VisCrimePredict: A System for Crime Trajectory Prediction and Visualisation from Heterogeneous data sources

Ahsan Morshed[1], Abdur Rahim Mohammad Forkan[1], Pei-Wei Tsai[1], Prem Prakash Jayaraman[1],
Timos Sellis[1], Dimitrios Georgakopoulos[1], Irene Moser[1], Rajiv Ranjan[2]
[1]Department of CSSE, Swinburne University of Technology, Melbourne, Victoria, Australia
[2]School of Computing, Urban Sciences Building, Newcastle University
{amorshed,fforkan,ptsai,pjayaraman,tsellis,dgeorgakopoulos,imoser}@swin.edu.au,raj.ranjan@ncl.ac.uk

ABSTRACT

Open multidimensional data from social media and similar sources often carries insightful information on social issues. With the increase of high volume data and the proliferation of visual analytics platforms, it becomes easier for users to interact with and select meaningful information from large data sets. The prevention of crime is a crucial issue for law-enforcing agencies tasked with maintaining societal stability. The ability to visualise crime patterns and predict imminent incidents accurately opens new possibilities in crime prevention. In this paper, we present VisCrimePredict, a system that uses visual and predictive analytics to map out crimes that occurred in a region/neighbourhood. VisCrimePredict is underpinned by a novel algorithm that creates trajectories from heterogeneous data sources such as open data and social media with the aim to report incidents of crime. VisCrimePredict uses a Long Short Term Memory (LSTM) algorithm for trajectory prediction. A proof of concept implementation of VisCrimePredict and an experimental evaluation of crime trajectory prediction accuracy using LSTM neural network concludes the paper.

KEYWORDS

Visual analytics, Crime Trajectory, LSTM Network, Trajectory prediction, Twitter Data

ACM Reference Format:
Ahsan Morshed[1], Abdur Rahim Mohammad Forkan[1], Pei-Wei Tsai[1], Prem Prakash Jayaraman[1], Timos Sellis[1], Dimitrios Georgakopoulos[1], Irene Moser[1], Rajiv Ranjan[2]. 2019. VisCrimePredict: A System for Crime Trajectory Prediction and Visualisation from Heterogeneous data sources. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3297280.3297388

1 INTRODUCTION

Recently, the increasing volume of multidimensional data from open data sources (e.g., historical data, crowdsensed data [9, 11]) and the ability to collect vast amounts of data from social media has presented new opportunities to solve social and community issues such as crime, health and migration. Visual analytics have been employed [18] to assist policy makers, law enforcement agencies and citizens in taking timely and effective decisions. They also help provide microscopic insights from the information gathered from such diverse data sources.

Most existing work and systems [21, 24, 25] illustrate historical crime information at an abstract or regional level. Andresen, Curman and Linning [3] provided a longitudinal analysis based on 16 years of crime data in Vancouver, Canada while Kounadi et al. [16] used a combination of historic and social media data to create population models to identify persons at risk. While these studies contribute significant advances in crime predictions, visual analytics including the capture and visual representation of crime information over time are still lacking.

In general, a trajectory is the path that a moving object follows through space as a function of time. Thus, it can be captured as a time-stamped series of location points. State-of-the-art surveys [17, 26] state that most modern trajectory algorithms are categorised based on factors such as distance, velocity, semantics, similarity, priority queue and segmentation. Segmentation-based algorithms are often used for the construction of trajectories derived from social media data. Most of the existing trajectory segmentation algorithms focus on refining information, such as the moving mode or the heading, by segmenting the trajectories into homogeneous segments based on criteria such as changing points and the uniformity in the duration or the distance [4, 19]. Many of them utilise the available information in both the spatial and the temporal domains to form the segmentation criteria. Some automatic or semi-automatic trajectory segmentation methods even require the addition of labels to find the segmentation points. For example, the Reactive Greedy Randomised Adaptive Search Procedure for semantic Semi-supervised Trajectory Segmentation (RGRASP-SemTS) algorithm [13] requires domain experts to label a subset of an existing trajectory set and a cost function to generate the segmentation criteria. However, crime trajectories do not have characteristics such as velocity, curvature and sinuosity. Existing trajectory segmentation criteria are not suitable for use in crime trajectory segmentation, since crime records are stringed into trajectories based on the categories of the crimes. The specific characteristics of the crime trajectory also impacts the visual analysis of crime information.

To predict future crimes from a crime trajectory, criminology relies on a number of interconnected theories. In particular, environmental crime theories discuss the influence of the environment on crime and assume that relatively rational actors take deliberate actions aiming to maximise their return when committing crime

SAC '19, April 8–12, 2019, Limassol, Cyprus
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297388

[2]. The assumption of cause and effect opens up opportunities for modelling crime, and crime prediction methods have used a variety of machine learning and statistical methods, such as deep learning, regression analysis, kernel density estimation (KDE), support vector methods and similar methods [1, 6, 7]. All models hinge on geographical location, time and nature of crimes [5, 15]. Zhuang et al. [27] have demonstrated the suitability of Long Short-Term Memory (LSTM) for developing trajectory models of crime.

In this paper, we propose, implement and demonstrate VisCrimePredict, a system for the visual analysis of multidimensional data on the macroscopic and microscopic levels to show trajectories of crime based on their spatial and temporal characteristics. The VisCrimePredict system incorporates a novel *Threshold-based Spatial Temporal Segmentation* (TbSTS) algorithm for the crime trajectory segmentation and uses Long Short Term Memory (LSTM) algorithm, a variant of well-known recurrent neural network, for trajectory prediction.

The rest of this paper is structured as follows; Section 2 describes the overview of the system, Section 3 briefly describes the proposed trajectory algorithm, Section 4 illustrates the crime trajectory prediction model, Section 5 presents a demonstration of the algorithm and case-study scenarios, Section 6 describes the result of trajectory prediction model and finally, Section 7 concludes the paper.

2 VISCRIMEPREDICT SYSTEM

In this section, we provide an overview of the system including the data collection mechanism to assimilate data from open data sources[1] and social media (e.g., Twitter) and its spatio-temporal characteristics.

2.1 System Overview

Figure 1 shows an overview of the VisCrimePredict System. Firstly, we construct the data management layer where we collect real-time open-source data and store it in a NoSQL database in JSON format for ease of manipulation and analysis. We also collect real-time social media data from the Twitter API over six months. In the analytic layer, we process the social media data using existing Natural Language Processing (NLP) tools (i.e, SpaCy)[20] in order to remove unnecessary key phrases and extract features such as the nature of crimes, location, and incident time in JSON format. To accomplish this, we have trained the existing NLP model by manually annotating 2000 crime-related tweets and used this model to extract only the crime related tweets. All information captured is integrated with existing data in the NoSQL database. These two sets of data act as inputs to the 'Trajectory Model' component, which computes a crime trajectory using the TbSTS algorithm. The computed trajectories are used as inputs for 'Trajectory Prediction' model. Having been trained with historical trajectory data, it computes the trajectory prediction. Finally, the visual analytics component, developed using Kepler.js [2] visually presents the crime trajectories generated, comprising the prediction of trajectories in a region, the nature of each crime, and additional information obtained from Twitter about the region.

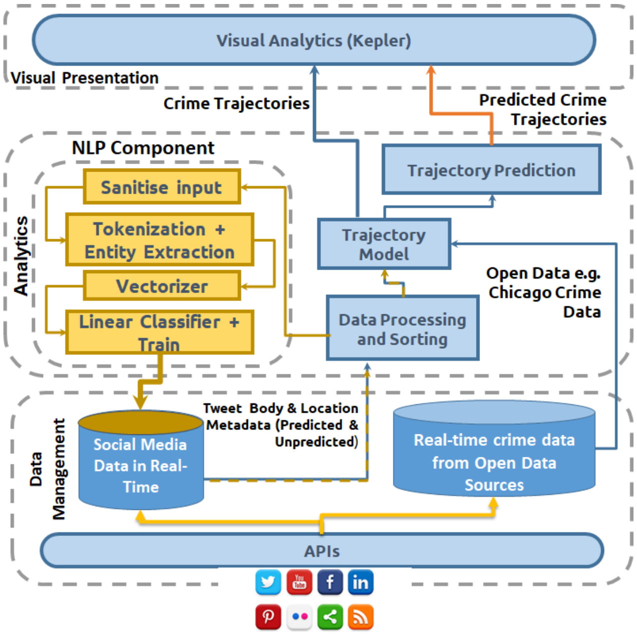

Visual Analytics (Kepler)
Visual Presentation
Crime Trajectories
Predicted Crime Trajectories
NLP Component
Trajectory Prediction
Sanitise input
Tokenization + Entity Extraction
Vectorizer
Linear Classifier + Train
Analytics
Trajectory Model
Open Data e.g. Chicago Crime Data
Data Processing and Sorting
Tweet Body & Location Metadata (Predicted & Unpredicted)
Data Management
Social Media Data in Real-Time
Real-time crime data from Open Data Sources
APIs

Figure 1: The VisCrimePredict System Diagram.

2.2 Spatial and temporal Characteristics

Taking into account Tobler's first rule of geography [23] "Everything is related to everything else, but near things are more related than distant things.", we must prioritise identifying the nature and location of crimes. Based on this notion, we consider date and time, crime type, crime description, geo-coordinates (latitude and longitude), location description and year as the main spatial and temporal characteristics.

3 CRIME TRAJECTORY MODELLING

The VisCrimePredict System system fetches the crime data from the records in the NoSQL database. Before the data can be analytically presented, some data manipulation is essential for reconstructing and segmenting trajectories from the raw data. Trajectory reconstruction is the process of merging different segments into one continuous trajectory. Trajectory segmentation is defined as splitting the continuous trajectory into homogeneous segments based on certain criteria. In general, these segments do not overlap and the segments are naturally equal to or shorter than the original trajectory. The details of the data manipulation and trajectory reconstruction/segmentation in VisCrimePredict are revealed in the following subsections.

3.1 Trajectory Reconstruction

Trajectory reconstruction is a technique commonly used to help track objects in computer vision. It is helpful for understanding the evolution of the targeted objects. For example, Kayumbi et al. [14] used the global trajectory reconstruction technique to track football players from independent video sequences captured by different cameras. Unlike object movement trajectories or trip-based trajectory reconstructions, crime records are recorded as independent

[1] such as open data sets of crime incidents, e.g., https://data.cityofchicago.org/Public-Safety/Crimes-2018/3i3m-jwuy
[2] https://kepler.gl/

events and a trajectory does not naturally arise from the string of events in our application. Thus, the crime records have to be properly refined in the preparation. To extract the crime trajectory from the input data, the crime type attribute in the crime event record is used as the label to group the crime records into multiple categories. Based on the date and time value in the records, the trajectories of the corresponding crime categories can be reconstructed. Nevertheless, linking all crime events in the same crime category into a single trajectory is not sufficient to provide information for further analysis and thus the trajectory reconstruction is followed by a trajectory segmentation process.

A simple example of a trajectory reconstruction process is the separation of robbery events from theft events so we can construct separate trajectories for both categories in the visualisation. However, in a single month, thousands of thefts can take place. Linking all of them into a single trajectory does not contain useful information for a law enforcement officer. Therefore, our next step in the process is trajectory segmentation.

3.2 Threshold-based Spatial Temporal Segmentation Algorithm (TbSTS)

To support the visual analytics provided by VisCrimePredict, the Period of Interest (PoI) and the Region of Interest (RoI) are variables decided by the user according to the chosen subject of analysis. Thus, the PoI and RoI in this application are kept as user-defined values. The PoI determines the size of the sliding window for filtering while the RoI helps focus the analysis in the bounded geographic area. All trajectories reconstructed in the above section are treated as the input and are fed into TbSTS [22]. The detail of the proposed TbSTS is given as follows.

Let $\mathbf{S} = \{S_i | i = 1, 2, \cdots, N\}$, $S_i = \{s_{i1}, s_{i2}, \cdots, s_{ij}, \text{ s.t. } j = |S_i|\}$, $s_{ij} =< lat, lng, ts >$ denote N reconstructed trajectories sorted by time in different crime categories as described in the above section, where lat, lng, and ts represent the latitude, the longitude, and the timestamp, respectively. The sliding window method is used to bound the analytic time frame defined by the user for extracting points on the trajectory (denoted by $\hat{S}_i$, where $\hat{S}_i \subset S_i$), which fit in the sliding window. Starting with element $j + 1$ element in $\hat{S}_i$, find the distance measure of both the $< lat, long >$ and the ts for the j and $j + 1$ elements by Equation (1):

$$\begin{bmatrix} D_s \\ D_t \end{bmatrix} = \begin{bmatrix} ED(\hat{S}_{i(j-1)_{<lat,long>}}, \hat{S}_{ij_{<lat,long>}}) \\ ED(\hat{S}_{i(j-1)_{<ts>}}, \hat{S}_{ij_{<ts>}}) \end{bmatrix} \quad (1)$$

where D_s and D_t denote the spatial and temporal distances between $\hat{S}_{i(j-1)}$ and $\hat{S}_{ij}$, respectively, and $ED(\cdot)$ is a function that returns the Euclidean distance of the input element. The trajectory $\hat{S}_i$ is split into two parts by removing the link between $\hat{S}_{i(j-1)}$ and $\hat{S}_{ij}$ if $D_s > T_s \bigvee D_t > T_t$, where T_s and T_t represent the user defined thresholds of the distance and the time difference between two temporal continuous events. The split trajectory is denoted by $\hat{S}_i^m$, where $\hat{S}_i = \hat{S}_i^1 \cup \hat{S}_i^2 \cup \cdots \cup \hat{S}_i^M$, M is the number of segmented trajectories of $\hat{S}_i$, and $\hat{S}_i^p \cap \hat{S}_i^q = \emptyset \; \forall p \neq q,\; p = 1, 2, \cdots, M$ and $q = 1, 2, \cdots, M$. The pseudocode of TbSTS is revealed in Algorithm 1.

```
**Input:** $< \hat{S}_i, T_s, T_t >$
**Output:** $< \{\hat{S}_i^p \text{ s.t. } p = 1, 2, \cdots, M\} >$
c ← 0; d ← 1; L = $|\hat{S}_i|$;
**forall** $j$ = 2 to $L$ **do**
    $D_s = ED(\hat{S}_{i(j-1)_{<lat,long>}}, \hat{S}_{ij_{<lat,long>}})$;
    $D_t = ED(\hat{S}_{i(j-1)_{<ts>}}, \hat{S}_{ij_{<ts>}})$;
    **if** $D_s > T_s$ or $D_t > T_t$ **then**
        c ← c + 1;
        $\hat{S}_i^c = \{\hat{S}_{id}, \hat{S}_{i(d+1)}, \cdots, \hat{S}_{i(j-1)} \mid (d+1) < j\}$;
        $\hat{S}_i \leftarrow (\hat{S}_i - \hat{S}_i^c)$;
        d ← j;
    **end**
**end**
**if** $\hat{S}_i \neq \emptyset$ **then**
    c ← c + 1;
    $\hat{S}_i^c = \hat{S}_i$;
**end**
```

Algorithm 1: TbSTS Algorithm

Table 1: Example Data within the PoI and RoI

S/N	Date/Time	Lat	Long	Category
1	13/12/2018 23:56:00	33.4434	-142.3398	Theft
2	14/12/2018 01:01:00	33.4451	-142.3436	Theft
3	14/12/2018 03:35:00	33.7698	-147.3247	Theft
4	14/12/2018 07:30:00	33.4413	-142.3295	Robbery
5	14/12/2018 08:58:00	33.4427	-142.3203	Theft

3.3 TbSTS example

To further illustrate how the TbSTS algorithm works, a simple example is provided in this section. Assume that we have five crime records (see Table 1), which are extracted by the sliding window within the given PoI and RoI.

In the trajectory reconstruction phase (described in section 3.1), the reconstructed trajectories from Table 1 are obtained shown in Figure 2. Then, both trajectories shown in Figure 2 are fed into TbSTS for trajectory segmentation.

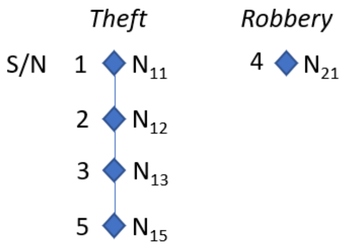


Figure 2: Reconstructed Trajectories

In this example, the total number of trajectories (N) is 2, T_s and T_t are assumed to be 180 (minutes) and 1,000 (meters), respectively. By feeding these trajectories into TbSTS, we can obtain one trajectory with three grouped events in the theft category and one grouped event in the robbery category (see Figure 3).

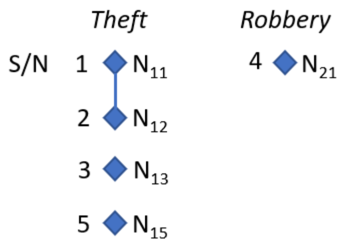


Figure 3: Segmentation Results of Trajectories

Based on the criteria given above, the link between N_{12} and N_{13} is removed because the distance between these two records exceeds T_s. In addition, the link between N_{13} and N_{14} is removed because the time difference between these two records exceeds T_t.

3.4 Trajectory filtering

In addition to the time, location and category (e.g., theft, robbery, burglary) information, crime data contains descriptions of crime locations (e.g., street, apartment, school, shop), police district number, city council ward number, police beat area number etc. All these are categorical numeric values, hence using TbSTS, trajectories for different crime categories can be formed by applying one or more filtering criteria using these attributes (e.g., create trajectories of robbery incidents in shopping mall with T_s km and T_t=30 minutes, create trajectories of theft incidents in district area 2 with T_s=5 km and T_t = 60 minutes). Let $S_F = \{S_i | i = 1, 2, \cdots, M\}$ be a filtered trajectory after applying filtering criteria F in trajectory S then $S_F \subseteq S$.

4 CRIME TRAJECTORY PREDICTION

The trajectories produced by TbSTS are used as current knowledge that serves as a basis for the prediction and visualisation of possible future crime events. Once we identify specific crime trajectories **S** for a crime category using T_s and T_t values, these are used to predict the next location and time in the trajectory. Let D_{s_i} and D_{t_i} be the spatial and temporal distances for the i-th trajectory from the $(i-1)$-th trajectory after applying reconstruction and TbSTS with or without filtering. Our aim is to predict the subsequent spatial and temporal distance values $D_{s_{i+1}}$ and $D_{t_{i+1}}$ of a given trajectory which enable us to identify the next location and time of a possible crime event. This is demonstrated in Figure 4. The X and Y axis in the figure refer to the spatial (D_s) and temporal (D_t) distances that correspond to the geographical distance between two coordinates and the time difference between two incidents in minutes. The blue line indicates the observed trajectory in current time and the red line indicates a possible recommendation of distance and time for future crime event for a specific crime category.

Predicting future (D_s,D_t) values of the selected trajectory is a regression problem. However, artificial neural networks have been successfully applied in solving such prediction problems [12]. In this work, we use a Long Short-Term Memory (LSTM) network [10]. LSTM is an extension of recurrent neural networks (RNN) [8], which are particularly well suited to predicting numeric values from time series or sequence data. RNN are networks with loops in them, allowing information to persist. The LSTM learning model decides how much of the previous internal state to forget and stores information required for new inputs in the internal memory. Since the trajectories we defined have spatio-temporal characteristics, an LSTM network can easily adapt to such changes. It is also capable of learning long-term relationships between points in a trajectory.

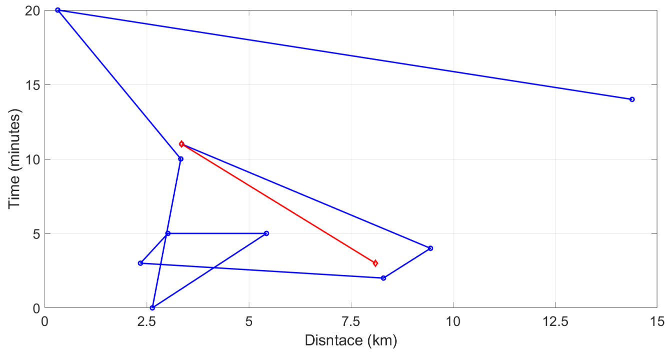


Figure 4: Trajectory prediction using spatial and temporal distance measure

4.1 Modelling LSTM for crime trajectory prediction

The core components of an LSTM neural network are an input layer and an LSTM layer. The input layer takes inputs as a sequence of trajectories into the network. An LSTM layer learns long-term dependencies between events of sequence data. The resulting network consists of fully connected input and output layers, similar to a recurrent neural network. This is shown in Figure 5.

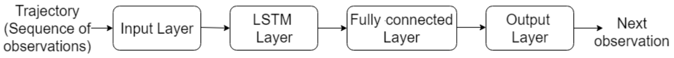


Figure 5: The LSTM Network for trajectory prediction

In the LSTM layer, a common LSTM unit is comprised of a memory cell, an input gate, an output gate and a forget gate. This is shown in Figure 6. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

Input, output and forget gates are denoted by i, o and f respectively. Let W be the the recurrent connection at the previous hidden layer and current hidden layer and U be the weight matrix connecting the trajectory input to the current hidden layer.

The hidden state h_t of an LSTM unit is calculated as expressed in Equations 2 – 7.

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \tag{2}$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \tag{3}$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \tag{4}$$

$$\bar{C}_t = tanh(x_t U^g + h_{t-1} W^g) \tag{5}$$

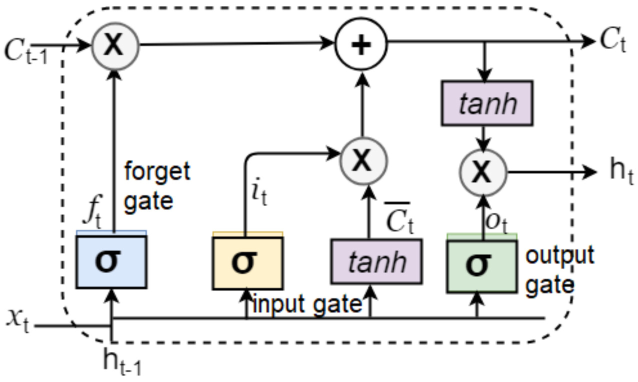


Figure 6: An LSTM unit in the LSTM network

$$C_t = \sigma(f_t * C_{t-1} + i_t * \bar{C}_t) \tag{6}$$

$$h_t = tanh(C_t) * o_t \tag{7}$$

All gates have the same equations with different parameters. These are called gates because the sigmoid function outputs the values of these vectors between 0 and 1. By multiplying them element-wise with another vector (ref. Equations 2–4), the network decides how much of that vector it wants to pass to the next layer. All gates have the same dimensions which is equal to the size of the hidden states. i defines how much of the newly computed state for the current input trajectory will pass through, f computes how much of the previous state will go through and o calculates how much of the internal state will be exposed to the next layer (fully connected layer) and next time step.

$\bar{C}$ is called candidate hidden state or cell state which is computed based on the current input and the previous hidden state. C is the internal memory of the LSTM unit, a combination of the previous memory multiplied by the forget gate and the newly computed hidden state, multiplied by the input gate. Thus, intuitively, it is a combination of the previous memory and the new input. In the example of our trajectory model, we want to include the information of the current incident in the cell state as well as all previous incidents in the trajectory path. It is possible to ignore the old memory completely (forget gate all 0s) or ignore the newly computed state completely (input gate all 0s). However, we used both in our network structure as we aim for a solution between these two extremes. h_t is output hidden state, computed by multiplying the memory with the output gate. Not all of the internal memory may be relevant to the hidden state used by other units in the network. *tanh* is used to adjust the values to the range 0 - 1.

Given information about a location (or point) in the trajectory sequence, our LSTM-based prediction model estimates the next value in the sequence. Here, the first value in the sequence is remembered across multiple samples. This would not be possible with Multilayer Perceptron and other non-recurrent neural networks. The LSTM network learns the difference between the sequence as well as between long sequences via backpropagation through the temporal value (difference in time).

The size of the input layer is equal to the number of features, which is two in our case (spatial and temporal distance from previous point). We have used 125 hidden units in the LSTM layer.

We trained and evaluated various LSTM network with different combinations of trajectories. In our case, an element in the sequence is a vector of numerical value. The simplest sequence element vector contains 2 values, D_s and D_t. The more complex sequence element contains other filtering attributes (e.g., crime location description) and attempts to predict all values in the vector of the next sequence. However, the increase in attributes in the element vector decreases the prediction performance. For simplicity we have only used element vector with two values (D_s and D_s) and used other attributes as filtering criteria (F).

5 CASE-STUDY DEMONSTRATION AND SCENARIOS

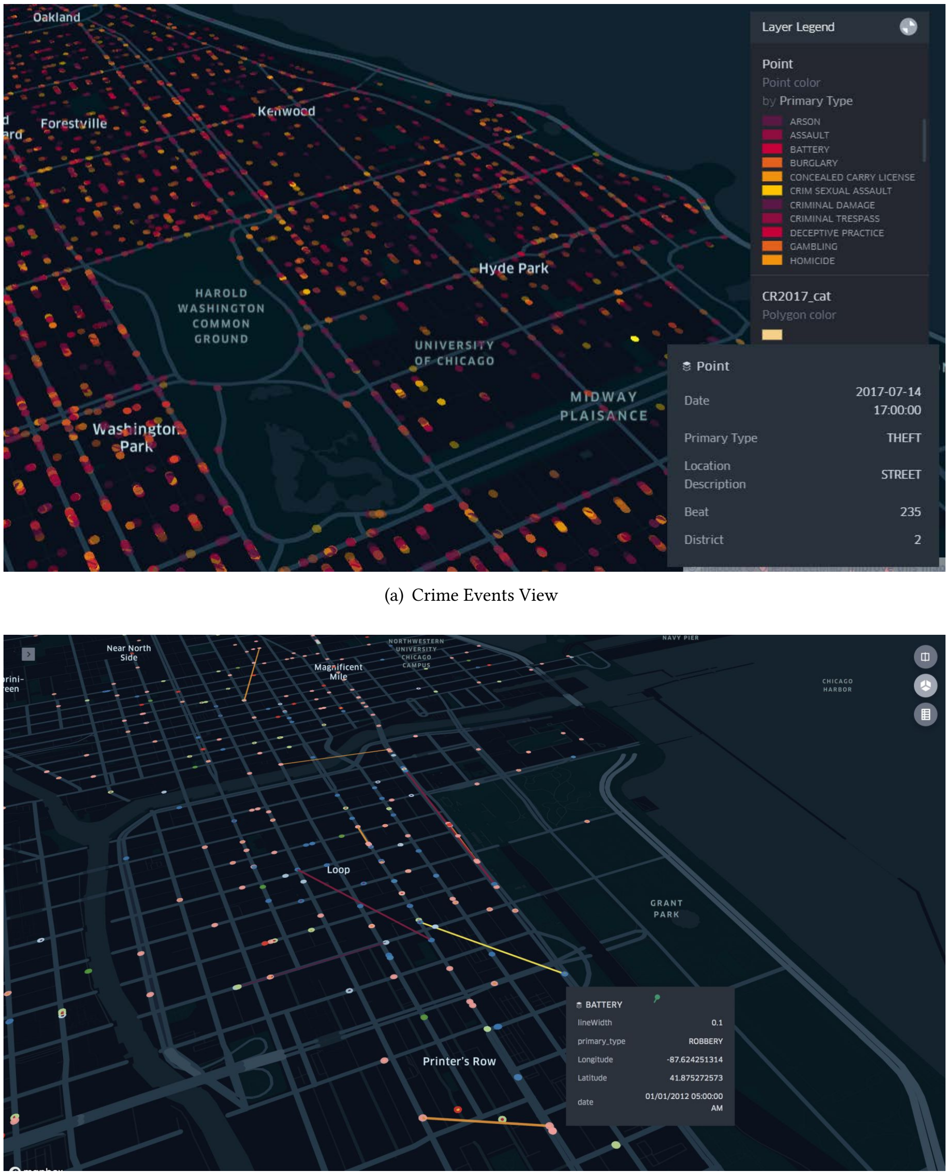
Our implementation can extract predictions from various crime data sets and real-time social media data. It uses NLP techniques to pre-process live data for extracting crime information, time and geo-location. We demonstrate its capabilities using a historic data set and related social media data.

As a case study for the experimentation, we have obtained the Chicago crime data set from 2001 to 2018 and a real-time data set consisting of 30,000 crime-related tweets for a period of three months, collected from the Twitter API, and matched with the Chicago data source[3]. For the LSTM-based prediction, we have used the crime data of the year 2017 only as it contains a sufficient number of sample trajectories, comprising a total of 31 crime categories, 128 location descriptions, 23 districts, 77 community areas, 50 wards and 274 beats. A number of open source tools and technologies including Python, Nodejs and the Kepler visualisation tool have been used in the implementation of VisCrimePredict.

5.1 Event Observation

Without defining any filtering criteria such as PoI, thresholds, districts, beats, the system displays all incidents of crime (represented as dots). Different colours are used to represent different crime categories. The visualisation provides the user with an overview of the historical crime density over time in the RoI, i.e. the Chicago area.

[3] Available: https://data.cityofchicago.org/Public-Safety/Crimes-2018/3i3m-jwuy



Point

Date	2017-07-14 17:00:00
Primary Type	THEFT
Location Description	STREET
Beat	235
District	2

(a) Crime Events View

BATTERY

lineWidth	0.1
primary_type	ROBBERY
Longitude	-87.624251314
Latitude	41.875272573
date	01/01/2012 05:00:00 AM

(b) Crime Trajectory View

Figure 7: Images from VisCrimePredict. The colours of the dots identify the type of crime

Demonstration scenario 1: A user (e.g., a law enforcement officer) wants to explore the trend of crimes that are happening in a given location. When the user opens the browser, a visualisation interface is initialised with a map view that has events presented as dots (see Figure 7(a)). If the user hovers over the dots shown on the map, a pop-up window reveals more information in relation to the crime event.

5.2 Trajectory Observation

If the user is interested in the trajectory of the crime over a bounded time frame, the system provides a crime trajectory by specifying the PoI and the connective thresholds (both spatial and temporal). Since the size of the sliding window is controlled by the PoI, the observation provided by VisCrimePredict is scalable.

Demonstration Scenario 2: When a user is interested in specific incidents within a period of time, he/she can indicate the PoI with a spatial and temporal threshold (T_s and T_t) which allows the user to obtain different connective relationships based on the threshold used in the query (see Figure 7(b). The user can also click or hover over trajectories to further discover the relationships and the information of the connective crime type, which provides the trajectory after the reconstruction and the segmentation process.

Demonstration Scenario 3: A user wants to see the crimes happening in their neighbourhood in real time as well as historic crimes in the same locality. When the user clicks on the interface, an overlay option is offered that shows real-time relationships with tweets. The blue box shows the trajectory established from tweets and also visualises the historic crime trajectory. (see Figure 8).

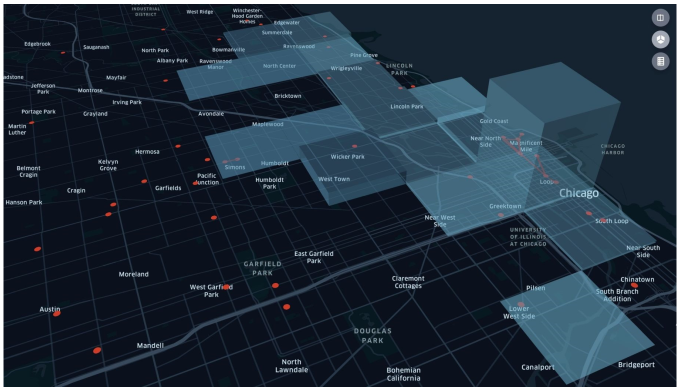


Figure 8: A view of real-time relationship between crime trajectory and tweets from social media data

5.3 Trajectory Prediction

If the user is interested in viewing the prediction trajectory of a category of crime for a given duration and distance, the system provides a crime trajectory by highlighting the next predicted trajectory.

Demonstration Scenario 4: A user can find out about a possible future crime event by providing the required filtering criteria with spatial and temporal thresholds. The trained LSTM model calculates possible future geo-coordinates from the trajectory, based on selected criteria. The next connection in the trajectory is presented to the user with a different colour. The user can hover over the trajectory for further information. This is shown in Figure 9.

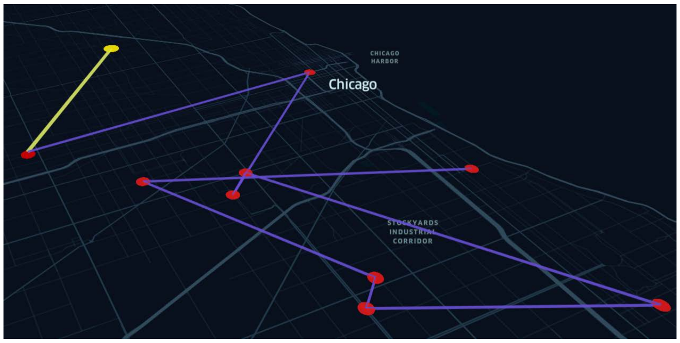


Figure 9: Viewing predicted trajectory (highlighted in yellow) in VisCrimePredict System

6 EXPERIMENTAL RESULTS

The illustrative nature of the visualisation improves when we apply TbSTS over reconstructed trajectories. For example, Figure 10 shows the trajectories of assault crime events for a single day (01-01-2017). Here, the points represent a location, whereas the colour of a point indicates the type of crime. As we can see, trajectories are easier to interpret when we apply the TbSTS algorithm with defined values of T_s and T_t.

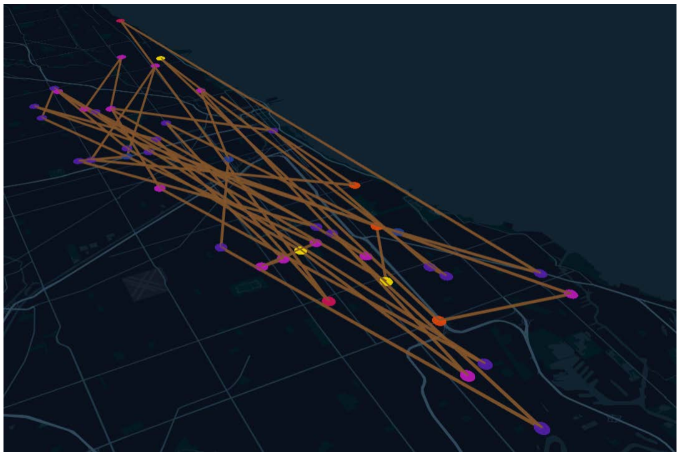


Figure 10: Trajectory of assault crime events for a one day without applying TbSTS

To assess the performance of the LSTM-based learning algorithm and its ability to generate accurate trajectory predictions over different crime categories we first used constant T_s and T_t values to generate trajectories during a whole year. In our case, T_s = 10 km and T_t = 30 minutes. For evaluating the performance of trajectory prediction, we used the top 15 crime categories which have higher numbers of trajectories. We only considered trajectories which have more than one point. 80% data was used for training, the remainder for testing.

We used 20% of the T_s and T_t values as error correction factors. This means that if the predicted value is within the range of the error correction factor (ϵ) of the specified temporal and spatial threshold, we consider it an accurate prediction. For example, for T_s = 10 km and T_t = 30 minutes error correction factor is ϵ_s=2 km and ϵ_t=6 minutes respectively. That is, if $|\widehat{D_{s_{i+1}}} - D_{s_{i+1}}| \leq \epsilon_s$ and $|\widehat{D_{t_{i+1}}} - D_{t_{i+1}}| \leq \epsilon_t$ where $\widehat{D}$ and D are the predicted and original distances, we considered this an accurate prediction.

For each crime category we then computed a normalised root mean square value (RMSE) between the predicted and actual values of D_s and D_t. The observed RMSE value for the top 15 crime categories trajectory model with T_s = 10 km and T_t = 30 minutes is presented in Table 2.

The average RMSE across all crime categories is 0.28. A higher value of RMSE is observed when other filtering criteria are used as input features for LSTM. From this observation we can conclude that the LSTM-based model yields good results in terms of predicting D_s and D_t. We have also observed that categories which have longer trajectories show better performance (low RMSE) than others. The observed results only use the RMSE with a error correction factor to evaluate the prediction performance. Besides providing

Table 2: Performance of Trajectory prediction for different crime categories using 1 year crime data

Crime type	No of trajectories	RMSE
Assault	11873	0.37
Battery	24402	0.25
Burglary	8443	0.29
Criminal Damage	14207	0.34
Theft	29807	0.17
Robbery	5231	0.15
Motor Vehicle Theft	6982	0.20
Narcotics	4301	0.39
Criminal Trespass	2122	0.27
Weapons violation	16520	0.33
Other offence	9667	0.32
Sexual assault	5721	0.19
Public peace violation	2201	0.31
Children offence	1090	0.28
Homicide	972	0.13

improvements to the current model, future work will focus on designing better suited metrics related to a more accurate prediction of trajectories including the use of additional contextual information.

7 CONCLUSION

In this paper we have proposed and demonstrated VisCrimePredict, a visual analytic system that allows visual exploration of crime incidents from multidimensional data obtained from diverse data sources including open data sources and social media. VisCrimePredict provides the capability to present trajectories of crime information propagation at the macro and microscopic levels using a novel threshold-based spatial temporal segmentation algorithm. The algorithm proposed here can generate trajectories from historical and social media data. We also applied the LSTM-based predictive model to the trajectories generated in order to find the next possible path in a crime trajectory, effectively predicting the occurrence of similar crime. Future work will be focused on exploring more microscoping predictive modelling by integrating more community and social media information and also will be added more comparative studies and evaluation based on the well-known machine learning algorithms.

ACKNOWLEDGEMENT

This work was partially supported by Australian Research Council project DP170102231.

REFERENCES

[1] Alan Agresti and Barbara F Agresti. 1970. *Statistical Methods for the Social Sciences.* CA: Dellen Publishers, Prentice Hall, Upper Saddle River, New Jersey, USA.

[2] Martin A Andresen. 2014. *Environmental criminology: Evolution, theory, and practice.* Routledge, London, UK.

[3] M. A. Andresen, A. S. Curman, and S. J. Linning. 2017. The Trajectories of Crime at Places: Understanding the Patterns of Disaggregated Crime Types. *Journal of Quantitative Criminology* 33, 3 (01 Sep 2017), 427–449.

[4] M. Buchin, A. Driemel, M. v. Kreveld, and V. Sacristán. 2011. Segmenting Trajectories: A Framework and Algorithms Using Spatiotemporal Criteria. *Journal of Spatial Information Science* 2011, 3 (2011), 33–63.

[5] Peng Chen, Hongyong Yuan, and Xueming Shu. 2008. Forecasting Crime Using the ARIMA Model. In *Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery - Volume 05 (FSKD '08).* IEEE Computer Society, Washington, DC, USA, 627–630. https://doi.org/10.1109/FSKD.2008.222

[6] Xinyu Chen, Youngwoon Cho, and Suk Young Jang. 2015. Crime prediction using twitter sentiment and weather. In *Systems and Information Engineering Design Symposium (SIEDS), 2015.* IEEE, IEEE, Charlottesville, VA, USA, 63–68.

[7] Matthew S Gerber. 2014. Predicting crime using Twitter and kernel density estimation. *Decision Support Systems* 61 (2014), 115–125.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning.* ACM, ACM, Pittsburgh, Pennsylvania, USA, 369–376.

[9] Alireza Hassani, Pari Delir Haghighi, and Prem Prakash Jayaraman. 2015. Context-aware recruitment scheme for opportunistic mobile crowdsensing. In *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on.* IEEE, 266–273.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Prem Prakash Jayaraman, Abhijat Sinha, Wanita Sherchan, Shonali Krishnaswamy, Arkady Zaslavsky, P Delir Haghighi, Seng Loke, and M Thang Do. [n. d.]. Here-n-now: A framework for context-aware mobile crowdsensing. Citeseer.

[12] Neil Johnson, David Hogg, et al. 1996. Learning the distribution of object trajectories for event recognition. *Image and vision computing* 14, 8 (1996), 609–615.

[13] A. Soares Junior, V. Times, C. Renso, S. Matwin, and L. A. F. Cabral. 2018. A Semi-Supervised Approach for the Semantic Segmentation of Trajectories. In *Proceedings of 2018 19th IEEE International conference on Mobile Data Management.* IEEE, USA.

[14] Gabin Kayumbi, Nadeem Anjum, and Andrea Cavallaro. 2008. Global trajectory reconstruction from distributed visual sensors. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on.* IEEE, IEEE, USA, 1–8.

[15] Keivan Kianmehr and Reda Alhajj. 2006. Crime hot-spots prediction using support vector machine. In *IEEE International Conference on Computer Systems and Applications, 2006.* IEEE, IEEE, Dubai, UAE, UAE, 952–959.

[16] O. Kounadi, A. Ristea, M. Leitner, and C. Langford. 2018. Population at risk: using areal interpolation and Twitter messages to create population models for burglaries and robberies. *Cartography and Geographic Information Science* 45, 3 (2018), 205–220.

[17] M. Krone, J. E Stone, T. Ertl, and K. Schulten. 2012. Fast visualization of gaussian density surfaces for molecular dynamics and particle system trajectories. *EuroVis-Short Papers* 2012 (2012), 67–71.

[18] M. Li, Z. Bao, F. Choudhury, and T. Sellis. 2018. Supporting Large-scale Geographical Visualization in a Multi-granularity Way. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18).* ACM, New York, NY, USA, 767–770.

[19] K. Lin, Z. Xu, M. Qiu, X. Wang, and T. Han. 2016. Noise Filtering Trajectory Compression and Trajectory Segmentation on GPS Data. In *Proceedings of the 2011 International Conference on Computer Science & Education.* IEEE, USA, 490–495.

[20] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations.* 55–60.

[21] A. Morshed, A. R. M. Forkan, T. Shah, P. P. Jayaraman, R. Ranjan, and D. Georgakopoulos. 2018. Visual Analytics Ontology-Guided I-DE System: A Case Study of Head and Neck Cancer in Australia. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC).* 424–429. https://doi.org/10.1109/CIC.2018.00064

[22] A. Morshed, P.-W. Tsai, P. P. Jayaraman, T. Sellis, D. Georgakopoulos, S. Burke, S. Joachim, M. S. Quah, S. Tsvetkov, J. Liew, and C. Jenkins. 2019. VisCrime: A Crime Visualisation System for Crim Trajectory from Multi-dimensional Sources. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining.* ACM, USA, –.

[23] T.Waldo R. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.

[24] D. Weisburd, S. Bushway, C. Lum, and S.-M. Yang. 2004. Trajectories of crime at places: A longitudinal study of street segments in the city of Seattle. *Criminology* 42, 2 (2004), 283–322.

[25] S. Yoo, T. Park, J. Song, and O. Jeong. 2017. A trajectory analysis system for social media contents using AsterixDB. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication.* ACM, ACM, Beppu, Japan, 46.

[26] Y. Zheng. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 29.

[27] Yong Zhuang, Matthew Almeida, Melissa Morabito, and Wei Ding. 2017. Crime Hot Spot Forecasting: A Recurrent Model with Spatial and Temporal Information. In *Big Knowledge (ICBK), 2017 IEEE International Conference on.* IEEE, IEEE, Hefei, China, 143–150.