources I have used (e.g., to extract data, ideas, words or phrases) are cited with full reference to the corresponding authors, the publishing house or the journal; this also applies to the Internet sources that I have used. I also confirm that I have personally written this thesis and the intellectual property rights belong to myself and to the University of West Attica. This work has not been submitted for any other degree or professional qualification except as specified in it.

Any violations of my academic responsibilities, as stated above, constitutes substantial reason for the cancellation of the conferred MSc degree.

The author
Botsarakos Georgios

To my daughter Panagiota.

I would like to express my gratitude to my supervisor Dr. Panagiotis Kasnesis, for his guidance throughout the preparations of the present postgraduate thesis. I would also like to thank the Hellenic Police for providing crime data for the area of Attica. Finally, I would like to thank Dr. Patrikakis Charalampos and Dr. Papadopoulos Pericles who honored me with their participation in the three-member examination committee.

Abstract

Dealing with crime is one of the main purposes of a society. Therefore, the law enforcement authorities with the limited resources they have are trying to fight crime, which is characterized by dynamic spatio-temporal changes, by distributing their forces accordingly. In order to make this distribution optimal, significant assistance could be provided by predicting spatiotemporal changes in crime.

The use of artificial intelligence and specifically deep learning has been used in several spatio-temporal prediction tasks, based on historical data, such as movement prediction. The subject of this postgraduate thesis is the examination of the application of deep learning methods for the prediction of spatial and temporal criminal activity based on historical criminal data. In this context, different deep learning models, namely LSTM (Long Short-Term Memory), CLSTM (Convolutional LSTM), 3DCNN (Three-dimensional Convolutional Neural Networks) and GCNN (Graph CNN), are examined in order to evaluate which one can better capture spatial and temporal correlations of crime patterns and produce the most accurate predictions.

Historical data of property crimes that occurred in the cities of New York City and Attica were used to train and evaluate the above mentioned models. For the case of New York, a corresponding dataset was used that is freely accessible in the "open data" that New York provides on the internet. While for the case of Attica, a dataset provided by the Hellenic Police upon request was used for the purpose of this study.

In order to feed these models with the corresponding data each time, the data were modified into different structures such as matrices and graphs. Moreover, for the evaluation of the models various standard metrics were used (MSE, RMSE, MAE, Accuracy Score and Roc Accuracy Score) depending on the case.

Finally, the model found to provide the most accurate predictions was the CLSTM model, which was able to better capture the spatiotemporal correlations between historical crime data. This model is a combination of the operations of CNN and LSTM deep learning models. The crime predictions in the case of New York were more accurate than in the case of Attica due to the existence of a large percentage of noise in the crime data of the Attica region.

Keywords

crime prediction, deep learning, neural networks, spatial-temporal deep learning, real-time prediction.

Περίληψη

Η αντιμετώπιση του εγκλήματος αποτελεί έναν από τους κυριότερους σκοπούς μίας κοινωνίας. Συνεπώς οι αρχές επιβολής του νόμου με τους πεπερασμένους πόρους που διαθέτουν προσπαθούν να καταπολεμήσουν την εγκληματικότητα η οποία χαρακτηρίζεται από δυναμικές χωροχρονικές αλλαγές κατανέμοντας ανάλογα τις δυνάμεις τους. Για να γίνει βέλτιστη αυτή η κατανομή σημαντική συνδρομή θα μπορούσε να παράσχει η πρόβλεψη των χωροχρονικών αλλαγών της εγκληματικότητας.

Η χρήση τεχνητής νοημοσύνης και συγκεκριμένα βαθιάς μάθησης έχει χρησιμοποιηθεί σε αρκετές περιπτώσεις πρόβλεψης χωροχρονικών δεδομένων, με βάση ιστορικά δεδομένα, όπως είναι η πρόβλεψη της κίνησης. Αντικείμενο της παρούσας μεταπτυχιακή διατριβής αποτελεί η εξέταση η εξέταση της εφαρμογής μεθόδων βαθιάς μάθησης για την πρόβλεψη της χωρικής και χρονικής εγκληματικής δραστηριότητας με βάση χωρικά και χρονικά ιστορικά εγκληματικά δεδομένα. Στο πλαίσιο αυτό, εξετάζονται διαφορετικά μοντέλα βαθιάς μάθησης, συγκεκριμένα ΜΜ (Μακροπρόθεσμη Μνήμη), ΣΜΜ (Συνελικτική ΜΜ), 3ΔΣΝΔ (Συνελικτικά Νευρωνικά Δίκτυα Τριών Διαστάσεων) και ΓΣΝΔ (Συνελικτικά Νευρωνικά Δίκτυα Γράφων), προκειμένου να διαπιστωθεί ποιο από αυτά μπορεί να αποτυπώσει καλύτερα χωρικές και χρονικές συσχετίσεις της εγκληματικότητας και να παράγει τις ακριβέστερες προβλέψεις.

Για την εκπαίδευση και αξιολόγηση των μοντέλων χρησιμοποιήθηκαν ιστορικά δεδομένα εγκλημάτων κατά της ιδιοκτησίας που έλαβαν χώρα στην πόλη της Νέας Υόρκης και στην Αττική. Για την περίπτωση της Νέας Υόρκης χρησιμοποιήθηκε αντίστοιχο σύνολο δεδομένων που είναι ελεύθερα προσβάσιμο στα «ανοιχτά δεδομένα» που διαθέτει η Νέα Υόρκη στο διαδίκτυο. Ενώ για την περίπτωση της Αττικής χρησιμοποιήθηκε σύνολο δεδομένων που παρασχέθηκε από την Ελληνική Αστυνομία κατόπιν αιτήματος για τον σκοπό της παρούσας έρευνας.

Προκειμένου να τροφοδοτηθούν τα εν λόγω μοντέλα με τα αντίστοιχα δεδομένα κάθε φορά αυτά τροποποιήθηκαν σε διαφορετικές δομές όπως μήτρες, γράφοι. Ενώ για την αξιολόγηση των μοντέλων χρησιμοποιήθηκαν διάφορες τυπικές μετρικές ανάλογα την περίπτωση (MSE, RMSE, MAE, Accuracy Score και Roc Accuracy Score).

Τέλος το μοντέλο που διαπιστώθηκε ότι παρείχε τις ακριβέστερες προβλέψεις ήταν το CLSTM το οποίο μπόρεσε να αποτυπώσει σε καλύτερο βαθμό τις χωροχρονικές συσχετίσεις μεταξύ των ιστορικών εγκληματικών δεδομένων. Το εν λόγω μοντέλο αποτελεί έναν συνδυασμό των χαρακτηριστικών των μοντέλων Συνελικτικών Νευρωνικών Δίκτυων (ΣΝΔ) και μοντέλων Μακροπρόθεσμης Μνήμης (ΜΜ) βαθιάς μάθησης. Όσον αφορά τα σύνολα δεδομένων οι προβλέψεις στην περίπτωση της Νέας Υόρκης ήταν ακριβέστερες από την περίπτωση της Αττικής λόγω ύπαρξης μεγάλου ποσοστού θορύβου στα εγκληματικά δεδομένα της περιοχής της Αττικής.

Λέξεις – κλειδιά

Πρόβλεψη εγκληματικότητας, βαθιά μάθηση, νευρωνικά δίκτυα, χωροχρονική βαθιά μάθηση, προβλέψεις πραγματικού χρόνου.

Table of Contents

List of Tables

List of figures

Acronym Index

CNN: Convolutional Neural Network

CLSTM: Convolutional Long short-term Memory

GNN: Graph Neural Network

LSTM: Long short-term Memory

RNN: Recurrent Neural Network

GCNN: Graph Convolutional Neural Network

MLP: Multi-layer Perceptron

SVM: Support-vector Machine

KNN: k-nearest neighbors

ST: Spatio-Temporal

MASE: Mean Absolute Scaled Error

MLP: Multi-layer Perceptron

MSE: Mean Square Error

MAE: Mean Absolute Error

MSLE: Mean Squared Logarithmic Error

RMSE: Root Mean Square Error

RAE: Relative Absolute Error

RRSE: Root Relative Squared Error

LEA: Law Enforcement Agency

ML: Machine Learning

DL: Deep Learning

SAtt: Spatial Attention

Tatt: Temporal Attention

SFTT: Spatial Features Then Temporal

TFTS: Temporal Features Then Spatial

ParB: Parallel Branches

PAI: Prediction Accuracy Index

FBI: Federal Bureau of Investigation

1 INTRODUCTION

Crime has major impact in many aspects of society. The types of crimes (crimes against property, crimes against life, etc.) as well as the extent to which they occur in an area can greatly affect the social and economic life of the people living in that area. It is therefore a high priority for every state to implement specific strategies to reduce criminal activity as much as possible. In this effort of the state, could the application of technologies such as Artificial Intelligence and more specifically Deep Learning (DL) help to design and implement the best strategies to fight crime?

1.1 The subject of this thesis

LEAs (Law enforcement agencies) play the most important role in this effort to reduce crime. In particular, police can act either proactively or reactively to prevent crime. A police patrol can act proactively in the sense that the apparent presence of police officers at a point prevents would-be criminals from taking illegal action because of the higher risk of being arrested. While a patrol acts reactively when it is called to face an ongoing crime and to arrest the perpetrator on the spot [1]. Ideally, there could be a patrol present at every moment in every point of every area. Nevertheless, the resources of each Police, whether they are human or material, are limited. Therefore, it is necessary to plan the static and mobile Police forces routes every time to achieve their optimal performance. This problem is called “Police Patrol Routing Problem” and it is a very complex and difficult task to solve because of the many parameters that must be taken into account [1]. For example, in order to make the utilization of the Police forces as efficient as possible, the distribution of routes and patrol areas must be such that, on the one hand, the maximum possible prevention for dealing with criminal acts is achieved, and on the other hand, the response of police officers to criminal acts will be as fast as possible.

However, what if it were possible to predict with sufficient accuracy where and when an increase in criminal activity would occur or take place? Then surely, the “Police Patrol Routing Problem” would be much easier to solve. The solution could simply be to route the patrols in such a way that they are near the area that is going to see an increase in criminal activity at the time this occurs. According to the theory of “near repeat of crime”, similar types of crimes will be committed in close proximity at similar times [2]. In fact, it is argued that it is very likely that the same perpetrator acts around the same hours with the same method of action “modus operandi” in the same areas [3]. Therefore, forecasting the time when there will be an increase in crime will be a catalytic contribution to the maximum utilization of the police force to create routes and areas of responsibility of the patrols that will move in the areas where crime is expected to increase.

Unfortunately, the prediction of the time when there will be an increase in crime in an area based on previous spatial and temporal data cannot be done with simple artificial intelligence algorithmic methods and rules due to the large volume and variability of spatial and temporal data. This requires the use of DL methods that have been shown in the past to be able to discover high-level features that are not otherwise apparent in such tasks [4].

1.2 Aim and objectives

This study aims to examine the application of different DL algorithms and architectures, to predict the spatial and temporal criminal activity based on spatio-temporal historical criminal data. In this context, various DL architectures such as Long short-term Memory (LSTM), Convolutional Neural Network (CNN) and Graph Neural Network (GNN) as well as combinations of them were tested in the present thesis. Such models have shown satisfactory results in the past, in terms of their ability to perceive spatiotemporal features from historical data [5], [6]. Therefore, an attempt is made to study the kinds of architectural DL models that can be more accurate in predicting the occurrence of criminal behavior.

1.3 Methodology

In the present thesis, different complex DL models based on combinations of common DL algorithms were used, in order to examine which model architecture performs better in spatio-temporal crime prediction tasks. For the same purpose two datasets were used to train and evaluate the DL models. The first dataset consists of reports of crimes committed in New York and reported to the Police during the years 2006-2020 [7]. The second dataset was created from historical crime data of Attica-Greece during the years 2011-2021 provided by the Hellenic Police for the purpose of this study. In addition, spatial data from various open data sources were used to configure the datasets in appropriate formats to feed the models [8], [9]. Therefore, the examined DL models were fed with data from the aforementioned datasets in different formats depending on the model's architecture each time. Thus, this work does not only focus on the effectiveness of various DL algorithms on specific datasets, but also examines how different data transformations affect the prediction accuracy results.

1.4 Innovation

As mentioned above, for examining the effectiveness of the developed models, a dataset was created using Attica crime data provided by the Hellenic Police for the purpose of the current study. The provided data, among other things, consisted of offense occurrence timestamps, area and unnumbered street names for anonymization purposes. The data were converted to geographic coordinates so that they could be used by the models. Therefore, the possibility of using DL algorithms for the spatio-temporal prediction of criminal activity in areas of Attica based on historical data was examined in this study.

1.5 Structure

The current thesis is structured as follows: Initially in the Chapter 1 there is a description of the basic DL models used in the study as well as a literature review of the algorithms that have been used to solve the same or similar problems in the past. Then in Chapter 2 an analysis and description of the datasets used as well as the transformations and additions made to them are presented. Moreover, Chapter 3 presents the architecture and the way the models used in this work

were constructed. and Chapter 4 demonstrates and discusses the results of the used DL models. And finally, the conclusions obtained from the use of the models in the two data collections are analyzed.

2 CHAPTER 1: Literature Review – Basic Models

Crime is a societal phenomenon that exhibits fluctuations. Based on the principles of the theory of "almost repetition of crime" these fluctuations have both spatial and temporal dependencies. Therefore, in order to make a prediction of the future criminal activity of an area, it is necessary to take into account two types of features. On the one hand, temporal characteristics such as the times of occurrence of criminal activity and their correlations need to be considered. On the other hand, it is necessary to take into account spatial characteristics of historical data such as geospatial points of manifestation of criminal activity and their correlations[10].

2.1 Basic Algorithms

Different kinds of DL algorithms have been developed for spatial or temporal data or the combination of them, therefore, in the continuation of this subsection, some of the most basic DL algorithms, which were used in this study, are presented.

2.1.1 Recurrent Neural Network

When it comes to sequential type of data like timeseries in order for a model to achieve more accurate predictions it needs to "remember" past values of the given sequence. In that way the models would be able to capture periodical characteristics (i.e. patterns) of the given data. So, simple architectures as fully connected feed forward neural networks are unable to locate these characteristics. In order to deal with these kind of features Recurrent Neural Networks (RNNs) were introduced. RNNs are based on the idea of using repetitions in the neural network model for achieving memorization of state [11]. Thus, RNNs can be presented as shown in Figure 1 as an cyclic graph where the output of a neuron level is going also to be the input to same level creating cycles in the graph (i.e. enable the neurons take into account the output of the previous states combined with the input of the current state for the current output) [12]. In fact, as illustrated in Figure 1, an input sequence (i.e. x_{t-1}, x_t, x_{t+1}) is fed to the hidden layer after being multiplied with a weight tensor U and added to the result of the previous hidden layer (e.g., h_{t-1}) after being multiplied with another weight tensor V. This result is passed followed by a non-linear activation function constituting h_t (see equation (1)). Finally, the hidden neurons (e.g., h_t) are multiplied with weight tensor W and passed through an non-linear activation function to output O_t (see equation (2)), which can be passed as inputs to the next network layers in case of having a deep neural network.

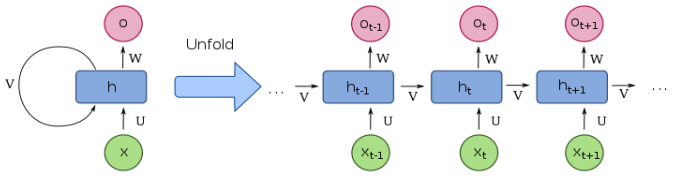

Figure 1: Recurrent Neural Network Abstract Architecture [88]

$$h(t) = f\big(Ux(t) + Vh(t-1)\big) \quad (1)$$

$$O(t) = g(Wh(t)) \quad (2)$$

Unfortunately, when it comes to RNNs there is a great drawback which can cause ultimately an issuein the accuracy of the predictions. This problem is called "Vanishing Gradient Problem". The source of "Vanishing Gradient Problem" is found in the back propagation through time process of the model. Specifically, during the process of training the model and at the stage of back propagating the gradients tend to decrease very quickly and eventually they became extremely low. So, it is difficult to capture long term dependencies with Vanilla RNNs.

The solution to this problem came with the introduction of Long short-term Memory (LSTM) neural networks [13]. LSTM uses memory "cells" consisting of gates which store short term activations. Specifically, as presented in Figure 2, LSTM memory cell uses sigmoid functions which produce an output between zero and one. This output is multiplied with the input of the cell thus zero means 'nothing should be let through,' and one means 'everything should be let through. This eliminates the "Vanishing Gradient Problem" because in every neuron we have a memory cell to store past information.

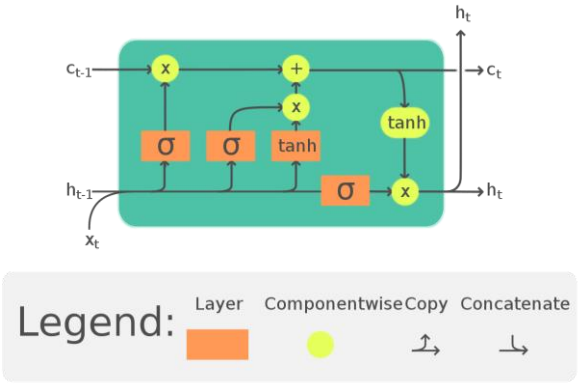

Figure 2: Long short-term Memory cell architecture [89]

2.1.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a kind of algorithm-architecture that is capable of perceiving autonomously patterns on the given data. Because of this ability their first applications were on image classification tasks. First appearance of multilayered CNNs was in 1989 and they were used to identify handwritten digits [14]. Specifically, the main idea of the CNN, in its simplest form is that the given input data are multiplied with matrices with weights called filters. These filters transform the input data and, in that way, tend to enhance or reduce specific features of the data. For example, if the matrix *I* shown in Figure 3 is an image and the blocks of the matrix represent its pixel values, then its regions are element-wise multiplied by a filter *K*(3*3) to produce the output *I*K* which may lead to image blurring or edges enhancement. That way by putting together many layers of filters (kernels), as in Figure 4, the produced model can learn in each layer specific patterns of the image or so-called features of the image, just by adjusting the filter values by the use of back propagation algorithm.

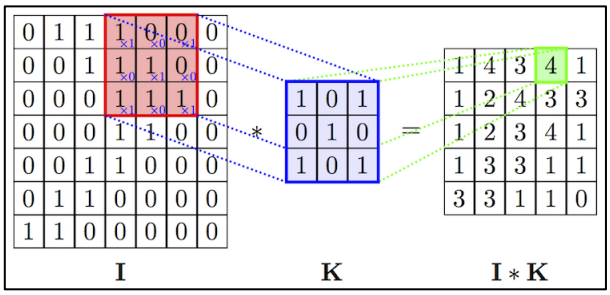

Figure 3: Convolution Procedure [90]

Therefore, for example in the first layer the model may learn the edges of an image and as it goes on deeper in its layers more abstract features can be learned, such as a face or ears if the given images have human faces or appearance of criminal activity the given images are criminal heatmaps [15][16]. In the example shown in Figure 4 an image with size (*n***n*) is given as an input to the CNN. This image is gray scaled, so it has only 1 channel. As the image's pixel values are multiplied in each layer by filters of (3*3) size each, they are produced stacked images, which at some point "hold" certain features of the given image regardless of their position in the image. Then in the third layer the images are given as an input to a MaxPooling layer with shape (2*2). MaxPooling is layer that propagates the maximum values of within the filter range to the next layer. Thus, the shape of the image is halved in this particular example, thus compressing the information and focusing on the most prominent features.

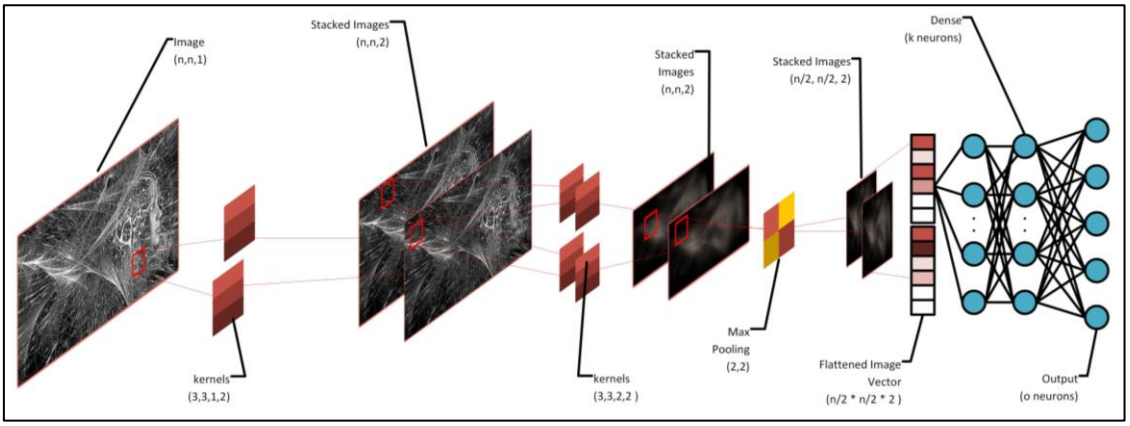

Figure 4: Convolutional Neural Network Example

Then the pixels of the image are flattened in order to be given as input to a layer of neurons where the final output is given by a set of (*o*) neurons that in a classification task should be equal to the classes to be predicted.

2.1.3 Graph Neural Network

Many real-world problems can be represented as graphs. Especially every problem that its parts can be represented as entities that have certain relationships between them can be represented as a graph. In ML, that nature of such kind of problems was used to the creation of Graph Neural Networks (GNNs).

In more detail, there has been proposed many different machine-learning algorithms that use graph representations to extract the spatial features. Some of them use algorithms such Random Walks in the graph while others use probability matrices to extract node representations [17],[18]. The most common methods used today in GNNs to extract nodes representations are convolution aggregations methods to aggregate messages passed through connections by the neighbors of the nodes [19]. Specifically in the basic approach in a GNN given a static graph it can represented as a set of nodes N connected by edges E as in equation (3).

$$G = (N, E) \tag{3}$$

By this representation, it is easy for the network to "understand" for each node which of the rest are the closest ones and therefore more related with the target node. So, in particular, in every layer of the GNN, for each node n the network uses an aggregate function (*f*) to aggregate the values (messages) of a set (*S*) of nodes that form the neighborhood of the target node (*n*) and then a combination function (*g*) is used to combine each node previous value with the aggregation result and update it with the new value as presented in equations (4), (5) and Figure 5 [20].

$$a_n^{(k)} = f^{(k)}(\{ c_n^{(k-1)} : n \in S(n)\}) \tag{4}$$

$$c_n^{(k)} = g^{(k)}(c_n^{(k-1)}, a_n^{(k)}) \tag{5}$$

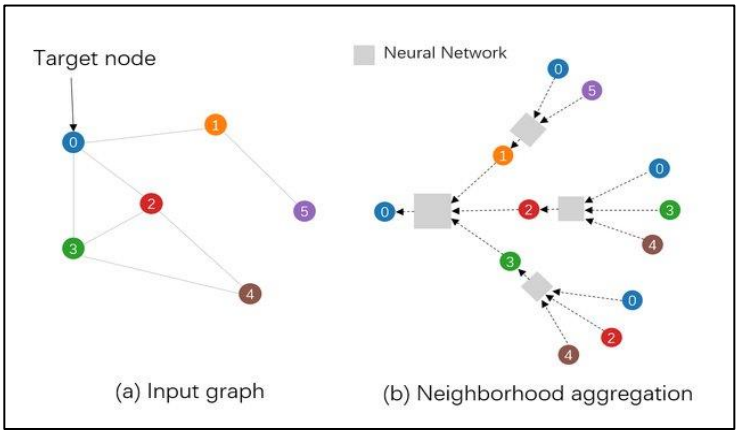

Figure 5: Aggregation process [91]

Therefore, the more layers the network has and the deeper it goes, the more the neighborhood of the nodes and the information that passes through the network increases. In the end, an activation function is used as in Figure 6 and we end up with each node being represented by a embedding which represents the nodes features [21].

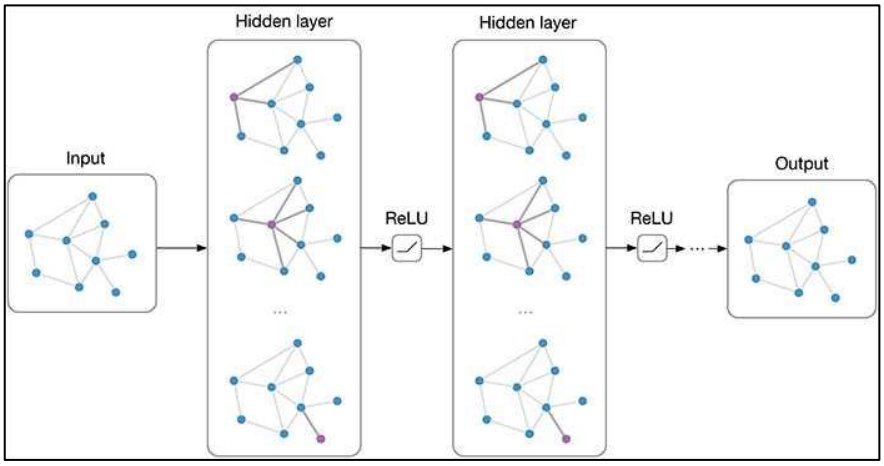

Figure 6: Graph Neural Network Example [92]

On the other hand, Graph Convolutional Networks (GCNs) use the same main idea as the GNNs which is the aggregated and combination functions but as Convolutional Neural Networks they use convolutions to create the node's embeddings. Specifically, GCNs use an adjacency matrix *A* of the nodes to create the graph and a degree matrix *D* that holds the degree of each node. The adjacency matrix is a matrix consisting of 0 for nodes that are not connected and 1 for those that are connected. In addition, each node has a vector of weights *W*. Therefore, when this matrix is

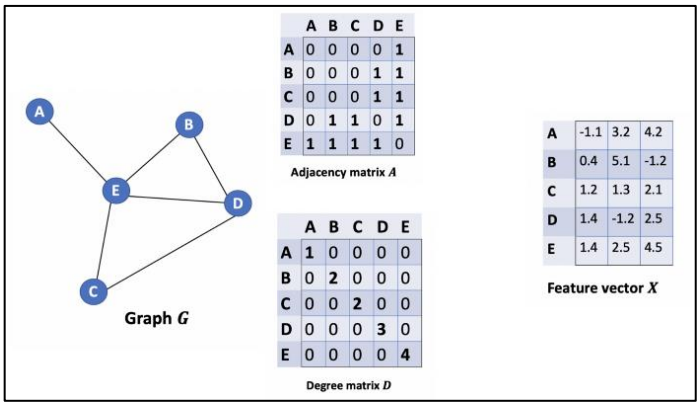

Figure 7: Graph Convolution Neural Network [93]

multiplied by the matrix of node features the representation of the graph with its attributes will be produced as is presented in the equation (6) and Figure 7 [4].

$$f\,(X\,,A)\;=\;ReLU(\,\widehat{D}^{-\frac{1}{2}}\hat{A}\widehat{D}^{-\frac{1}{2}}XW\,+\,b) \tag{6}$$

Then in the combination function the node's weights are multiplied with the aggregated features from its neighborhood creating its next feature vector. So, by stacking layers of GCNs you can create embeddings for each node that represent higher features of the task with collected information from more distant nodes [4].

2.2 Literature Review

As it is mentioned before Spatio-Temporal prediction of crime is based in theory of "near repeat of crime", which means similar types of crimes will be committed in close proximity at similar times [2],[3]. In fact, the "near repeat of crime" is based on other theories as "Routine Activity theory", "Rational Choice theory" and "Crime Pattern theory". According to these theories the criminal is considered that in order to act (s)he must take into account three variables. These variables, as in most human choices, are effort, risk, and reward. Effort means that the easier a target is to reach, the more likely it is to be chosen by the criminal. Risk is about the negative consequences of the choice and the probability that they will occur. In the case of the Criminal, for example, the risk concerns the possibility of being caught and the punishment (s)he will have in this case. Finally, the reward refers to the profit that an option has. In the case of the criminal profit can be the financial benefit (s)he can get from a theft for example. Thus in accordance with the theories of routine and pattern a criminal is choosing to act by a similar way ("modus operandi") in the same areas and in the same times. That way his risk and his effort are reduced because (s)he knows well the area in which (s)he acts as well as the way in which (s)he acts [22]-[24].

Therefore, in tasks such as the spatio-temporal prediction of criminal activity before the application of M and DL methods, but even today, additional statistical analysis algorithms and analysis models such as ARIMA are used to predict the future trend [25]-[28]. These kinds of algorithms can perform better in some cases than ML and DL algorithms in specific tasks. But when the prediction is based on a multitude of different features that cannot be classified then the use of ML algorithms and even more DL is required for accurate predictions. In the past many studies used ML and DL algorithms to predict crime activity. The data used for this task varies from just historical crime data in some cases while in other cases other types of data were combined them such as climate, population or even holiday data [4], [29], [30]. In the continuation of the chapter, applications of machine and DL algorithms are presented in tasks of predicting criminal activity. While in the end of the chapter a brief presentation of DL algorithms used in cases of spatiotemporal prediction similar to the case of crime prediction is given.

2.2.1 Machine Learning Algorithms used for Crime Prediction

When the features needed to be taken into account for the crime prediction task are such that can be extracted from the data and be fed to an algorithm, ML algorithms show astonishing results

of accuracy in their predictions. Therefore, in the literature many different ML algorithms have been implemented in different datasets for crime prediction.

Wu et al. [31] used random forest, neural network and Bayesian network in a crime YD County People's Procuratorate dataset. Their target was not spatio-temporal prediction but to predict the possibility of a person committing specific types of crimes by using as input features like education, gender, age and profession. In addition to this, Dash et al [32] approached the task of spatio-temporal crime prediction using analytical methods. The objective of this work was not just to predict the overall crime activity in Chicago city but the crime activity for various regions of Chicago city concerning the similar features of the areas like social structure, Schools in the area, police stations, Libraries, and emergency calls, and neighbor communities. To predict the number of crimes in a given community for a given year and month, they applied and compare three types of regression models, namely, polynomial regression, support vector regression, and auto-regressive model.

McClendon et al. [33] also used linear regression, additive regression, and decision stump algorithms to predict the crimes in Mississippi state using a dataset provided by the University of California-Irvine. The dataset consists of socio-economic data from the 1990 Census, law enforcement data from 1990 Law Enforcement Management and Admin Stats survey. To evaluate their results they used Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), Root Relative Squared Error (RRSE) and correlation coefficient. The best result was shown by linear regression model.

Yuki et al [34] used random forest, decision tree and different ensemble methods such as extra trees, bagging and AdaBoost on the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system to predict the crime occurrences. The features they used to train the algorithms are spatial features like Location Description, Block, Location, Latitude, Longitude and temporal features like Month, Day, Hour, Minute and Second that a crime occurred.

Safat et al [25] tested the performance of many, different, ML algorithms such as logistic regression, decision trees, random Forest, MLP (Multi-layer Perceptron), Naïve Bayes, SVM (Support Vector Machine), XGBoost and KNN (K-Nearest Neighbors) to the task of predicting criminal activities based on historic data. They used crime datasets of Chicago and Los Angeles to test and evaluate the performance of the algorithms. These datasets among others consist of timestamps, locations and types of the crimes occurred from years 2001 to 2019 and from years 2010 to 2018. While they evaluated the performance of the algorithms based on the metrics accuracy, precision, recall and F1-score. From the implemented algorithms XGBoost which is the most complex one showed better results with 94% and 88% accuracy on both the Chicago and Los Angeles datasets.

Zhang et al. [29] used a property crime occurrence dataset of a China town for years 2014-2017, for their predictions. They split the town in 150m x 150m grids and in each block, they

summed the crime occurrences on two weeks periods. Additionally, they used a dataset with points of interest such as department stores which they assigned to each block of the grid. In addition, they assigned to each block the information of the density of the road network, while they scaled the crime occurrences using MinMaxScaler to the range [0,1]. They implemented KNN, random forest, SVM and Naïve Bayes ML algorithms to predict the next two weeks crime occurrences considering thirteen units of two weeks past crime occurrences. The best performance among the ML models was resulted by the random forest algorithm.

2.2.2 Deep Learning Algorithms used for Crime Prediction

In cases where the complexity of the primary data is great and the prediction is required to be based on high-level characteristics of the data which cannot be easily extracted then the use of DL algorithms in general and specifically in the task of spatio-temporal crime prediction is indicated [35]. Therefore, in the continuation of the present sub-chapter, a review is made of the cases where DL algorithms were used for crime prediction.

To begin with Ramirez-Alcocer et al [36] used 50 units of LSTM layers on Chicago crime dataset consisted of features like district and an Federal Bureau of Investigation (FBI) coding reporting system to predict if for a specific crime the criminal will be arrested.

Safat et al. [25] in addition to ML algorithms also tested the performance of a DL algorithm on the same datasets of Chicago and Los Angeles mentioned in the previous chapter. The algorithm they used was a LSTM. For the use of LSTM, the given datasets were preprocessed to reduce noise and transformed into stationary. The model was trained for 40 epochs and with batch size 33 and 31. To evaluate the performance of the LSTM model they used RMSE and MAE metrics in which they obtained results of 12.66 and 11.70 respectively for the Chicago case and 8.78 and 6 respectively for the Los Angeles case.

Krishnan et al. [37] used Recurring LSTM networks to predict total crime occurrences in an area. For their model they used Adam optimizer [80] and ReLU activation function. They also scaled the counted crimes in range [0,1] using keras MinMaxScaler. To evaluate their results they used RMSE metric. Thus, measuring the total number of offenses and not one area and ignoring the exact point where they took place, they only made a temporal prediction of the offences(i.e., dealt the data as a time series).

Wang et al. [38] worked on a dataset consisted of crimes occurred in Atlanda between the years 2009 and 2016. They implemented a LSTM neural network with 32 neurons. Regarding the shape of the data used to train the model, they split the data into grids and tested random combinations of cell size and number of days to predict next-day crime. The combination that had

the best results was choosing a 0.05 degree cell length and width of the coordinates and using 50 days to predict the next one.

Zhang et al. [29] besides using ML algorithms also worked with Vanilla CNNs and LSTM Networks. And in this case as in ML they used the historical crime data of a Chinese city together with static data of the city such as points of interest where there is increased traffic flow and the density of the road network.

However, the use of LSTM can only perceive the temporal correlation of the data and fails to perceive its spatial correlation. Thus, more complex DL models were developed which can perceive the spatiotemporal correlation of the data.

For example, Stalidis et al [35] studied the application of DL algorithms for the spatio-temporal prediction of crime. They compared the performance of 3 different DL architectures. These three architectures differed in the order in which they extract the spatial and temporal features from a historical crime dataset. In the first architecture Spatial Features Then Temporal (SFTT) they used popular DL models in order to extract the spatial features of the data and then LSTM in order to extract the temporal features. In their second architecture Temporal Features Then Spatial(TFTS), they first extracted the temporal characteristics and then the spatial ones. While in the third architecture Parallel Branches (ParB) the extraction of the spatial and temporal characteristics was done in parallel using connections to two modules each of which extracted the spatial and temporal characteristics respectively. Specifically, they tested the performance of convolutional neural networks like VGGNet [84], ResNet [46], FastMask [85] and FastResidualMask as modules of their models to extract spatial features. The CNN modules consist of MaxPooling operations, Average Pooling Operations, Convolution Operations and concatenations. They tested their models in many different open crime datasets of cities like, Seattle, Minneapolis, Philadelphia, San Fransisco and Metropolitan DC. The data was reformed in spatial grids of size (16 x 16) and (40 x 40), as it is the most common approach and in each block they summed the crime occurrences. To evaluate their results they used precision, recall, F1-score and Prediction Accuracy Index (PAI) [85] metrics. The first architecture (SFTT) produced the most promising results.

Stec et al. [39] also used DL algorithms to predict the next day crime occurrences from historical crime data of the cities of Chicago [40] and Portland, partitioned in spatial grids. They also combined the crime data with weather and population data in order to have more accurate results. To evaluate their results, they used the Mean Absolute Scaled Error (MASE) metric because of the sparsity of the data. The algorithms they implemented for the predictions were a Feed Forward Network, a CNN, a RNN and a combination of CNN with RNN to predict at the same time spatial and temporal aspects. The count in each grid cell was classified in ten classes depending on the range of the value.

Ye et al. [41] proposed a model consisting of coherent neural networks with various connections and connections between them called DIRNet. DIRNet's main parts are ResUnits and

IncUnits. ResUnits as it was mentioned before are CNN layers connected with skip connections that help which help to eliminate the "Gradient Vanishing Problem" when the network is too deep. IncUnits or Inception Units on the other hand are being created by stacking asymmetric convolution layers in three branches to capture the different type of high featured of the data. They tested their model using the New York City crime dataset and the results showed that it's performance outperformed other DL and ML models like SVM, Rnadom Forest and ST-ResNet when the time window surpasses the 100 days. In order to have more accurate results they combined the crime dataset with the 311 dataset of New York City. The 311 dataset consists of citizen complaints about damage, cleanliness and damages in the area.

Esquivel et al. [42] proposed a Convolutional Long-Short term Memory neural network (CLSTM) for crime prediction in a dataset consisted of theft crimes in the city of Baltimore during the years 2016 to 2018. The model is consisted of 16 filters of Convolutional layers with Keras TimeDistributed Wrapper to in order to perform the operations on all levels of the input data. Then the model has a max Pooling layers reducing the size of data in half. Then the output is flattened and fed to 2 layers of LSTM unit which produce the final output. Then input of the model is 8*8 shaped grids of the city of Baltimore consisting of the number of crimes occurred in each block of the grid. The output has a shape of 64 *1 which are the blocks flattened. Therefore, the network is given stacked grids of the city each grid represents the number of crimes occurred in a day and the model predicts the next day crimes. The best results are shown on 4 days input which resulted in 0.8923 accuracy and 0.6079 AUC-ROC.

Wang et al. implemented a DL spatio-temporal architecture called ST-ResNet [43],[44]which is first created and implemented by Zhang et al. to the task of predicting crowd flows in the city [45]. The model consists of 3 main modules identical modules. Each module main architecture consists of a 1-dimensional CNN layer, six layers of Residual Convolution Units and a 2 dimensions CNN layer. Residual Units and Residual Neural Networks generally are based on the idea of exept from feeding the input in the current layer they also use a "skip connection" to forward the input to a next layer. As it is shown to equation (7) every layer (l) has an output $X^{(l+1)}$, input $X^{(l)}$ and a residual function F that skips one neuron.

$$X^{(l+1)} = X^{(l)} + F\,(X^{(l)}) \qquad (7)$$

The central idea of the residual learning is to learn the additive residual function F with respect to $X^{(l)}$ [46], [47]. Therefore, a network may have several layers and since the information also passes through another route and is connected to subsequent layers, not much information is lost during the back propagation process. Then the outputs of the 3 modules are fused using learnable weight matrices and the output is combined with the output of a 2 layer fully connected module which was fed with weather and holiday type of data.

The abovementioned three modules have to extract different types of features. So each module is fed with different aggregations of the dataset. The first one needs to extract the crime

trend so it is fed with weekly grouped data. The second one needs to extract the crime period, so, it is fed with daily grouped data and the third one needs to extract the nearby features so it is fed with hourly grouped data. For an optimization loss function they choose Adam optimizer, and they resulted in 80.87-84.78 accuracy and 0.184 RMSE.

Another way to approach problems that require the representation of space in an algorithm, such as spatiotemporal crime prediction, is to represent it using graphs. This is how types of graph neural networks can be implemented to represent the spatial aspect of the problem. In this context Han et al. [4] proposed an integrated model consisted of Spatio-Temporal Graph Convolution Network (ST-GCN) modules and LSTM modules for predicting crime. ST-GCN modules consisted of a GCN module and a ST-ResNet module. The ST-GCN was used to capture spatial correlation between the data and the LSTM module was used to capture temporal correlations. In the proposed model they used 3 ST-GCN components to capture different aspects of the data. The first ST-GCN component was fed with 3 years, before the day to be predicted, data to capture the trend. The second one ST-GCN component was fed with 3 weed years data, before the day to be predicted, to capture the periodic correlations and the third one was fed with 3 days data, before the day to be predicted, to capture the nearby correlations. Then the outputs were fused and fed to the LSTM module consisted of 5 LSTM layers with ReLU activation function. To evaluate the performance of their model they used RMSE and Mean Absolute Percentage Error (MAPE) metrics. The dataset used is Chicago, which was divided into regions according to its communities. Finally, their model predictions were evaluated in Chicago dataset obtaining 0.39 MAPE and 1.03 RMSE which outperformed other models like Ridge, random forest and LSTM.

2.2.3 Algorithms used for Spatio-Temporal Tasks

Apart from crime prediction, ML and DL algorithms are used in many other studies where the problem has a spatiotemporal structure like traffic prediction or human activity recognition [48], [49]. In the continuation of the present chapter, a brief presentation of cases of using DL models for the prediction of data with spatiotemporal structure is made where, due to the similarity of the projects, they can also be used for the spatiotemporal prediction of crime.

For example, similar to the crime prediction task, Nikparvar et al [50] used pure LSTM neural networks in the task of predicting simultaneously spread of a pandemic of Covid-19 in many states. While similar implementations have been made in the traffic sector. More specifically, usually in the case of vehicle motion prediction, measurements from motion sensors that are scattered in various places are used as input. These measurements refer to maximum movement speed characteristics. For example, a simple LSTM model is used in the past to predict the time needed to travel from one point to another [51]. Thus, the similarity of the characteristics of such studies with that of the spatio-temporal prediction of criminal activity can be easily perceived,

since in both cases for the prediction an attempt is made to create models that can extract the temporal correlation of the data on the one hand and the temporal correlation on the other.

In order to capture contemporary the spatio-temporal correlations on traffic flow and predict the traffic, Guo et al [48] proposed a novel DL model: Attention based Spatial-Temporal Graph Convolution Network (ASTGCN). The model is composed by three modules with the same architecture. Each module is taking as input different time series segments to capture recent and longer time period patterns at the same time as was done in the aforementioned case of called ST-ResNet [43], [44]. Each module is composed by two Spatio Temporal blocks with residual connections in order to have more accurate results. A Spatio Temporal block contains a SAtt (Spatial Attention) module, a TAtt (Temporal Attention) module, and a GCN module. The SAtt and the TAtt modules are using attention mechanisms. The first one captures spatial correlations while the second one captures temporal correlations. Each of these attention mechanisms produces a correlation matrix which in each cell contains the degree of correlation between nodes or time points respectively. A GCN module is then used (see subchapter 2.1.3) to capture the spatial correlations between the nodes of the graph. Finally, the outputs of the modules are merged together to produce the final output of the module after passing through a fully connected layer.

The CLSTM model mentioned above to be used for forecasting theft crimes in the city of Baltimore is used also in other tasks like the prediction of rainfalls in a certain area [52] or the prediction of demands of travel in an area [53]. The main difference between these two cases is that in the first study the CLSTM modules of their model were stacked while in the second case of travel demands prediction they used three identical separate modules which were fed with different groupings of the data (closeness, period and trend) as it was done in the case of ST-GCN mentioned in the previous subsection. A CLSTM model was also used in a Copenhagen Area dataset to predict the time needed to travel between links with busses [54]. In that case the input was two-dimensional matrices. The first dimension of the matrices contained the time required to pass through different links [1-*n*] while the second dimension of the matrices contained the different timesteps [1-*k*]. An encoder-decoder type implementation of CLSTM model proposed by Shi et al. [55] was used in order to capture more efficiently higher features. In the encoder module of the model two CLSTM layers were stacked while in the decoder module two CLSTM modules were also stacked. The final output of the model was given by a fully connected layer.

Another spatio-temporal model architecture that was used in traffic and transport domain was that of Graph Convolutional neural networks [56]-[58]. More specifically, Yu et al proposed a STGCN implementation for the prediction of the traffic in an area [56]. The proposed model consists of stacked Spatio-Temporal Convolutional Blocks. Each block is created by two Temporal Gated Convolution Blocks surrounding a Spatial Graph Convolution Block connected also by residual connections. The Temporal Gated Convolution Block contains a 1-D causal convolution with a width-Kt kernel followed by gated linear units (GLU) as a non-linear function. The Spatial Graph Convolution Block algorithm follows the concept of the Graph Convolutional Neural Network (described in subchapter 2.1.3).

Geng et al. proposed a novel DL model called spatiotemporal multi-graph convolution network (ST-MGCN) to address the problem of ride-hailing demand forecasting [57]. The model consists of three main identical modules each module contains a contextual gated recurrent neural network (CGRNN) block and GCN network block. The contextual gated recurrent neural network (CGRNN) is using RNN blocks and a channel-wise attention mechanism to extract the temporal correlations. The final output of the model is created by aggregating the output of the three modules. But each module is fed with graphs representing the regions of the given areas with different correlations among them. Specifically, the first module takes as input graphs whose nodes are connected according to their proximity, the second module takes as input graphs whose nodes are connected according to their similarity of Points of Interest (POI) in the areas and the connections in the third module's graphs are made concerning the transportation connectivity.

Finally, Wang et al. proposed a spatial temporal graph neural network framework mainly consisting by spatial graph neural network (S-GNN) layers, GRU layers and the transformer layers, to predict traffic [58]. Firstly, the S-GNN layers are fed with graphs whose nodes are connected according to their proximity in order to capture the spatial relations between the nodes of the graph. In this task the nodes represent traffic sensors. Secondly the output for each node is passed as input to the gated recurrent unit (GRU) layer which is capturing the temporal relations among the nodes. Finally, the model uses transformer layers which consist of multi-head attention layer [59], with batch normalization procedures and residual connections, that help to capture long term temporal correlations among the nodes.

3 CHAPTER 2: Datasets Analysis

Spatio-temporal crime prediction using artificial intelligence, as presented in the literature review in the previous chapter, is mainly approached as a supervised learning problem. This means that in order to achieve the desired accuracy results in the final predictions it is necessary to use appropriate annotated datasets in addition to the development of the appropriate algorithms. Specifically, for the spatiotemporal prediction of crime, the data needs to be spatially and temporally labeled as a minimum. Such data have been made open-sourced online by various cities in an effort to publicize the data of government agencies, to enhance transparency and research [60]. For this study two crime datasets was used in order to train, evaluate and compare DL algorithms. The first dataset is the NYPD Complaint Historic dataset, available online on the open data repository of New York City [8]. The second dataset is a dataset created by data provided by the Hellenic Police for the purpose of this study. In the continuation of this chapter, a presentation and an analysis of the data in question is made. A presentation of the methods used to transform the data into structures suitable each time for use by the DL algorithms is included, as well as other datasets used for these formations.

3.1 NYC Dataset

The NYPD Complaint Data Historic dataset [7] is a dataset publicly available in the open data of New York City repository [8]. The dataset consists of data concerning all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD) from 2006 to the end of year 2020. The dataset contains among others time, date, coordinates, and type of a crime occurrence.

3.1.1 Features

The columns of the dataset used for the present study are presented in Table 1. In particular, the latitude and longitude fields were used in order to be able to determine the places where the crimes were committed. The coordinates are provided in EPSG 4326 encoding which follows the World Geodetic System 1984 standard established and maintained by the United States National Geospatial-Intelligence Agency since 1984 [61]. The start and end dates and times of the crime (*CMPLNT_FR_DT, CMPLNT_FR_TM, CMPLNT_TO_DT, CMPLNT_TO_TM*) were also used to determine the time limits of the occurrence of the crimes. While the separation of each crime type was made based on the scope of the description of that type (*OFNS_DESC*) in order to be able to train the algorithms with data containing only related offense types.

Table 1: NYPD Complaint Data Historic dataset columns description

COLUMN NAME	**COLUMN DESCTIPTION**
CMPLNT_NUM	Randomly generated persistent ID for each complaint

CMPLNT_FR_DT	Exact date of occurrence for the reported event (or starting date of occurrence, if *CMPLNT_TO_DT* exists)
CMPLNT_FR_TM	Exact time of occurrence for the reported event (or starting time of occurrence, if *CMPLNT_TO_TM* exists)
CMPLNT_TO_DT	Ending date of occurrence for the reported event, if exact time of occurrence is unknown
CMPLNT_TO_TM	Ending time of occurrence for the reported event, if exact time of occurrence is unknown
OFNS_DESC	Description of offense corresponding with key code
Latitude	Midblock Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)
Longitude	Midblock Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)

3.1.2 Exploratory Data Analysis

The NYPD Complaint Data Historic dataset in general consists of 7,375,993 instances of crime occurrences. Regarding the data contained in the particular dataset, as shown in the following graph there are 17,339 instances where there are no coordinates for the spot of the crime, 18,823 instances where the type of crime is not available, 655 instances where there is no starting

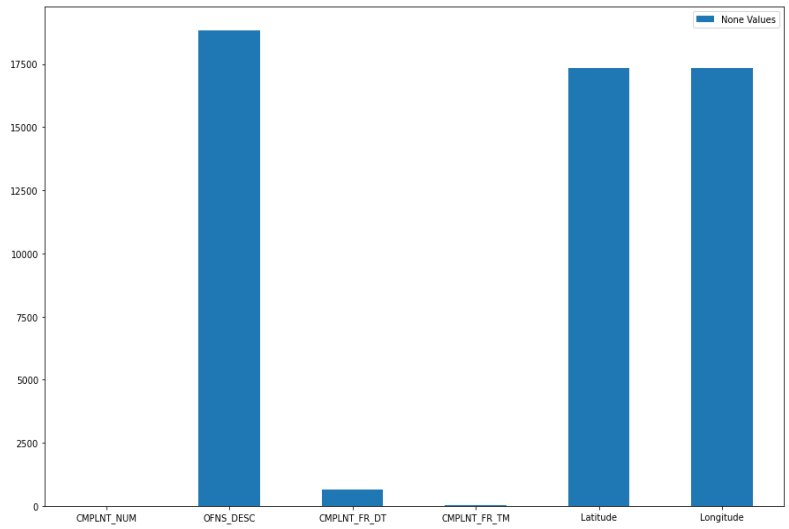

Figure 8: Empty Values in NYPD Complaint Data Historic dataset

date of the crime and 48 instances where there is no starting time for the crime. While there are 1,704,204 and 1,699,541 instances, where there is no ending date and time of the crime respectively.

These features are the main features of the dataset that were used as it is mentioned above and, therefore, the instances that are empty were removed from the data. The total number of different types of offenses present in the data is 71. Figure 9 presents the 30 most frequently occurring offenses in NYPD Complaint Data Historic dataset. As shown in the said figure property crimes such as theft, robbery and burglary have the cumulative highest incidence. Specifically, these crimes constitute the 33% of all crimes. Moreover, these types of crimes follow the theory of "near crime repeat" to a greater extent due to their characteristics and for this reason this study is mainly concerned with the prediction of such crimes.

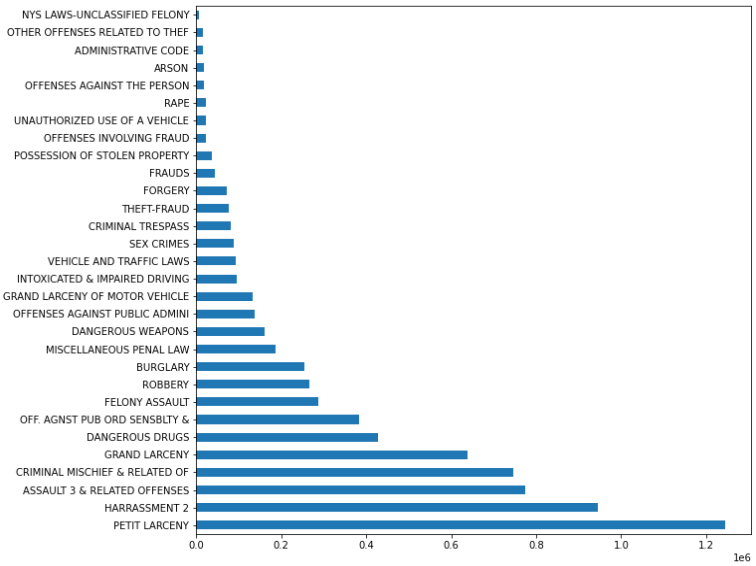

Figure 9: The 30 most frequently occurring offenses in NYPD Complaint Data Historic dataset

In terms of the spatial part of the data, New York City is located between the coordinates -74.30 and -73.70 Longitude and 40.50 and 40.90 Latitude, but as seen in Figure 10 (on the left) there are incorrect entries that are far outside the city space. Therefore, these values must be removed from the data otherwise, they will introduce noise into the data making predictions difficult. Therefore, by restricting the data so that the coordinates are between the New York City limits as mentioned above, the data is obtained as shown in Figure 10 (on the right). These results in the overall final heat map shown in Figure 11. This map shows the total number of offenses for the time period from the year 2006 to the year 2020, for the entire city of New York. As can be seen on the map, the total number of crimes in the areas is mostly at the same levels, while areas

with low levels (darker colors) of crime appear. Furthermore, Figure 12, where the total number of crimes per day in the city is presented, a periodicity appears over the course of each year.

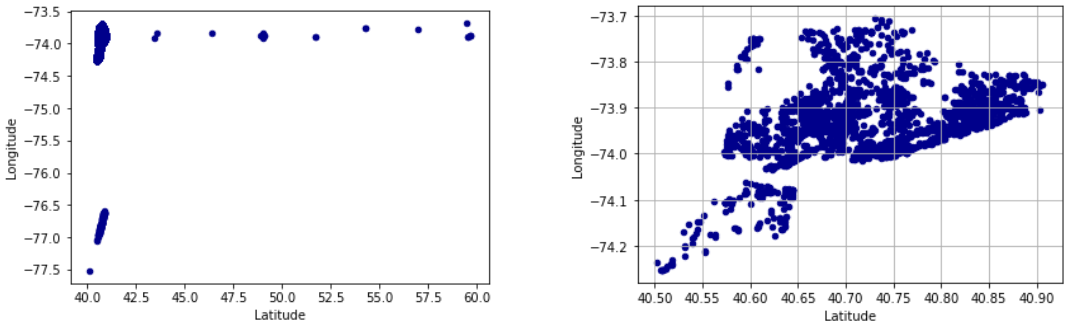

Figure 10: New York City crime dataset – crime coordinates (left) and crime coordinates cleaned (right)

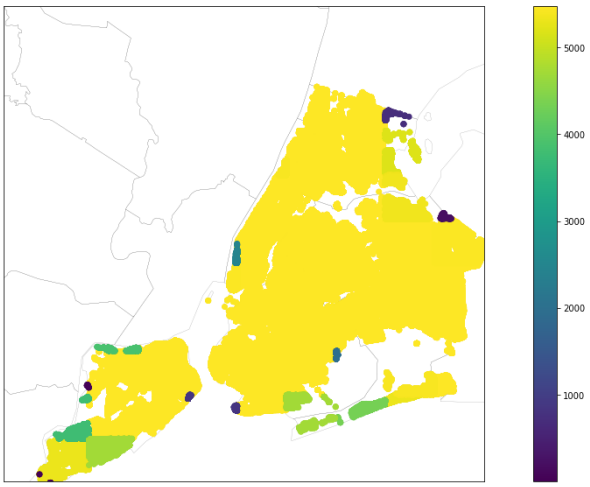

Figure 11: New York Crime Heat Map

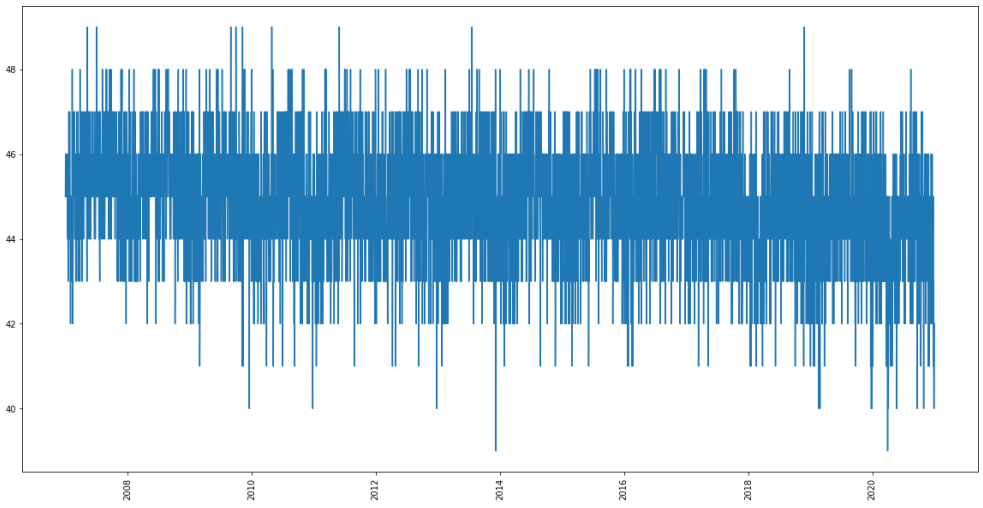

Figure 12: New York Crimes summed by day

3.1.3 Transformations – Use of other Datasets

New York City covers an area of 1,213 km² [62]. Therefore, the scope is too large for crime data to be directly used to predict crime. So, to evaluate the prediction algorithms and, also, to to use the algorithm predictions by the Police, the dataset must be divided into regions that have some meaning to the authorities and the predictions will be made concerning those regions. Consequently, a dataset with the boroughs and the neighborhoods of the NYC [63] was used in order to categorize the data based on their coordinates in respective neighborhoods and areas. More specifically, New York City is composed of five boroughs ('Bronx', 'Queens', 'Staten Island', 'Brooklyn', 'Manhattan') as they appear in different colors in Figure 13 and each borough is divided neighborhoods. While the geographical area covered by each borough of Manhattan is shown in Table 2. Thus, this dataset contains for each neighborhood the information about which boroughs it belongs to and as well as the coordinates of the area it encloses. The coordinates are given in the form of a polygon with points forming the boundaries of each neighborhood.

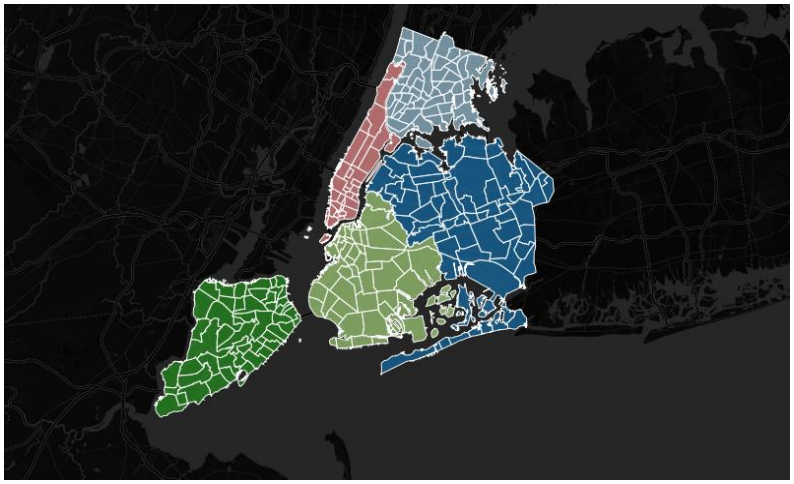

Figure 13: New York City divided in boroughs and neighborhoods [71]

Table 2: Geographical Coverage of New York Boroughs

Borough-Area	**Area(Km^2)**
Manhattan	59.1
Bronx	110
Queens	280
Brooklyn	180
Staten Island	152

In order to categorize the instances of the NYPD Complaint Data Historic dataset the Geopandas library was used [64]. Geopandas library is an open source library that extends the pandas library [65] and it is suitable for geocoded data analysis. Geopandas uses geometric shapes to manipulate geocoded data. Geopandas expects from the datasets to have a column called "geometry" with the shape of each object defined by coordinates. Thus, when it comes to a point on the map, the shape is defined by a "Point" which includes the geographic longitude and latitude of the point, while when it comes to an area the shape is defined by a "Polygon" which includes the geographic coordinates of the points making up the boundaries of the area, as in the case of NYC neighborhoods dataset.

Therefore by converting spatial data of the NYPD Complaint Data Historic dataset into "Geometry Points" it was possible to classify each instance of the dataset, by the neighborhood

and borough where the crime took place, by checking in which "Geometry Polygon" of the NYC neighborhoods dataset it is contained. By that way it was possible to perform spatial join on the two datasets, creating a new dataset with the information of neighborhoods and boroughs to work with. With this procedure, incorrect records of points which were not within the boundaries of the whole of New York and had not been removed in a previous stage, were removed.

Regarding the temporal characteristics of the dataset, the data of the date and time of the crime were merged creating timestamp index, based on which the data were sorted so that it was possible to use them as time series for their later use. During the creation of the time index, incorrect entries in the date fields were also removed. That is, dates prior to the year 2006 and later than the year 2020.

3.2 Attica Dataset

As it is mentioned above, for the purposes of this study, historical crime data, for the region of Attica, were provided by the Hellenic Police, following a relevant request. The data concern crimes against property such as thefts, burglaries and robberies that took place, in the Attica region, during the period 01/01/2011-21/12/2021. The provided data by correspondence with the NYPD Complaint Data Historic dataset contains crime categorization as well as spatial and temporal determinations, so that the effectiveness of the prediction algorithms on the two datasets can be compared. The source of the data is similar to that of the NYPD Complaint Data Historic. That is, in this case too, it is about historical data of citizen complaints about crimes.

3.2.1 Features

Regarding the features of the data provided by the Hellenic Police, these are the ones shown in the Table 3. The features *crime_type*, *type_stat_crime_type* and *stat_crime_type* concern the type and criminal categorization of the crime as well as its object. The features *time_from* and *time_to* are the starting and the ending time of the crime respectively. Specifically, in case that the crime had a known duration, fields *time_from* and *time_to* represent the start and the end time of the crime, while in case the exact time of the crime is not known, these features represent the possible time period during which the crime took place. Finally, the features *perif_enot*, dimot_enot and street are the description of the district, municipality and street where the crime was committed.

Table 3: Attica historic crime data features description

FEATURE NAME	**FEATURE DESCTIPTION**
ID	Randomly generated persistent ID for each complaint
crime_type	Exact crime type. The article of the criminal code violated by its description

type_stat_crime_type	Type of Statistical Categorization of the Offense. It mainly concerns the object on which the offense was committed. (e.g. bicycle theft)
stat_crime_type	Statistical categorization of the offence. More general category in which the offense falls (e.g. theft-burglary)
time_from	Starting timestamp of occurrence for the reported event
time_to	Ending timestamp of occurrence for the reported event
country	The country is Greece for every instance
perif_enot	Name of Regional unit where the offense was committed
dimot_enot	Name of municipal unit where the offense was committed
street	Name of the street where the offense took place

As can be seen in Table 3, the dataset does not include any street numbers, which would determine the exact spot of the crime. However, the street number was omitted on purpose in order to anonymize the data. Otherwise by having the exact number of the crime occurrence it would be possible the victim of the crime to be identified indirectly, especially in the case that the crime occurred in private residences. Then the data would be categorized personal data within the meaning of article 4 of Regulation(EU) 2016/679 [66].

3.2.2 Exploratory Data Analysis

The Attica Crime data consists of 10.172.735 instances with 10 features. But, as can be seen in Figure 14, a large percentage of the instances have empty values in the features "*time_to*" and "*street*". More specifically the field "*time_to*" is empty in 3.935.185 instances, which is approximately the 39% of the total crime instances. This is not a problem for the data since, as mentioned above, the "*time_to*" field is entered only when there is no exact time of the crime or it is not known. On the other hand, as it turns out, there are 1,370,030 instances where the feature "*street*" is empty. This constitutes the 13% of the total data. Thus, since the dataset does not include the numbers of the streets, the non-existence of the street itself introduces a lot of noise into the data, and this is because the municipal units "*dimot_enot*", which is the next field of spatial determination, constitute very large areas in order to make predictions based on of them and draw conclusions.

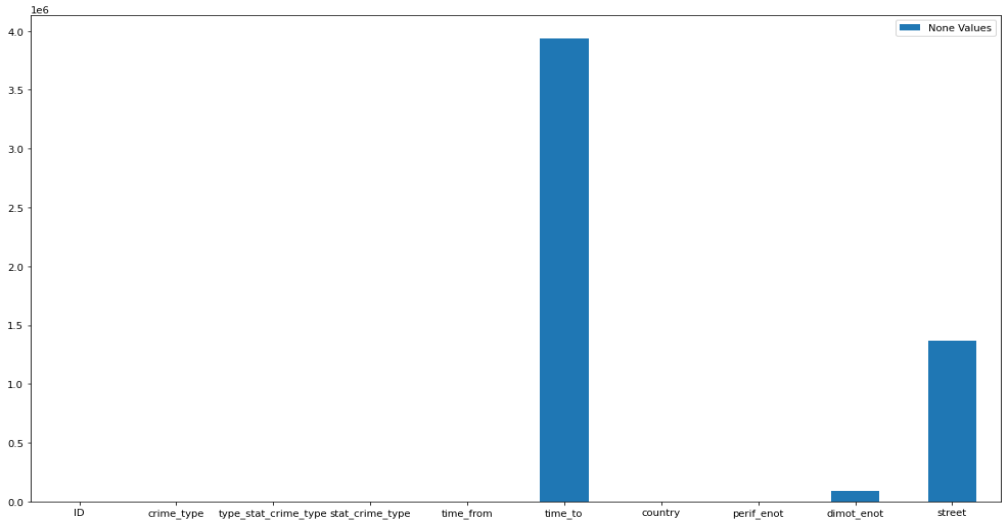

Figure 14: Hellenic Police Attica Crime Data – Sum of Empty Values

Regarding the distribution of the types of crimes against property, as shown in Figure 15, the majority of crimes concern theft, followed by vehicle theft. Robberies also represent a significant percentage of committed crimes. As it can be seen, the distribution of types of property crimes is similar to that seen in the New York dataset.

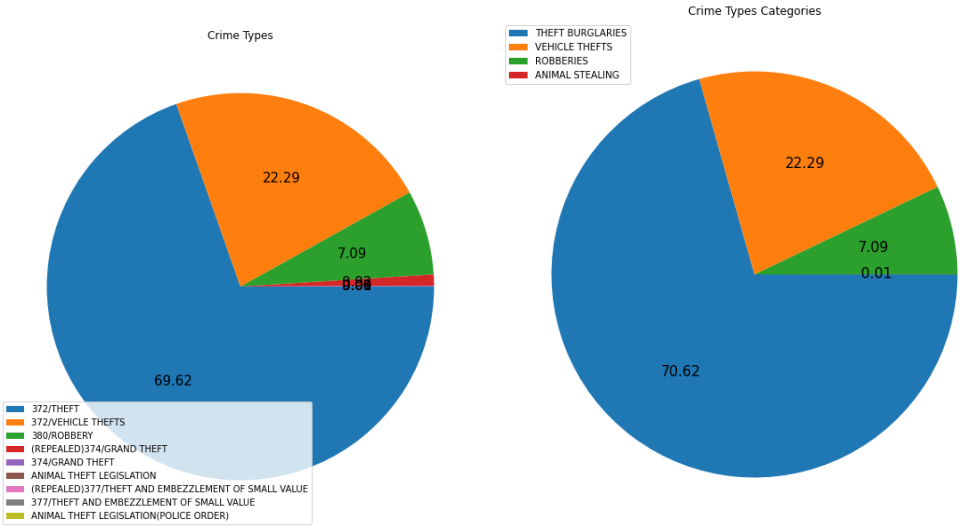

Figure 15: Hellenic Police Attica Crime Data – Crime Types Percentage

Regarding the spatial characteristics of the data as shown in Figure 16 there are 4 unique regional units, 379 unique municipal units and 9,363 unique street names. These numbers of unique values are significantly smaller than the 10,172,735 instances of the dataset. In addition, if these attributes are combined for each instance into comma separated alphanumerics, it turns out

that the dataset contains 31,398 unique addresses result. Therefore, the dataset of 10,172,735 instances concerns only 31,398 unique locations.

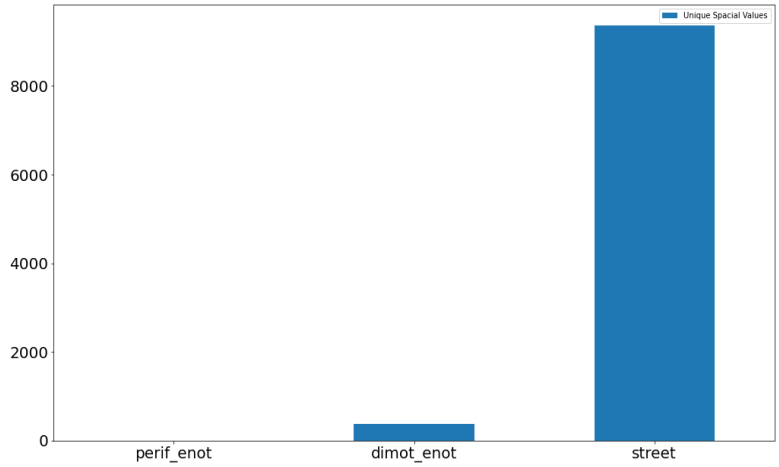

Figure 16: Hellenic Police Attica Crime Data –spatial unique values

Regarding the temporal characteristics of the offenses in the specific dataset, as can be seen in Figure 17, where the total number of crimes per day is presented, a periodicity also appears in this case. In this case, if we compare the results with those of the city of New York shown in Figure 12, they show greater variations in the total number of offenses. It is also noteworthy that after the beginning of the year 2020 there is a rapid drop in property crimes, which is even greater in the year 2021.

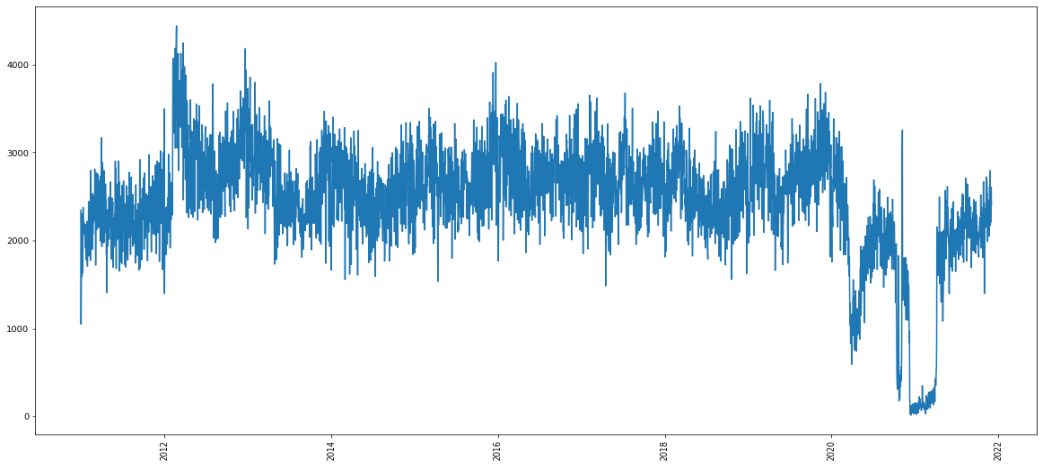

Figure 17: Attica Total Property Crimes per Day

3.2.3 Transformations –Other Datasets Uses

The locations of the crimes in the New York City Complaint Dataset, as it is presented before, are geocoded. Geocoding is the process of converting addresses (like a street address) into

geographic coordinates and it helps in transforming the data into forms suitable for training ML and especially DL algorithms in spatiotemporal prediction tasks, like the ones described in the subchapter 1.2. In addition, by using geocoded spatial data, it is easier to remove outliers, because it is possible to limit the data to those that are geographically involved in an area as it was done in the case of the New York City Complain Data. Whereas when the algorithms predictions are produced in a geospatial format, it is easier to visually check the effectiveness simply by plotting the distribution of the results and comparing them with the real historic data. Therefore, for all these reasons, but also in order to have a common structure in the training data of the algorithms, so that a fair comparison can be made, the dataset provided by the Hellenic Police was geocoded.

Geocoding can be done automatically by using http protocol to call Geocoding APIs. These APIs when called, with the address of a place as a parameter, they return the coordinates of that place. Popular open source platforms that provide such Web Services are Google Maps Platform [67], Nominatim platform [68] which uses open street map data and others. Unfortunately, these platforms on the free version have a restrictions either on the number of request that can be done per day, either on the time required on the response for each request. For example Nominatim API has 1 second minimum response time limit per request of a user. The process of calling the API for all instances of the Attica crime dataset will need 10,172,735 seconds (118 days). Consequently, in this case different tactics were followed in order to reduce the required requests and the total time of those required to geocode the data. As mentioned in the previous subchapter, it turns out that the dataset includes only 31,398 unique addresses. Therefore, these concatenated unique addresses were isolated in a new dataset. Then through the Geopy library [69], the APIs of Nomatim and Google Maps Platform were used in order to geocode these unique addresses. Moreover in order to speed up the geocoding process multiple workers working in parallel in different threads were created. Finally, the geocoded data were merged with the original dataset by taking into account the concatenated addresses. While only the addresses for which coordinates were found were used. This way the merged dataset includes 10,084,530 instances.

In order to remove incorrectly geocoded data, a dataset that includes the geospatial data of the Regions of Greece [70] was used, as was also done in the case of New York. The dataset that was used is publicly available on the open geodata repository of Greece and it contains the geographic boundaries of the 13 Regions of Greece, in “Geometric Polygon” structures. From these regions, only the region of Attica was used, the borders of which are shown in Figure 18 on the left, and then the boundaries of the islands included in this region were also removed in order to limit the crimes to those committed only on land according to the case of New York City. This is how the geographical boundaries of the Attica region emerged as shown in Figure 18 on the right. Then the boundaries mentioned above were used in order to remove any coordinates outside of them.

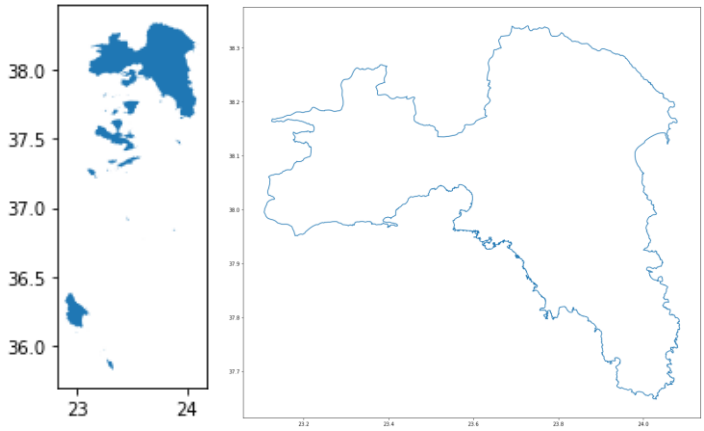

Figure 18: Attica Region boundaries from the Greece Regions Dataset

After the data has finally been geospatially limited to the land part of Attica, it is necessary, in order to have similarity with the dataset of the city of New York, to make a geospatial separation of the instances of crimes in areas. Of course, in the case of the data provided by the Hellenic Police, this kind of separation exists at least with labels, since the geocoding of the data was based on this. However, in order to have absolute correspondence in the information available between the two datasets, the separation needs to be done geospatially based on the coordinates. Thus, since there will be information on the boundaries of the areas, on the one hand, errors will be limited, on the other hand, it will be possible then extracting further features such as which areas are neighboring each other.

Additionally, for the same reasons, a dataset of the boundaries of the neighborhoods of Athens [71] was used which was integrated with the data of Attica so that it is possible to divide the municipality of Athens into neighborhoods corresponding to the process of the division of the boroughs of New York. For example, the municipality of Athens occupies an area of 38.96 km² and the neighborhoods into which it can be divided are 143. Manhattan occupies an area of 59.1 km² and is divided into 35 neighborhoods. Thus they can then be used as input to the algorithmic divisions of smaller regions into subregions in both datasets.

The merged boundaries of the resulting regions are those shown in Figure 19. Specifically, in purple color are shown the boundaries of the Attica region, in pink color are shown the boundaries of the municipalities and in green color are the boundaries of the neighborhoods of the municipality of Athens.

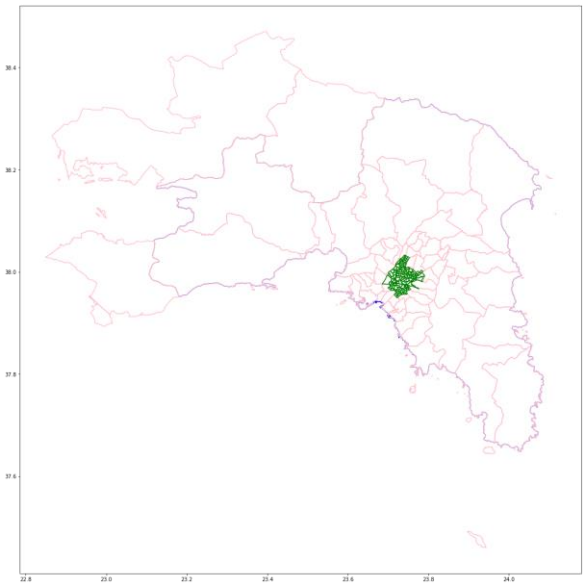

Figure 19: Joined Attica region – municipalities and neighborhoods boundary datasets

The region of Attica in this dataset is divided into 4 prefectures. The prefecture of Athens, Piraeus, West Attica and East Attica. The size of these prefectures is shown in Table 4. As it can be seen the prefectures are significantly larger than the New York boroughs showed in Table 2. Therefore, smaller areas need to be created in order to have comparable results. One of these smaller areas can be the Athens Municipality the neighborhoods of which was joined in the dataset as it was mentioned before.

Table 4: Attica Regional Units coverage

Prefecture	**Area(m^2)**
Athens	361
West Attica	1004
East Attica	1513
Piraeus	929

Regarding the beginning and end dates of the crimes due to their concepts and specifically due to the cases where the exact time of the crime was not known, a new feature was created which

is considered to be the actual time of the crime and is calculated based on the average time of the two moments in time. The algorithm for calculating said features is as follows.

Algorithm 1: Algorithm for computing

```
Input: time_from array, time_to array
Output: DATE_TIME array
for each time_from, time_to  do
        if time_to is null
                DATE_TIME = time_from
        else
                DATE_TIME = (time_to + time_from) /2
```

Finally, in order to give a picture of the distribution of crimes in areas, Figure 20 shows the crimes with red color that took place in the areas of the dataset based on the dates as they were previously formed for one year.

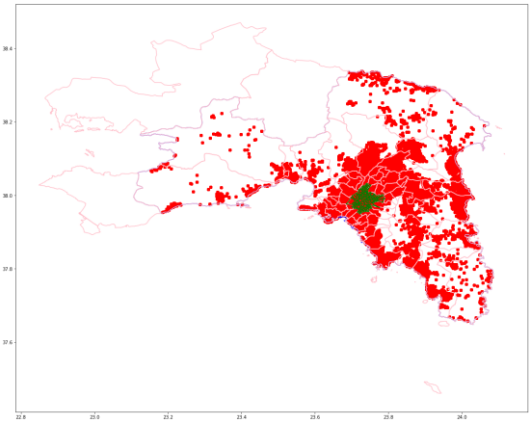

Figure 20: One-year Attica property crimes example

4 CHAPTER 3: Methodology – Description of Used Models

Spatio-temporal prediction tasks are a difficult field of application of artificial intelligence techniques, since it is required to create algorithms that on the one hand take into account many different types of characteristics of the given data, and on the other hand, to extract spatial and temporal patterns simultaneously. To address this complexity that characterizes the field in question, this study tested various DL algorithm architectures, each with different characteristics and capabilities. However, the diversity of these algorithms presupposes different forms of input and output data. Thus, for each of them, different techniques were used to transform the data of the two aforementioned datasets into suitable formats in order to feed them and compare their results. Also, in the present study, in each algorithm different formats of the data were tested in order to find out which of them gives better results.

From the above it follows that the selection of the appropriate formats of the data, the exact architecture of the DL algorithms as well as the selection of the appropriate high parameters of the algorithms is a difficult process as it includes a large number of parameters combinations to be examined. Therefore, in order to deal with this complexity, the Weights & Biases platform was used [72]. Weights & Biases is a ML platform which among other it provides methods to log parameters and results and visualize them in order to compare them and draw conclusions. To save time and computing resources, the Weights & Biases platform was used, in the case of the crimes that took place in the Manhattan area to find the optimal combination of parameters, to create the best models as well as to find the optimal partition of the regions. Subsequently, these models were tested for effectiveness in the remaining areas of the city of New York as well as Attica. In the continuation of this chapter, the DL algorithms that were applied with their different variations are presented, as well as the algorithms based on which the data transformations were made for each DL model produced.

4.1 Long Short-Term Memory

To begin with the first algorithm that was implemented in the current study is the LSTM. As it was mentioned before LSTM is a specific case of RNN suitable for use in sequential data predictions like time series. Consequently, an LSTM type model was the first model that was used in the current study for the prediction of the crimes.

4.1.1 Input Data Preparation

In the case of the LSTM model 2 different approaches were followed concerning the structure of the input and the output data. In the first approach the datasets were grouped by time periods and by neighborhoods. Specifically, different time intervals such as hour, quarter hour, eight hour and day were tested and the sum of crimes per neighborhood in these time intervals were calculated. For example, in Figure 21 we can see the crimes against property that occurred

in each New York neighborhood, in one hour range colored according to the magnitude of the sum of the crimes in the specific neighborhood.

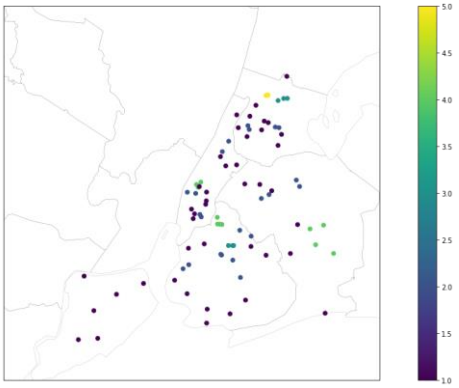

Figure 21: One-hour Neighborhoods crimes heat-map

Then the sums were scaled to the range [0-1] using the MinMaxScaler of the sklearn library [73], and finally a pivot table was created. Each instance of the final pivot table contained the sums for every neighborhood for the selected time interval.

In the second approach, the areas were divided into k blocks of equal areas following the Algorithm 2 presented below.

Algorithm 2: Create Area Grid Algorithm

```
Input: geographic_coordinates_array, n
Output: id_array
max_Longitude = get max Longitude from the array
min_Longitude =  get min Longitude from the array
max_Latitude   =   get max Latitude from the array
min_Latitude   =   get min Latitude from the array
cols = get n evenly spaced numbers over (min_Latitude, max_Latitude) interval
rows = get n evenly spaced numbers over (min_Longitude, max_Longitude) interval
for each col, row in  cols, rows do
        id = calculate_id (col, row, cols_number)
        add id to id_array
return id_array
```

Due to this algorithm, in order to do split an area, the maximum and minimum values of longitude and latitude respectively are found from the data. Then the intervals connecting the minimum and maximum values of the coordinates are calculated and divided into equal subintervals. The n+1 points separating the coordinate intervals are stored and sorted. In this way, $k = n*n$ divisions of the area into blocks are created. In order to find which block each instance of

the data belongs to, each instance is assigned a "col" number that represents the sequence number of the interval where the instance's latitude is located and a "row" number that represents the sequence number of the interval that is the longitude of the instance. So, it can be considered that each instance belongs to a cell of a table with coordinates (row, col). Finally, in order to identify each block, an “id” was calculated based on the following equation (8).

$$id = \left(row \times (cols_{number} - 1)\right) + col \quad (8)$$

Therefore, each instance was assigned an id which represents the serial number of the cell of the table it contains, starting a numbering from the position (0,0) to the position (n,n).

Having now all instances classified into blocks ids the same procedure followed in the neighborhoods grouping case is followed. So, in this case different time intervals such as hour, quarter hour, eight hour and day were tested and the sum of crimes per block in these time intervals were calculated.

4.1.2 Model Architecture

One case of the proposed LSTM architecture can be seen in Figure 22. In that case the input is instances of crimes against property in the Manhattan borough grouped by 8 hour intervals and by 36 neighborhoods of the Manhattan. So in this case the input consists of 63 8hour timesteps. The output is the prediction of the crimes that will take place the next 8 hours in each of the 36 neighborhoods of the Manhattan. Therefore, it is understood that the model receives as input multiple time series (36 in this particular case) and tries to simultaneously predict the values of these time series in the next unit of time.

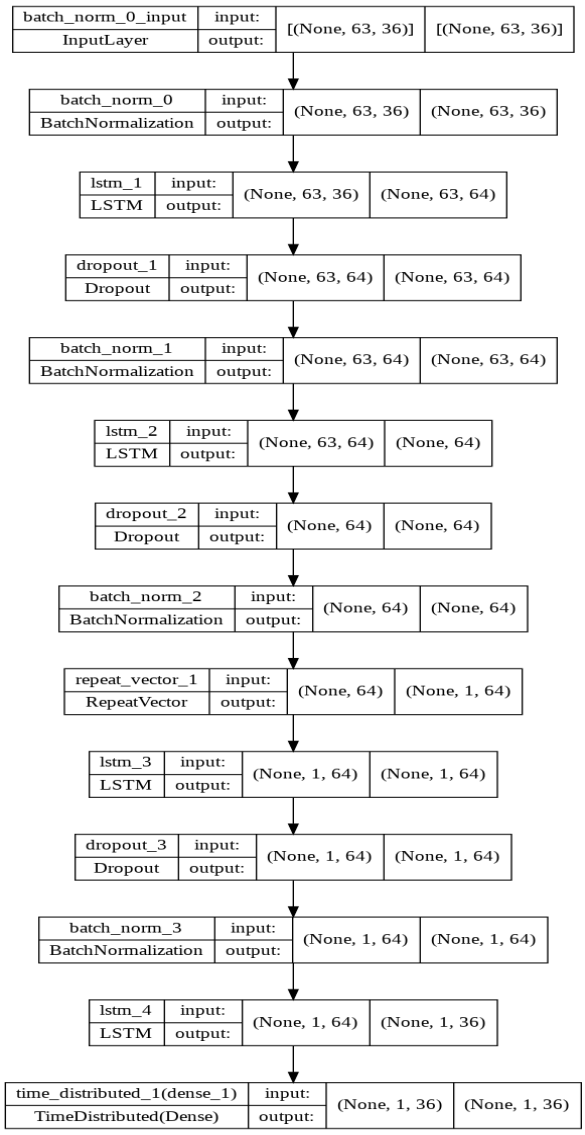

Figure 22: LSTM Model Architecture

As can be seen in the figure, internally the model consists of 4 LSTM stacked layers of 64 units each. While between the LSTM Layers there are BatchNormalization and dropout layers. BatchNormalization layers normalize the data on each layer, keeping the mean close to 0 and standard deviation close to 1, that way in deep neural networks the epochs needed for training is reduced [74], [75]. Dropout layers set some portion of the inputs of the layer randomly to 0 at each step on the training process. In this way randomness is introduced into the training of the algorithm and thus overfitting can be avoided [76], [77]. In this case dropout rate is set to 0.1. In the model a repeat vector layer is inserted in case that multiple steps prediction is used. Then the given input will be cloned as many times as the steps that need to be predicted, in order to create the right shape of the output for the next layers. Finally a Dense Time Distributed Layer, with linear activation function, is implemented, in order to get the final output. The time distribution is

implemented in case that we want to predict multiple steps, so that the dense layer will be applied in every temporal slice. For this model the mean square error loss function to compute the errors between the real and the predicted values. While the keras implementation of the RMSprop [87] algorithm, with 0.001 learning rate was used as an optimizer to this model [78]. The RMSprop optimizer implements the Gradient Descent algorithm with parameterized momentum in order to get the global minimum of the loss.

4.2 CLSTM

The LSTM algorithm as mentioned earlier is only able to capture the temporal characteristics of the crime data and therefore treats the crime data of the given areas as if they were different independent time series. In order to address this issue, it is necessary to use more complex DL algorithms that will be able to capture the spatial characteristics of the data. Therefore, a more complex algorithm CLSTM (see in subchapter 2.2) was implemented. The CLSTM is an algorithm that combines the characteristics of both CNN and LSTM algorithms [55]. As a result, spatial and temporal features can be captured simultaneously. In the continuation of the chapter, the structure of the data and how it was created in order to be suitable input for the CLSTM model as well as the architecture of the model are presented.

4.2.1 Input Data Preparation

In this application the input data should be structured as sequential images in order to be fed in CLSTM model, and convolutional operations to be applied to them. So the datasets for this case were treated as serial spatial representations of the areas which change per unit of time. To achieve this, Algorithm 2 was first applied as mentioned in the previous sub-chapter to divide the regions into blocks. But this algorithm does not produce ids and data for a block which did not show any criminal activity in the entire dataset. Therefore, in order to fill in the empty blocks in the "images", after grouping the data per time unit and produced ids, a grouping was made of the produced ids and the missing ones from 0 to n^2 were filled in with zero values for all data. Then each instance of the data was normalized to [0-1] range and reshaped to the shape (n,n). For example, Figure 23 illustrates the result of the above-mentioned procedure for the sub-dataset of the Manhattan borough crimes.

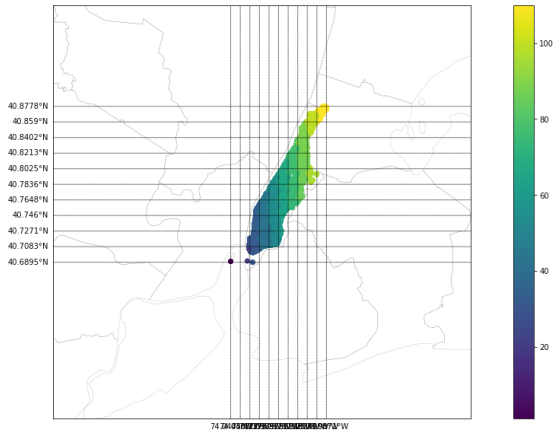

Figure 23: Manhattan Grid Map

As can be seen from the figure, in each area there are on the one hand blocks inside the area that do not have criminal activity, and on the other hand, blocks outside the area that are set to have zero activity because they do not belong to the area. Nevertheless, in order to be able to feed the data to the CLSTM model, the data instances should have a square shape just like when it comes to an image. This creates sparsity in the input data. Therefore, afterwards the dataset consists of instances of (*n*,*n*) shape, which in each cell contain a number of range [0-1], which represents the percent of the criminal activity of the corresponding block for a specific time range.

In addition, a binary dataset was created by replacing the data values that are greater than 0 with 1. Thus, the new dataset has in a cell the value 0 if no crime was committed in the corresponding block during the specific time period and the value 1 if even one was committed crime in the given time period. In this way, the prediction of crimes in an area can be tested as a binary classification task of the blocks of the area.

Finally, the data split into train set, validation set and test set. The validation set consists of two-year crime data, test set consists of one-year crime data and the rest years constitute the train sets. It should be noted that in this way no data leakage between the sets is occurred.

4.2.2 Model Architecture

The CLSTM model applied as it can be seen in Figure 24 mainly consists of 4 stacked ConvLSTM2D layers. For the current model, the keras implementation of the CLSTM algorithm was used [79]. A ConvLSTM2D layer uses memory cells like the case of LSTM neural network to keep the state between long-range dependencies (described in the subchapter 2.1.1), but the main difference here is that the inputs, the hidden states, all the gates and all the transformations are made on 2D tensors, with the use of convolutional operations [55]. Therefore, in each layer the parameters of a CNN and a LSTM module have to be set.

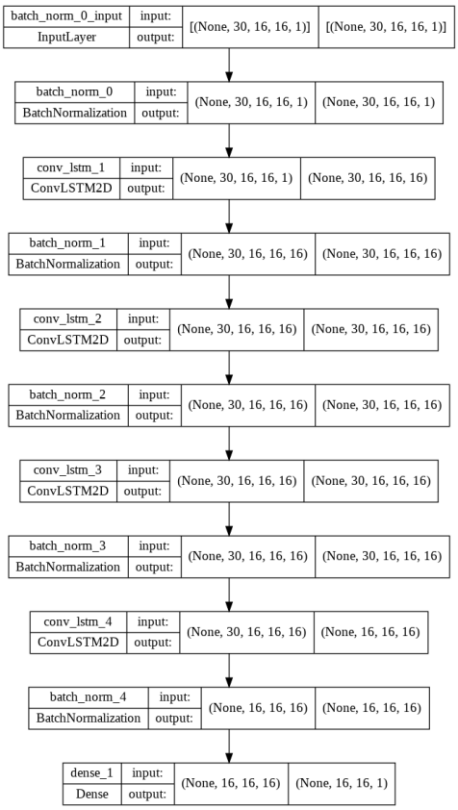

Figure 24: ConvLSTM Model Architecture

Specifically, in Figure 24 we can see a model built to be fed with 30 timesteps of (16, 16) shaped area grids in order to predict the next timestep area grid. In particular, the Convolutional window size (kernel) dimensions was set in shapes (8, 8), (4, 4), (3, 3), and (2, 2) in order to have convolution operations that will extract spatial correlations between sub-regions with different size, of the given areas, as the model goes deeper and deeper. Moreover, the number of the output filters in the convolution were set in sizes 16. These sizes are quite small when it comes to CNNs but in this case the CLSTM layers are quite complicated and increasing their size would increase the model size and the time required for predictions drastically.

BatchNormalization layers are inserted between the CLSTM layers to normalize the data inside the model, while the final output of the model is given by a dense layer. Regarding the optimizer in this particular architecture, the keras implementations of both the RMSprop optimizer and the Adam optimizer were tested. Adam optimizer also implements the Gradient Descent algorithm but except from using only momentum it also uses adaptive learning rate to update the parameters [80], [81].

In the case of prediction using the CLSTM model, two different approaches to the crime prediction problem were tested as mentioned above. The first is the prediction of the amount of

crime per block while the second is the prediction of the presence or absence of crime in a block. In the first approach, the last dense layer of the CLSTM model has a ReLU activation function in order to produce the final output because numbers greater than 0 need to be produced. While in the second approach, the Dense layer of the model has a sigmoid activation function in order to produce numbers in range [0-1] as a probability.

Regarding the loss functions in the first case the Mean Squared Logarithmic Error (MSLE) loss function was used which can be interpreted as a measure of the ratio between the true and predicted values. This loss function treats small differences between small true and predicted values approximately the same as large differences between large true and predicted values. On the other hand, in the second case, the keras implementation of the Binary Crossentropy function was used.

4.3 3DCNN

The next model tested in the present study for crime prediction is that of 3D Convolutional Neural Network(3DCNN). In fact, a 3DCNN component does not differ from a normal CNN except for the fact that the Convolutional operations as mentioned before are done on a 3-dimensional level as shown in the following Figure 25.

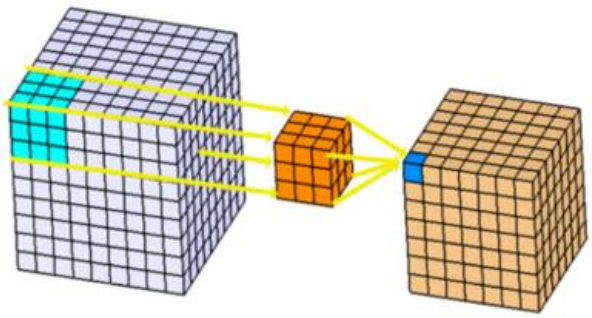

Figure 25: 3D Convolution Operation [95]

Therefore, here we use the time dimension of the data as the third dimension of the input data of the 3DCNN model. As a consequence, the structure of the input data of this model is the same with the one described before for the CLSTM model. So the same algorithms were used in this case to prepare the data.

4.3.1 Model Architecture

Regarding the 3DCNN model as it can be seen in Figure 26 an encoder-decoder model structure was tested. In an encoder-decoser architecture is attempted to create a DL model that can

learn to transform a given input in a higher representation in the encoding phase, to decode that representation to the right output in the decoding phase.

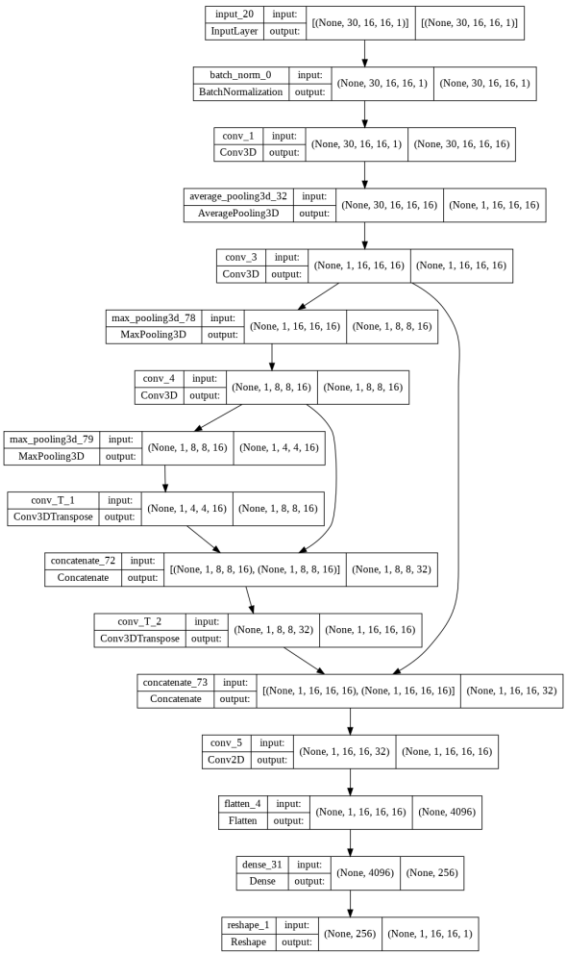

Figure 26: 3DCNN Model Architecture

Therefore, in this particular model initially in the encoding phase 3D pooling operations like 3DAveragePooling and 3DMaxPooling are used in order to downsample the input data and get the higher representation of them. 3DAveragePooling and 3DMaxPooling are the same pooling operation that are used in the regular CNNs but instead of operating on 2-dimentional data they are 3Dfilters for data with 3 dimensions. In particular, 3DAveragePooling layer is used in this case to reduce the temporal dimension of the input, while the 3DMaxPooling layers are used to reduce the spatial dimensions of the input. By using the 3DAveragePooling layer to reduce the temporal dimension the next layer gets as input for each block the average value of the crimes occurred in the input time. So, that way the model can capture each blocks trend of the crime for that time range. On the other hand, 3DMaxPooling layer is capturing the areas with a higher concentration of crimes than their surroundings. Then in the decoding phase Conv3DTranspose layers are used in order to upsample the input. A Transposed Convolution operation works the opposite way of the normal Convolution operation. Instead of reducing the size of the input data they increase it

[82]. The convolutional parameters are set in the model in a way that on the decoding phase the output will be formed in the wanted shape that needs to be predicted.

Unfortunately, the procedures of downsampling – upsampling using pooling operations can cause the loss of information that is necessary for the model in order to make the right predictions. To address this problem in this model, as it is shown in the Figure 26, residual connections were used. Residual connections, as it was described in the subchapter 2.2.2, can transfer data from one layer to subsequent layers bypassing a plurality of layers. Then these data are concatenated with the output of the layer in that stage and the concatenated output is given as an input in the next layer. That way "information loss" can be avoided. Also, a BatchNormalization layer has been added to this model, while the same parameters applied to the CLSTM model were applied to the activation functions, the loss function and the optimizers of this model.

4.4 GCLSTM

The last approach applied to the task under study was the representation of spatial entities and their correlations in graphs. This way it becomes possible to use Graph Neural Networks for the spatio-temporal prediction of criminal activity, as their use has also been described in the literature. In particular, a Graph Convolutional Long Short-Term Memory (GCLSTM) model was implemented that uses Graph Convolution modules to capture the spatial correlations and LSTM modules to capture the temporal correlations of the given data. Therefore, the methods and algorithms applied to represent the data in graphs are first described, while the architecture of the model in question is then analyzed.

4.4.1 Input Data Preparation

As it is mentioned above the spatial correlations of the data need to be represented into graph in order to be fed to the model. Thus, corresponding to the methods followed in the CLSTM case there were also here two different approaches to the spatial data.

According to the first approach, the nodes of the graph are the neighborhoods of a region and the connections at the nodes are made between neighboring regions. In order to implement this, a dataset was created in which each neighborhood is matched with all the neighborhoods that have at least one common point on their boundaries. For this procedure the geometric boundaries from the Geocoded datasets that was presented before were used. An example of the produced dataset, for 5 Manhattan neighborhoods, is shown in Table 5.

Table 5: Manhattan neighborhoods neighboring subset

neighborhood	**neighbours**
Battery Park City	Financial District, Tribeca
Central Park	East Harlem, Harlem, Midtown, Upper East Side, Upper West Side

Chelsea	Flatiron District, Gramercy, Greenwich Village, Hell's Kitchen,Midtown, West Village
Chinatown	Civic Center, Little Italy, Lower East Side, SoHo, Tribeca, Two Bridges
Civic Center	Chinatown, Financial District, Tribeca, Two Bridges

Having the above mentioned dataset in order to pass the graph structure information in the Graph Convolution layer of the model the only thing needed to be done was to create an immutable adjacency matrix. The following Algorithm 3 was used to produce this matrix.

Algorithm 3: Neighborhood Adjacency Matrix Creation Algorithm

Input: neighboring_dataset, empty
Output: adjacency_matrix
adjacency_matrix = create a (neighborhood, neighborhood) shaped zero filled crosstab
for each **neighborhood** in **neighboring_dataset** do
    neighbours_array = get_neighbours of the **neighborhood**
    for each **neighbour** in **neighbours_array** do
        adjacency_matrix[**neighborhood**, **neighbour**] = 1
return adjacency_matrix

Therefore, the returned matrix has values equal to 1 in the cells where the two areas are neighborhoods. An example of the neighborhood representation graph produced by the corresponding Manhattan adjacency table is shown in Figure 27.

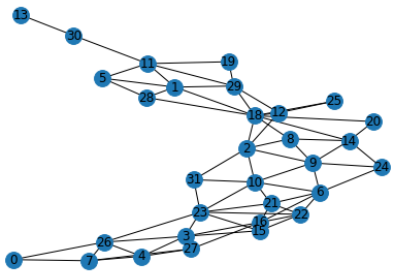

Figure 27: Manhattan Neighborhoods Graph

On the other hand, according to the second approach, the area is divided in a grid with blocks as it was done in the previous models. Therefore, the nodes of the graph in this case are the blocks in which the areas were divided and the connections between those nodes are made based on the adjacency of them. In order to achieve the connections between the neighboring blocks a procedure of producing ids with the use of equation (8) (described in subchapter 3.1.1.) was followed, while

the column, row values of each produced id were kept to be used in the next step of the procedure. In particular, firstly the Algorithm 2 was used in order to create the area grid and annotate the data with block ids. In this case blocks of the grid which showed none crime occurrences should not be present in the data show there is no need to fill the missing ids with zero values. Then, by using the column, row values for each block id in the dataset, all the blocks of a matrix that are one step away from the block and they exist already in the set of the blocks with crimes, were marked as neighbors by updating the corresponding cell of the adjacency matrix. This is how the desired adjacency matrix is finally obtained, which will be fed to the model to understand the relationships between the nodes. Figure 28 for example shows a graph produced by dividing the Manhattan into (16,16) shaped grid.

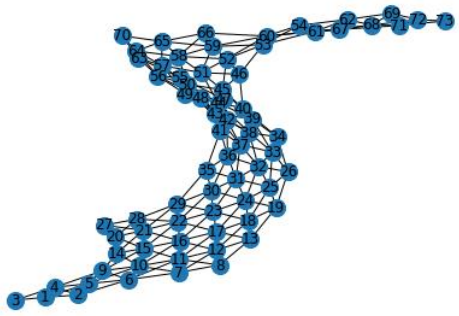

Figure 28: Manhattan Graph by Grid Map

4.4.2 Model Architecture

Regarding the architecture of the GCLSTM model now, it is more complex than the previous models tested as it can be seen in Figure 29. The proposed model here was inspired by the keras example of implementation in the Traffic forecasting domain [56], [83]. In an abstract level the used GCLSTM model mainly consists of the 4 stacked Graph Convolution modules and 4 stacked LSTM modules.

The input data firstly are fed to the Graph convolution modules in order to get the spatial correlations between the data. Then the output is reshaped in a proper form for the LSTM modules, in order to extract the temporal correlations of the data.

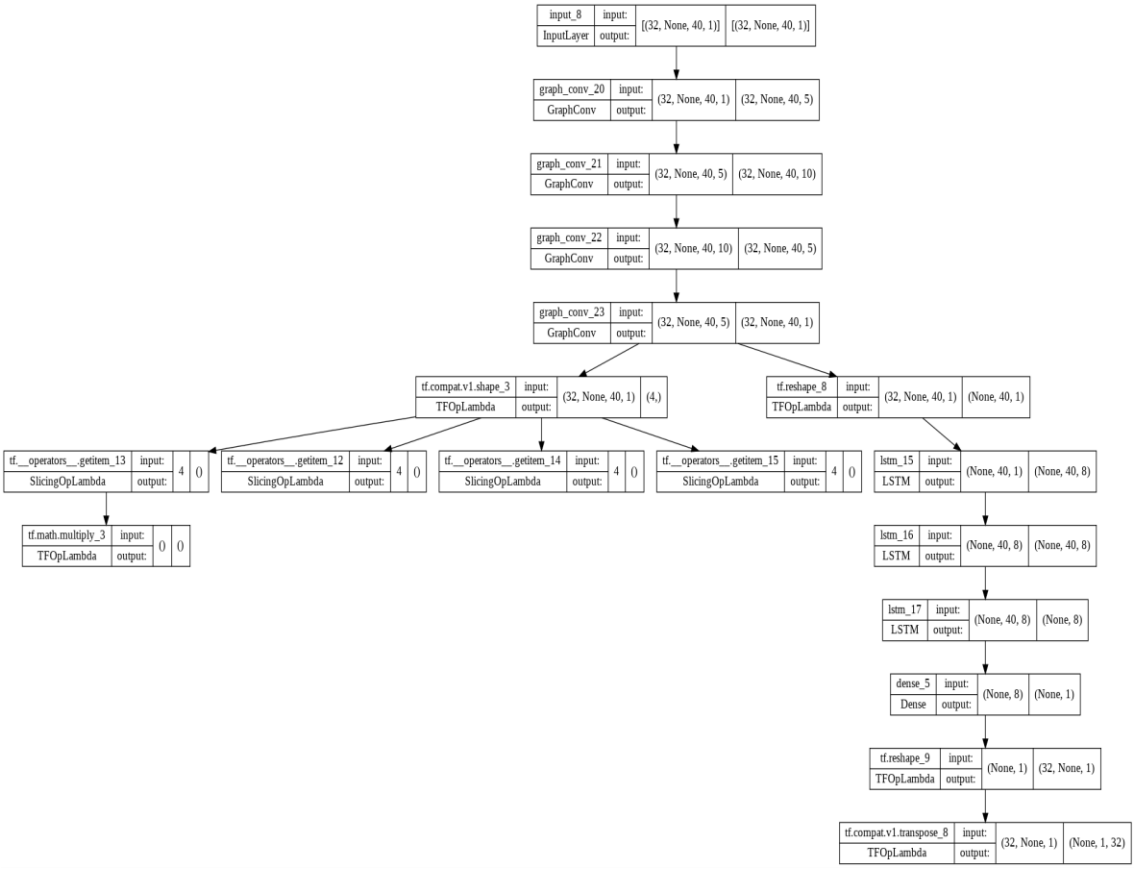

Figure 29: GCLSTM model architecture

Specifically, the model shown in **Figure 29** is fed with the Manhattan data consisted of 32 regions. Also, the adjacency matrix of the areas in question with shape (32,32) is given as input to the model. Thus, in each Graph convolution layer, the messages between the nodes are aggregated based on the adjacency matrix and the new weights are calculated each time of the node by combining the results (as described in chapter 1.1.3). In this model, the mean value calculation function and the concatenation of the results have been selected as aggregation, combination functions respectively. Furthermore, regarding the size of the weight vectors, sizes 10 and 5 were chosen, while the ReLU activation function was chosen to return the graph modules outputs. Then regarding the LSTM layers, each layer consists of 8 LSTM units. The final output of the model is given by a dense layer with a ReLU activation function.

5 CHAPTER 4: Results

As mentioned above, the parameterizations of the different models were chosen based on the results they showed in the case of Manhattan crimes. For the training and evaluation of the models, the corresponding data were selected in each case in order to be able to compare the effectiveness of the different algorithms. In particular, the crime data for the Attica region only concern crimes against property. Therefore, in order to be able to compare the results and draw valid conclusions, only instances conserning property crimes were selected from the New York City dataset. Additionally in cases where the data were reformed in spatial grids, the number of the blocks of the grids were chosen so that the block size is roughly the same as the block size in the Manhattan case. While in the case of the Attica dataset from each region were selected areas that have the approximately the same size as the Manhattan area in order to limit down the sizes of the models. Specifically, **Table 6** shows the number of parts into which the coordinates of the area were divided according to their size, in order to create the blocks. For example, models showed to perform better in the Manhattan area when it was divided into (16 x 16) blocks grids. In this case the Manhattan blocks cover an area of approximately $350m^2$. So, then all the areas were divided so that the blocks cover an area of about 350 square meters.

Table 6: Areas Grid Size

Area-Name	Coordinates Divisions Number	Area-Coverage
Manhattan	16	59.1
Bronx	22	110
Queens	34	280
Brooklyn	28	180
Staten Island	26	152
Athens Municipality	13	38.1
Piraeus Prefecture	16	50.42
West Athens Area	16	65
East Athens Area	16	62

In order to evaluate and compare the models, corresponding metrics were selected in each case. Specifically, in the cases where the task is to predict the amount of crime activity that was occurred in an area (Continuous Values Tasks), three different metrics were chosen for the evaluation. These metrics are the Mean Square Error, the Mean Average Error and the Root Mean Square Error.

Therefore, Table 7 shows these three metrics for the LSTM model on the New York City and the Attica datasets.

Table 7: LSTM results

New York City			
	MSE	**MAE**	**RMSE**
Manhattan	0.0003	0.0093	0.0174
Bronx	0.0001	0.0052	0.0103
Queens	0.0005	0.0103	0.2341
Brooklyn	0.0024	0.2546	0.0493
Staten Island	0.0001	0.0027	0.0127
Attica			
Athens Prefecture	0.0007	0.0072	0.0280
West Athens Prefecture	0.0007	0.0082	0.0266
East Athens Prefecture	0.0032	0.0040	0.0179
Piraeus Prefecture	0.0023	0.0184	0.0480
Athens Ota	0.0002	0.0018	0.0158

The errors seem to be low but actually they don't represent the reality. Especially, in the Figure 30 which shows the comparison of the predictions with the actual crimes, this weakness of the model is easily discernible. The model is expectedly unsatisfactory since it cannot capture the spatial correlations between regions as it was mentioned before. The model only manages to capture only the trend of the areas producing common predictions and for this reason the errors range very low. However, in the present study the use of the vanilla LSTM algorithm was chosen in order to have a basic model with its results as a measure of comparison.

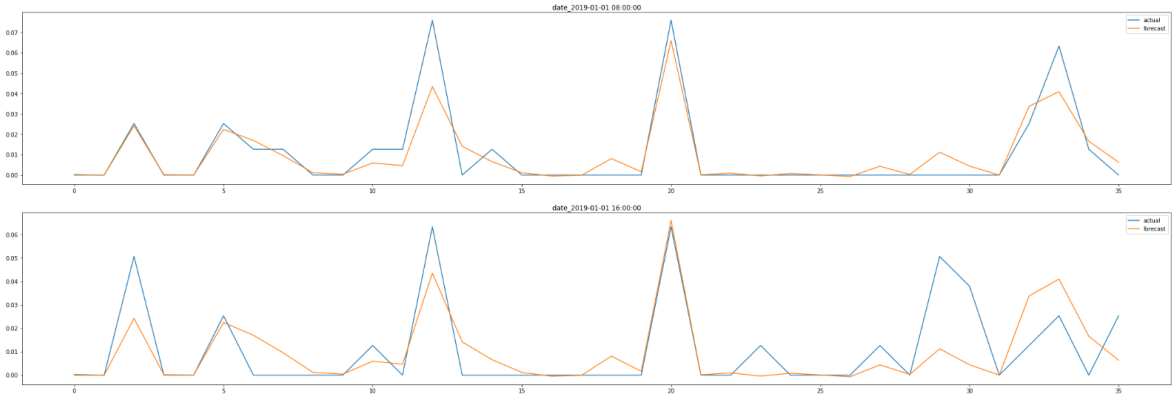

Figure 30: Manhattan Crime Predictions using LSTM model example

The next algorithm, which was measured for the accuracy of its predictions, is the CLSTM algorithm. The performance of the CLSTM model is shown by the metrics presented in Table 8. In the case where the models try to predict the occurrence of a crime into block (Binary Values Tasks) the Accuracy and the Roc Accuracy Scores where additional chosen, in order to evaluate the results. Additionally, due to the sparsity of the crime occurrences in the areas the aforementioned metrics were calculated in two different concepts. In the first concept, the metrics were calculated in the full set of blocks into which each area was divided. On the other hand, in a second concept the calculation of the metrics was done only for the blocks that showed criminal behavior. In this way, blocks that are not included in the selected area but must be present with zero values in order to form square shapes suitable for the model in question, can be extracted from the final result. Furthermore, in this way it is possible to compare the results with those of the model tested afterwards.

Table 8: ConvLSTM results

Error Metrics for blocks with Crimes					
New York City					
	Continuous Values Task			**Binary Values Task**	
	MSE	**MAE**	**RMSE**	**Accuracy_Score**	**Roc_Acc_Score**
Manhattan	0.6300	0.4800	0.8000	0.8528	0.8136
Bronx	0.1059	0.1349	0.3255	0.9676	0.8055
Queens	0.2726	0.2681	0.5221	0.9507	0.8383
Brooklyn	0.3105	0.3174	0.5572	0.9745	0.7585
Staten Island	0.0370	0.0490	0.1900	0.9963	0.5757
Attica					
Athens Prefecture	20.14	1.3	4.48	0.8855	0.8010
West Athens Prefecture	11.77	0.73	3.43	0.9768	0.6331
East Athens Prefecture	19.83	0.95	4.45	0.9635	0.6815
Piraeus Prefecture	6.8702	0.3992	2.62111	0.9694	0.6768
Error Metrics for All Blocks					
New York City					
	Continuous Values Task			**Binary Values Task**	
	MSE	**MAE**	**RMSE**	**Accuracy_Score**	**Roc_Acc_Score**
Manhattan	0.18	0.13	0.42	0.9632	0.9584
Bronx	0.0591	0.0754	0.24312	0.9847	0.9080
Queens	0.1405	0.1384	0.3749	0.9780	0.9253
Brooklyn	0.1831	0.1873	0.4279	0.9869	0.8712
Staten Island	0.017	0.023	0.1339	0.9990	0.7419
Attica					
Athens Prefecture	17.7	0.87	4.21	0.9381	0.8988
West Athens Prefecture	4.55	0.28	2.13	0.9867	0.7798
East Athens Prefecture	3.17	0.15	1.78	0.9880	0.8682
Piraeus Prefecture	2.6837	0.1562	1.6382	0.9805	0.8605

From the aforementioned table it can be seen firstly that the crime occurrence task, the results of which are shown on the right of the table (Accuracy and Roc Accuracy), is easier to be predicted and the corresponding model showed much better results. Also, comparatively, the models in both tasks seem to have much more accurate predictions in the data of New York compared to the data of the regions of Attica. A better sense of the performance of the models can be obtained from Figure 34-34, which show examples of the real crime occurrences and the predictions of the models. As can be seen, the models can extract from the data both their temporal and spatial correlations. Thus, even in the event that the exact occurrences of crimes are not predicted, a prediction can be made that is spatially close to reality.

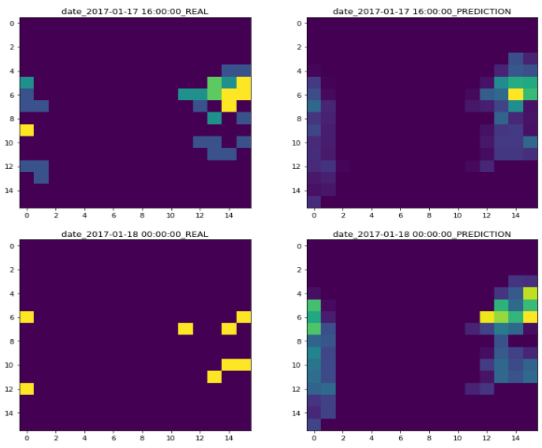

Figure 33: Continuous Values Task- Manhattan Real and Predicted data Example

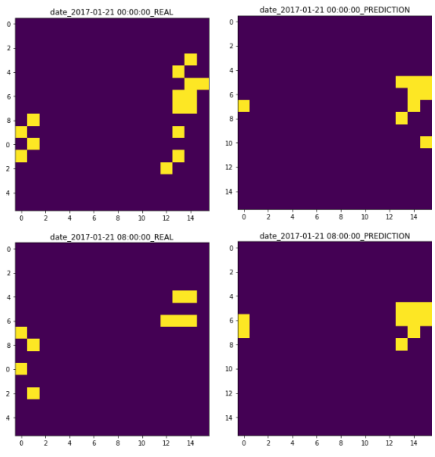

Figure 34: Binary Values Task- Manhattan Real and Predicted data Example

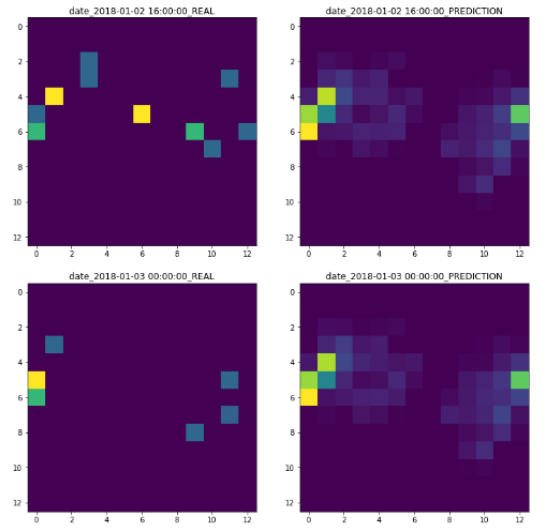

Figure 32: Continuous Values Task- West Attica Real and Predicted data Example

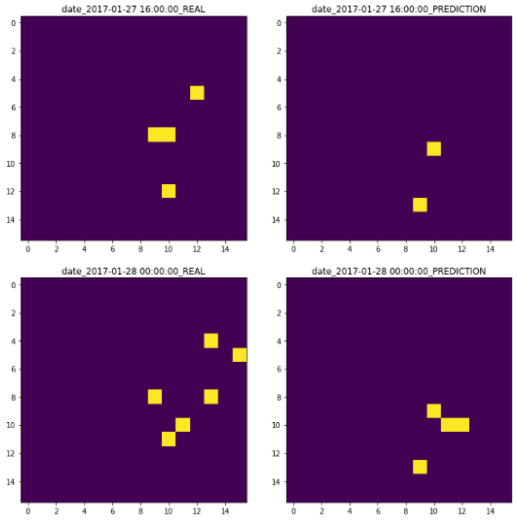

Figure 31: Binary Values Task- West Attica Real and Predicted data Example

Having the performance results of CLSTM models the next model tested was 3DCNN. Unfortunately, despite the use of residual connections in the architecture of the model, the information lost during the training of the algorithm is very large, with the result that the

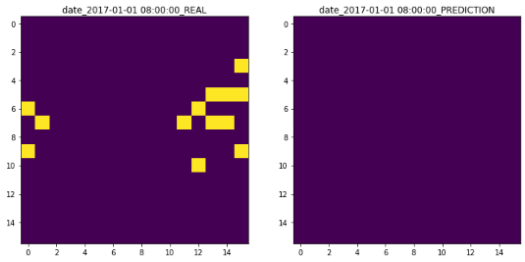

Figure 35: 3DCNN Manhattan Real and Predicted data Example

architecture in question cannot produce predictions. For example, in Figure 35 the inability of said model to produce predictions can be seen.

Finally, Table 9 shows the results of the metrics for the GCLSM model, which is the last model tested. In this model, as mentioned above, two different separations of the data were tested. In the first one, the graphs were created based on the neighborhoods and in the second one, the graphs were created by dividing the regions into blocks of equal size. Thus, the table shows the metrics for both separation methods.

Table 9: Graph Convolutional LSTM Results

Error Metrics for Neighbourhoods			
New York City			
	MSE	**MAE**	**RMSE**
Manhattan	0.9152	0.6807	0.9566
Bronx	1.1974	0.7326	1.0942
Queens	0.6454	0.5412	0.8027
Brooklyn	1.1741	0.7803	1.0835
Staten Island	0.0800	0.1379	0.2925
Attica			
Athens Prefecture	0.99	0.4215	0.9981
Error Metrics for Grid Blocks			
New York City			
	MSE	**MAE**	**RMSE**
Manhattan	0.8600	0.5300	0.9200
Bronx	0.8600	0.5049	0.9320
Queens	0.4757	0.4934	0.6897
Brooklyn	0.5488	0.5768	0.7408
Staten Island	0.0745	0.1072	0.2730
Attica			
Athens Prefecture	10.1400	0.9300	3.1800
West Athens Prefecture	4.7800	0.6100	2.1800
East Athens Prefecture	7.8100	1.0123	2.7952
Piraeus Prefecture	10.0853	0.9271	3.0500

As shown in the table, the GCLSM model, despite showing that it can understand the spatial and temporal correlations of the data to a certain extent, its predictions are insufficient and much worse than those of the CLSTM model. But especially in the case of the data of the Athens area, and specifically in the separation of the data into neighborhoods, the model seems to make more accurate predictions than in the case of the separation into blocks in the case of CLSTM model. This difference in performance may be due to the size of the blocks.

Summarizing from the algorithms tested it is clear that the algorithm showing the best results is CLSTM as it can be seen in Figure 36 and Figure 37. The remaining algorithms have little or no ability to perceive spatiotemporal correlations between data. While specifically for the crime data of Attica even the CLSTM model was able to simply perceive the areas of concentration of crimes.

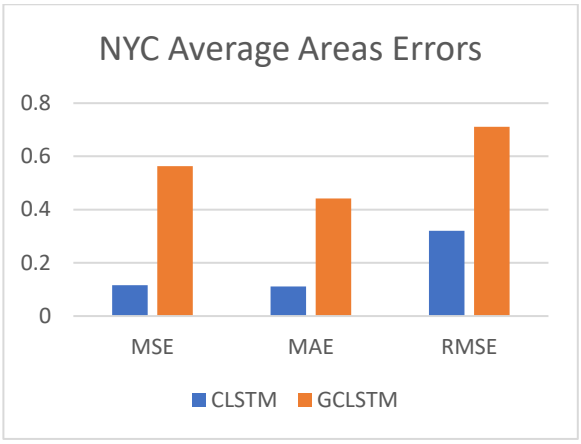

Figure 37: CLSTM-GLSTM New York City Areas Average Errors

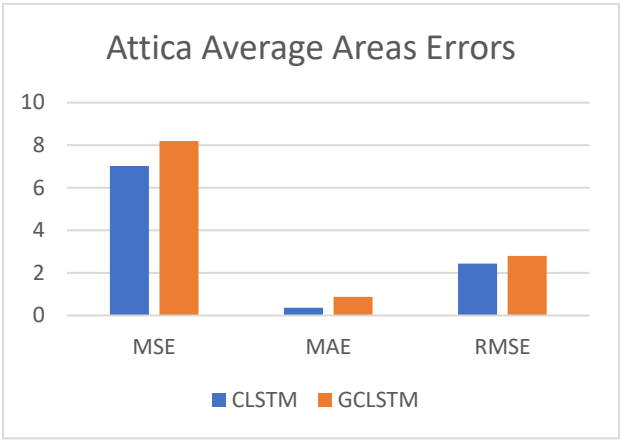

Figure 36: CLSTM-GLSTM Attica Areas Average Errors

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

In this chapter, the conclusions drawn from the results of the application of the above-mentioned DL models are firstly presented, while concerns about the application of artificial intelligence in the field of crime prediction are mentioned. Finally, different methods and approaches that can be applied in order to be able to create models with more accurate spatiotemporal crime prediction are mentioned.

Conclusions

In this study, different DL models were implemented in the crime prediction task in order to find which of them predicts more accurate the future criminal activity of an area based only on historical crime data. From these models, the Convolutional Long Short-Term Memory (CLSTM) model produced more accurate predictions in comparison with the rest models. While the Graph CLSTM model was able to capture in some level the spatio-temporal features of the data. Thus, it can be concluded that DL models must have a complex architecture consisting of individual modules, each of which will be able to capture an aspect of the data's characteristics (i.e., topology). For example, in the CLSTM model convolutional operations are able to capture the spatial correlations between the historic crime data instances, while LSTM operations are able to capture temporal correlations between them.

Regarding the data preprocessing, it is found that the structure of the data has a very large effect on the accuracy of the predictions. Specifically, as presented, the division of areas into grids consisting of (16 * 16) blocks and the use of 8 hours as a unit of time, in the case of New York City very good results were obtained. Conversely, in the case of Athens, the forecasts were not as accurate. Therefore, each areas spatial features like the street layout and the sparsity of the crime occurrences may have great impact in the accuracy of the predictions. The absence of address numbers in the Attica data definitely contributed to this fact. Specifically, during the geospatial coding, the absence of numbers moved all the places where the crimes were committed to the middle of the corresponding street. Therefore, especially on major roads, the displacement of the points was such that it removed the points from their real environment thus introducing into the data unacceptable levels of noise. In contrast, in the case of the New York crime data as well as in other crime datasets, the spatial data was anonymized at the block level, resulting in a low spatial bias.

Second Thoughts

The use of the DL by law enforcement agencies in a production environment, even though it may look promising, it has some major drawbacks. Firstly, by using such decision-making tools, it is not possible to determine how a prediction was derived from the model, and therefore on which characteristics the drawing of the strategic distribution of forces should be based. Thus, these procedures lack of explainability. This may create mistrust in the application of DL models in crime prediction by law enforcement. Secondly, crime is phenomenon characterized by dynamic

changes. As it is said before the crime occurrences are based mostly on three contributing factors, people with criminal motives, suitable criminal targets, and the lack of guardians [4]. Consequently, by predicting the crime to be occurred in the area and reallocating the police forces, the factor "the lack of guardians" is changed. So, the crime distribution in this area will be affected in turn, which will make the predictions inaccurate.

Future Development

In the present study, as in the relevant literature, it appears that the spatio-temporal prediction of crime using DL can have very promising results, especially if problems such as those mentioned in the above subsections are addressed. To deal with these problems there are several approaches that can be followed. In particular, such approaches are:

- The creation of anonymized crime datasets by methods that do not have large spatial displacement of crime points so that spatial relativity is lost. An example of such methods is the New York dataset where the data are anonymized at the block level.
- The generation and use of additional dataset by the DL models. Specifically, as mentioned above, it is found that the special characteristics of crime in each region have a different effect on the production of forecasts. Consequently, by feeding the model with additional spatio-temporal data (i.e., extra features) characterizing each region such as the points of economic interest (supermarkets, banks, pedestrian streets, etc.) and population will cause more accurate personalized forecasts per region.
- The creation of models fed with different time aspects of the data. In particular, as it was found comparatively in the 3DCNN model case, the loss of information is the main failure factor of the model. In order to address this loss, the model can be simultaneously fed with averages of crimes that occurred in the area at different scales such as yearly, weekly and 8 hourly. This way the loss of information can be reduced, giving the model the ability to capture different perspectives of the data.
- The creation of more complex models that include more sub-modules that each of them extract a different kind of correlation between the data.
- Creating models that produce the information they relied on for their prediction in order to make the results explainable.
- Using different models to predict dynamic features that will feed the crime prediction models. Crime depends on dynamically changing characteristics such as the distribution of police forces and the weather. Therefore, the prediction of such data and feeding the model with them will help the model make as realistic predictions as possible.

Bibliography – References – Online sources

[1] M. Dewinter, C. Vandeviver, T. Vander Beken, and F. Witlox, "Analysing the Police Patrol Routing Problem: A Review," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 3, p. 157, Mar. 2020, doi: 10.3390/ijgi9030157.

[2] S. D. Johnson, L. Summers, and K. Pease, "Offender as Forager? A Direct Test of the Boost Account of Victimization," *J. Quant. Criminol.*, vol. 25, no. 2, pp. 181–200, Jun. 2009, doi: 10.1007/s10940-008-9060-8.

[3] W. Bernasco, "Them Again?: Same-Offender Involvement in Repeat and Near Repeat Burglaries," *Eur. J. Criminol.*, vol. 5, no. 4, pp. 411–431, Oct. 2008, doi: 10.1177/1477370808095124.

[4] X. Han, X. Hu, H. Wu, B. Shen, and J. Wu, "Risk Prediction of Theft Crimes in Urban Communities: An Integrated Model of LSTM and ST-GCN," *IEEE Access*, vol. 8, pp. 217222–217230, 2020, doi: 10.1109/ACCESS.2020.3041924.

[5] Department of Computer Science and Engineering, Adama Science and Technology University, Adama, Ethiopia. *et al.*, "Designing Time Series Crime Prediction Model using Long Short-Term Memory Recurrent Neural Network," *Int. J. Recent Technol. Eng. IJRTE*, vol. 9, no. 4, pp. 402–405, Nov. 2020, doi: 10.35940/ijrte.D5025.119420.

[6] S. F. Tekin and S. S. Kozat, "Crime Prediction with Graph Neural Networks and Multivariate Normal Distributions." arXiv, Dec. 16, 2021. Accessed: Jul. 12, 2022. [Online]. Available: http://arxiv.org/abs/2111.14733

[7] "NYPD Complaint Data Historic." https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i (accessed Jul. 12, 2022).

[8] "NYC OpenData." https://opendata.cityofnewyork.us/ (accessed Jul. 12, 2022).

[9] "GEODATA.gov.gr." http://geodata.gov.gr/en/ (accessed Jul. 12, 2022).

[10] H. Liu and D. E. Brown, "Criminal incident prediction using a point-pattern-based density model," *Int. J. Forecast.*, vol. 19, no. 4, pp. 603–622, Oct. 2003, doi: 10.1016/S0169-2070(03)00094-3.

[11] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, "Learning representations by back-propagating errors," p. 4, 1986.

[12] J. L. Elman, "Finding Structure in Time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990, doi: 10.1207/s15516709cog1402_1.

[13] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[14] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/neco_a_00990.

[15] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Aug. 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[16] K. Fu, Z. Chen, and C.-T. Lu, "StreetNet: preference learning with convolutional neural network on urban crime perception," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle Washington, Nov. 2018, pp. 269–278. doi: 10.1145/3274895.3274975.

[17] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning Graph Representations with Global Structural Information," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, Melbourne Australia, Oct. 2015, pp. 891–900. doi: 10.1145/2806416.2806512.

[18] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks." arXiv, Jul. 03, 2016. Accessed: Jul. 22, 2022. [Online]. Available: http://arxiv.org/abs/1607.00653

[19] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges." arXiv, May 02, 2021. Accessed: Jul. 22, 2022. [Online]. Available: http://arxiv.org/abs/2104.13478

[20] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "HOW POWERFUL ARE GRAPH NEURAL NETWORKS?," p. 17, 2019.
[21] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs." arXiv, Sep. 10, 2018. Accessed: Jul. 22, 2022. [Online]. Available: http://arxiv.org/abs/1706.02216
[22] L. E. Cohen and M. Felson, "Social Change and Crime Rate Trends: A Routine Activity Approach," *Am. Sociol. Rev.*, vol. 44, no. 4, p. 588, Aug. 1979, doi: 10.2307/2094589.
[23] P. Brantingham and P. Brantingham, "Criminality of place: Crime generators and crime attractors," *Eur. J. Crim. Policy Res.*, vol. 3, no. 3, pp. 5–26, Sep. 1995, doi: 10.1007/BF02242925.
[24] M. Hechter and S. Kanazawa, "Sociological Rational Choice Theory," p. 24, 1997.
[25] W. Safat, S. Asghar, and S. A. Gillani, "Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques," *IEEE Access*, vol. 9, pp. 70080–70094, 2021, doi: 10.1109/ACCESS.2021.3078117.
[26] E. W. Burgess, "The Study of the Delinquent as a Person," p. 25, 2022.
[27] N. Ibrahim, S. Wang, and B. Zhao, "Spatiotemporal Crime Hotspots Analysis and Crime Occurrence Prediction," in *Advanced Data Mining and Applications*, vol. 11888, J. Li, S. Wang, S. Qin, X. Li, and S. Wang, Eds. Cham: Springer International Publishing, 2019, pp. 579–588. doi: 10.1007/978-3-030-35231-8_42.
[28] G. Borowik, Z. M. Wawrzyniak, and P. Cichosz, "Time series analysis for crime forecasting," in *2018 26th International Conference on Systems Engineering (ICSEng)*, Sydney, Australia, Dec. 2018, pp. 1–10. doi: 10.1109/ICSENG.2018.8638179.
[29] X. Zhang, L. Liu, L. Xiao, and J. Ji, "Comparison of Machine Learning Algorithms for Predicting Crime Hotspots," *IEEE Access*, vol. 8, pp. 181302–181310, 2020, doi: 10.1109/ACCESS.2020.3028420.
[30] M. Hou, X. Hu, J. Cai, X. Han, and S. Yuan, "An Integrated Graph Model for Spatial–Temporal Urban Crime Prediction Based on Attention Mechanism," *ISPRS Int. J. Geo-Inf.*, vol. 11, no. 5, p. 294, Apr. 2022, doi: 10.3390/ijgi11050294.
[31] S. Wu, C. Wang, H. Cao, and X. Jia, "Crime Prediction Using Data Mining and Machine Learning," in *The 8th International Conference on Computer Engineering and Networks (CENet2018)*, vol. 905, Q. Liu, M. Mısır, X. Wang, and W. Liu, Eds. Cham: Springer International Publishing, 2020, pp. 360–375. doi: 10.1007/978-3-030-14680-1_40.
[32] S. K. Dash, I. Safro, and R. S. Srinivasamurthy, "Spatio-temporal prediction of crimes using network analytic approach." arXiv, Oct. 30, 2018. Accessed: Jul. 25, 2022. [Online]. Available: http://arxiv.org/abs/1808.06241
[33] L. McClendon and N. Meghanathan, "Using Machine Learning Algorithms to Analyze Crime Data," *Mach. Learn. Appl. Int. J.*, vol. 2, no. 1, pp. 1–12, Mar. 2015, doi: 10.5121/mlaij.2015.2101.
[34] J. Q. Yuki, Md. M. Q. Sakib, Z. Zamal, K. M. Habibullah, and A. K. Das, "Predicting Crime Using Time and Location Data," in *Proceedings of the 2019 7th International Conference on Computer and Communications Management*, Bangkok Thailand, Jul. 2019, pp. 124–128. doi: 10.1145/3348445.3348483.
[35] P. Stalidis, T. Semertzidis, and P. Daras, "Examining Deep Learning Architectures for Crime Classification and Prediction," *Forecasting*, vol. 3, no. 4, pp. 741–762, Oct. 2021, doi: 10.3390/forecast3040046.
[36] U. M. Ramirez-Alcocer, E. Tello-Leal, and J. A. Mata-Torres, "Predicting Incidents of Crime through LSTM Neural Networks in Smart City Domain," p. 6, 2019.
[37] A. Krishnan, A. Sarguru, and A. C. S. Sheela, "PREDICITVE ANALYSIS OF CRIME DATA USING DEEP LEARNING," p. 10.
[38] S. Wang and K. Yuan, "Spatiotemporal Analysis and Prediction of Crime Events in Atlanta Using Deep Learning," in *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, Xiamen, China, Jul. 2019, pp. 346–350. doi: 10.1109/ICIVC47709.2019.8981090.
[39] A. Stec and D. Klabjan, "Forecasting Crime with Deep Learning," *ArXiv180601486 Cs Stat*, Jun. 2018, Accessed: Mar. 06, 2022. [Online]. Available: http://arxiv.org/abs/1806.01486

[40]"Chicago Data Portal." https://data.cityofchicago.org/
[41]X. Ye, L. Duan, and Q. Peng, "Spatiotemporal Prediction of Theft Risk with Deep Inception-Residual Networks," *Smart Cities*, vol. 4, no. 1, pp. 204–216, Jan. 2021, doi: 10.3390/smartcities4010013.
[42]N. Esquivel, O. Nicolis, B. Peralta, and J. Mateu, "Spatio-Temporal Prediction of Baltimore Crime Events Using CLSTM Neural Networks," *IEEE Access*, vol. 8, pp. 209101–209112, 2020, doi: 10.1109/ACCESS.2020.3036715.
[43]B. Wang, D. Zhang, D. Zhang, P. J. Brantingham, and A. L. Bertozzi, "Deep Learning for Real Time Crime Forecasting," *ArXiv170703340 Cs Math Stat*, Jul. 2017, Accessed: May 02, 2022. [Online]. Available: http://arxiv.org/abs/1707.03340
[44]B. Wang, P. Yin, A. L. Bertozzi, P. J. Brantingham, S. J. Osher, and J. Xin, "Deep Learning for Real-Time Crime Forecasting and its Ternarization," *ArXiv171108833 Cs Math Stat*, Nov. 2017, Accessed: May 02, 2022. [Online]. Available: http://arxiv.org/abs/1711.08833
[45]J. Zhang, Y. Zheng, and D. Qi, "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction." arXiv, Jan. 10, 2017. Accessed: Jul. 23, 2022. [Online]. Available: http://arxiv.org/abs/1610.00081
[46]K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." arXiv, Dec. 10, 2015. Accessed: Jul. 24, 2022. [Online]. Available: http://arxiv.org/abs/1512.03385
[47]K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks." arXiv, Jul. 25, 2016. Accessed: Jul. 24, 2022. [Online]. Available: http://arxiv.org/abs/1603.05027
[48]S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 922–929, Jul. 2019, doi: 10.1609/aaai.v33i01.3301922.
[49]C. Li, Z. Cui, W. Zheng, C. Xu, and J. Yang, "Spatio-Temporal Graph Convolution for Skeleton Based Action Recognition." arXiv, Feb. 27, 2018. Accessed: Jul. 31, 2022. [Online]. Available: http://arxiv.org/abs/1802.09834
[50]B. Nikparvar, Md. M. Rahman, F. Hatami, and J.-C. Thill, "Spatio-temporal prediction of the COVID-19 pandemic in US counties: modeling with a deep LSTM neural network," *Sci. Rep.*, vol. 11, no. 1, p. 21715, Dec. 2021, doi: 10.1038/s41598-021-01119-3.
[51]Y. Duan, Y. L.V., and F.-Y. Wang, "Travel time prediction with LSTM neural network," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Nov. 2016, pp. 1053–1058. doi: 10.1109/ITSC.2016.7795686.
[52]S. Kim, S. Hong, M. Joh, and S. Song, "DeepRain: ConvLSTM Network for Precipitation Prediction using Multichannel Radar Data." arXiv, Nov. 07, 2017. Accessed: Aug. 01, 2022. [Online]. Available: http://arxiv.org/abs/1711.02316
[53]D. Wang, Y. Yang, and S. Ning, "DeepSTCL: A Deep Spatio-temporal ConvLSTM for Travel Demand Prediction," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Jul. 2018, pp. 1–8. doi: 10.1109/IJCNN.2018.8489530.
[54]N. C. Petersen, F. Rodrigues, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Syst. Appl.*, vol. 120, pp. 426–435, Apr. 2019, doi: 10.1016/j.eswa.2018.11.028.
[55]X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," p. 9.
[56]B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," *Proc. Twenty-Seventh Int. Jt. Conf. Artif. Intell.*, pp. 3634–3640, Jul. 2018, doi: 10.24963/ijcai.2018/505.
[57]X. Geng *et al.*, "Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 3656–3663, Jul. 2019, doi: 10.1609/aaai.v33i01.33013656.
[58]X. Wang *et al.*, "Traffic Flow Prediction via Spatial Temporal Graph Neural Network," in *Proceedings of The Web Conference 2020*, Taipei Taiwan, Apr. 2020, pp. 1082–1092. doi: 10.1145/3366423.3380186.

[59] A. Vaswani *et al.*, "Attention is All you Need," p. 11.
[60] U. M. Butt, S. Letchmunan, F. H. Hassan, M. Ali, A. Baqir, and H. H. R. Sherazi, "Spatio-Temporal Crime HotSpot Detection and Prediction: A Systematic Literature Review," *IEEE Access*, vol. 8, pp. 166553–166574, 2020, doi: 10.1109/ACCESS.2020.3022808.
[61] "World Geodetic System (WGS84)." https://earth-info.nga.mil/index.php?dir=wgs84&action=wgs84
[62] "United States Census Bureau." https://www.census.gov/ (accessed Aug. 15, 2022).
[63] "NYC NEIGHBORHOODS." https://data.beta.nyc/dataset/pediacities-nyc-neighborhoods (accessed Apr. 15, 2022).
[64] "Geopandas." https://geopandas.org/en/stable/ (accessed Aug. 16, 2022).
[65] "pandas." https://pandas.pydata.org/ (accessed Aug. 16, 2022).
[66] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. Accessed: Aug. 18, 2022. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02016R0679-20160504&qid=1532348683434
[67] "Google Maps Platform." https://developers.google.com/maps/documentation/geocoding (accessed Aug. 18, 2022).
[68] "Nominatim." https://nominatim.org/ (accessed Aug. 18, 2022).
[69] "Geopy." https://geopy.readthedocs.io/en/stable/ (accessed Aug. 16, 2022).
[70] "Regions of Greece." https://geodata.gov.gr/dataset/periphereies-elladas (accessed Aug. 16, 2022).
[71] "Athens Municipality Neighborhoods Boundaries Dataset." https://geodata.gov.gr/en/dataset/op1a-re1tov1wv-anuou-a0nvaiwv (accessed Aug. 16, 2022).
[72] "Weights & Biases." https://wandb.ai/ (accessed Aug. 16, 2022).
[73] "sklearn MinMaxScaler," Aug. 16, 2022. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[74] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." arXiv, Mar. 02, 2015. Accessed: Aug. 24, 2022. [Online]. Available: http://arxiv.org/abs/1502.03167
[75] "BatchNormalization keras Layer," Aug. 16, 2022. https://keras.io/api/layers/normalization_layers/batch_normalization/
[76] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," p. 30.
[77] "Dropout keras Layer," Aug. 16, 2022. https://keras.io/api/layers/regularization_layers/dropout/
[78] "RMSprop keras Optimizer." https://keras.io/api/optimizers/rmsprop/ (accessed Aug. 16, 2022).
[79] "Keras ConvLSTM2D layer." https://keras.io/api/layers/recurrent_layers/conv_lstm2d/ (accessed Aug. 16, 2022).
[80] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." arXiv, Jan. 29, 2017. Accessed: Aug. 25, 2022. [Online]. Available: http://arxiv.org/abs/1412.6980
[81] "keras Adam optimizer." https://keras.io/api/optimizers/adam/ (accessed Aug. 16, 2022).
[82] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning." arXiv, Jan. 11, 2018. Accessed: Aug. 26, 2022. [Online]. Available: http://arxiv.org/abs/1603.07285
[83] "keras Traffic forecasting using graph neural networks and LSTM example." https://keras.io/examples/timeseries/timeseries_traffic_forecasting/ (accessed Dec. 08, 2022).
[84] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015. Accessed: Sep. 06, 2022. [Online]. Available: http://arxiv.org/abs/1409.1556
[85] H. Hu, S. Lan, Y. Jiang, Z. Cao, and F. Sha, "FastMask: Segment Multi-scale Object Candidates in One Shot." arXiv, Apr. 11, 2017. Accessed: Sep. 06, 2022. [Online]. Available: http://arxiv.org/abs/1612.08843

[86] S. Chainey, L. Tompson, and S. Uhlig, "The Utility of Hotspot Mapping for Predicting Spatial Patterns of Crime," Secur J, vol. 21, no. 1–2, pp. 4–28, Feb. 2008, doi: 10.1057/palgrave.sj.8350066.
[87] G. Hinton, "RMSprop Algorithm," 2012. Accessed: Sep. 06, 2022. [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
[88] "Recurrent Neural Network" https://en.wikipedia.org/wiki/Recurrent_neural_network (accessed Aug. 16, 2022)
[89] "Long Short Term Memory" https://en.wikipedia.org/wiki/Long_short-term_memory (accessed Aug. 16, 2022)
[90] "Convolutions"https://people.minesparis.psl.eu/fabien.moutarde/ES_MachineLearning/TP_convNets/convnet-notebook.html (accessed Aug. 16, 2022)
[91] "GNNVis: A Visual Analytics Approach for Prediction Error Diagnosis of Graph Neural Networks"https://www.researchgate.net/publication/346143176_GNNVis_A_Visual_Analytics_Approach_for_Prediction_Error_Diagnosis_of_Graph_Neural_Networks (accessed Aug. 16, 2022)
[92] "Graph Neural Network" https://theaisummer.com/Graph_Neural_Networks (accessed Aug. 16, 2022)
[93] "Graph Convolutional Networks (GCN)" https://www.topbots.com/graph-convolutional-networks/ (accessed Aug. 16, 2022)
[94] "PEDIACITIES-NYC-NEIGHBORHOODS" https://data.beta.nyc/dataset/pediacities-nyc-neighborhoods/resource/35dd04fb-81b3-479b-a074-a27a37888ce7 (accessed Aug. 16, 2022)
[95] "3D Convolutions" https://www.kaggle.com/code/shivamb/3d-convolutions-understanding-use-case/notebook (accessed Aug. 16, 2022)