Hierarchical Spatio-Temporal Pattern Discovery and Predictive Modeling

Chung-Hsien Yu, *Member, IEEE,* Wei Ding, *Senior Member, IEEE,* Melissa Morabito, and Ping Chen

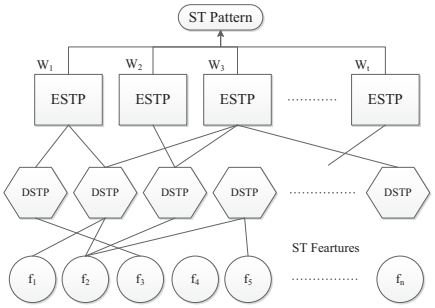
Abstract—We propose a new approach, CCRBoost, to identify the hierarchical structure of spatio-temporal patterns at different resolution levels and subsequently construct a predictive model based on the identified structure. To accomplish this, we first obtain indicators within different spatio-temporal spaces from the raw data. A distributed spatio-temporal pattern (DSTP) is extracted from a distribution, which consists of the locations with similar indicators from the same time period, generated by multi-clustering. Next, we use a greedy searching and pruning algorithm to combine the DSTPs in order to form an ensemble spatio-temporal pattern (ESTP). An ESTP can represent the spatio-temporal pattern of various regularities or a non-stationary pattern. To consider all the possible scenarios of a real-world ST pattern, we then build a model with layers of weighted ESTPs. By evaluating all the indicators of one location, this model can predict whether a target event will occur at this location. In the case study of predicting crime events, our results indicate that the predictive model can achieve 80% accuracy in predicting residential burglary, which is better than other methods.

Index Terms—Spatio-temporal Pattern, Hierarchical Learning, Predictive Model, Crime Forecasting

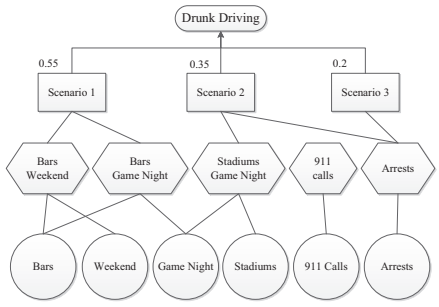
1 Introduction

A spatio-temporal (ST) pattern is regarded as the repeated sequence or association of certain ST events or ST features [1], [2], [3]. To identify these sequences or associations, such as the ST patterns of crime occurrences [4], appropriate distance-based and duration-based measurements are needed to constrain the size or shape of the pattern. Real-world ST patterns can be of different sizes and shapes over time, and non-uniformly distributed over space. This nonstationarity property of ST patterns was recognized by Ratcliffe in the study of crime patterns [5] and has been mentioned in climate studies [6], [7]. In the paper, we propose a new approach, named Cluster-Confidence-Rate-Boosting (CCRBoost). Our approach (1) mitigates the nonstationarity in identifying ST pattern by constituting an ST pattern as a hierarchical structure (see Figure 1) and (2) constructs a predictive model based on this hierarchically learned pattern. Specifically, we identify the local abstracted patterns from distributed representations and then hierarchically learn a global ensemble pattern built upon the identified local patterns [8].

To begin, we gather the indicators from the original data. These indicators are spatio-temporal features because each indicator represents an underlying factor of a spatio-temporal context. For instance, if a pattern of drunk-driving incidents frequently occurs in locations near bars, one indicator that can be used in this pattern is the number of bars



(a) Hierarchical Spatio-Temporal Pattern



(b) Drunk Driving Spatio-Temporal Pattern

Fig. 1. (a) The hierarchical structure of an ST pattern. We propose using Distributed Spatio-Temporal Patterns (DSTPs) to capture the hierarchical structure of an ST pattern and then constructing a predictive model based on this hierarchical structure. (b) An example of the ST pattern is the occurrences of drunk-driving incidents. The number of bars, the day of the week, whether there is a sporting event, 911 calls, and other factors are used as the ST features. The leftmost DSTP represents a local pattern of the drunk-driving incidents occurring near bars during weekends. The leftmost ESTP (Scenario 1) represents a global pattern of drunk-driving incidents near bars during weekend game nights. Combining the different scenarios, a hierarchical ST pattern of drunk-driving incidents can be identified.

- *This manuscript is an extended version of the conference paper, titled "Crime Forecasting Using Spatio-Temporal Pattern with Ensemble Learning", published in Proceedings of Advances in Knowledge Discovery and Data Mining-18th Pacific-Asia Conference (PAKDD), part II, pp. 174-185, Tainan, Taiwan, May 13-16, 2014.*
- *Chung-Hsien Yu, Wei Ding, and Ping Chen are with the Department of Computer Science, University of Massachusetts Boston, Boston, MA, 02125. Dr. Wei Ding is the Corresponding Author.*
E-mail: csyu, ding@cs.umb.edu, Ping.Chen@umb.edu
- *Melissa Morabito is with the School of Criminology and Justice Studies, University of Massachusetts Lowell, Lowell, MA, 01854.*
E-mail: Melissa_Morabito@uml.edu

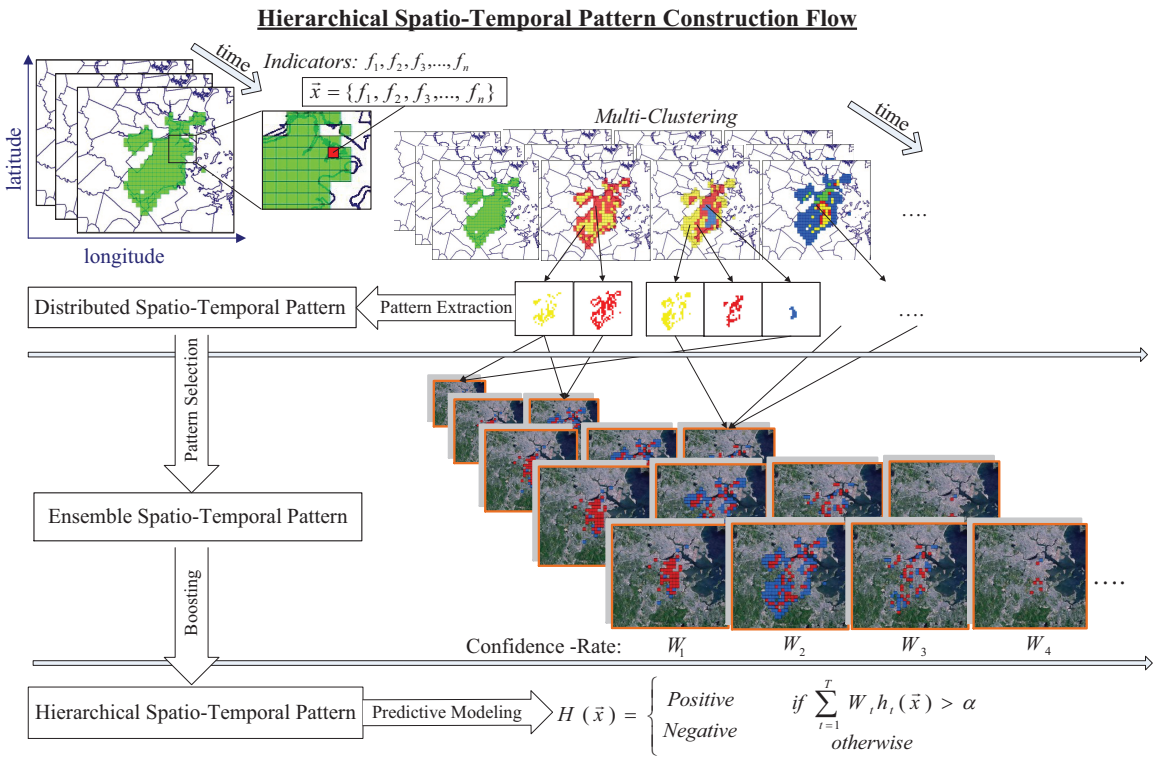


Fig. 2. The flowchart of the proposed CCRBoost approach. It starts with the feature construction stage during which all indicators of different time periods at each location are generated. Using the indicators of one location within the same time period, we construct a feature vector $\vec{x}$. A Distributed Spatio-Temporal Pattern (DSTP) is extracted from the distribution of the locations with similar feature vectors. Next, we select the most suitable DSTPs to build an Ensemble Spatio-Temporal Pattern (ESTP) via a greedy algorithm. Through boosting, we iteratively assign a confidence-rate W to each ESTP as its weight and then combine all the ESTPs into one model. Finally, we construct a strong hypothesis $H(\vec{x})$ based on this hierarchical model which is considered as a global spatio-temporal pattern. $H(\vec{x})$ is used to predict the occurrence of the target events. (Best viewed in color)

near a certain location. We then define the novel concept of distributed spatio-temporal pattern (DSTP). A **DSTP** is embedded in a distribution consisting of the collective locations with the similar indicators within the same time period. Next, by applying the **Negative-Sample-Trimming** theorem introduced in Theorem 2.3, we significantly reduce the computational complexity in DSTP discovery.

After identifying all the DSTPs at different granularity levels and different time periods, we build another hierarchy of ST pattern by combining different DSTPs and define this type of combination as an ensemble spatio-temporal pattern (ESTP). Without applying any chronological or geographical constraints during this hierarchical pattern construction, an ESTP can represent the ST pattern of various regularities or a non-stationary pattern. We formally define DSTP and ESTP in Section 2.1.

Using only one ESTP is insufficient to capture the complexity of a real-world ST pattern. For example, the drunk-driving incidents frequently occur not only in locations near bars on Saturday nights but also in locations near stadiums or arenas during sports seasons. To consider all the possible scenarios of an actual ST pattern, an ensemble learning method is needed to build a model with multiple ESTPs. The goal is to use this model to predict the occurrence of a target event or incident at given locations. In Figure 2, we give a broad view of our hierarchical model to illustrate the proposed approach.

1.1 Challenges and Our Proposed Solutions

In order to build a hierarchical model, our CCRBoost approach addresses three difficult challenges: (1) To identify all the DSTP candidates at various granularity levels and different time periods; (2) To select the most suitable DSTPs as the combination for constructing an ESTP; (3) To formulate the correlations of multiple ESTPs fitting into a predictive model.

First, we design a multi-level clustering method to identify the local distributions at different granularity levels by varying the number of clusters. These distributions are not-mutually-exclusive sub-partitions from which the features can be learned more efficiently [8]. Thus, our DSTP discovery is embedded with a feature selection process which chooses the most delegated indicators to represent an underlying ST pattern. We then apply this method to the chronologically dissected datasets in order to identify the DSTPs at different timespan.

Not all DSTPs can be used to form the ESTPs. The DSTPs learned at a local level could be redundant or overlapping at a global level, or even irrelevant. Moreover, a real-world ST pattern is a complex phenomenon which is difficult to capture within a single ESTP. To overcome the remaining two

problems, we adopt a boosting approach, which embeds with a greedy search algorithm to effectively select the most representative DSTPs from the entire pool of DSTPs to form an ESTP, to construct the predictive model by one layer of ESTP at a time. In our design, each layer of ESTP is assigned a confidence factor as its weight which is also the correlation in this predictive model. By examining all the indicators of one location, this model forecasts the occurrence of a target event at this location. The theoretical analysis of this boosting algorithm for hierarchical learning is illustrated in Section 2.2 and the detailed discussion of the proposed CCRBoost approach is given in Section 3.

1.2 Crime Pattern Discovery

In this research, we apply our proposed approach to crime pattern identification and forecasting for three main reasons. First, scientists are still longing to craft ways to foresee future crime by studying crime patterns [4], [9], [10], [11], [12], [13], [14]. Secondly, from the early findings of these studies, evidence suggests that crime patterns are not only closely associated with societal conditions but also tightly correlated with spatial-temporal factors. Since our approach aims to produce a predictive model using the spatial-temporal patterns, forecasting crime is a very suitable application. Finally, we have collaborated with a police department in a large Northeastern city in the US to obtain 4-years of historical crime data, from January 1st, 2006 to December 31st, 2009. These data are used to evaluate our approach. Our experimental results show that the proposed predictive model has achieved 80% on accuracy in predicting residential burglary. The results of our empirical evaluations are provided in Section 4.

The proposed approach is a general way of hierarchically identifying the ST pattern for predictive modeling, and is not limited to forecasting crime. For example, it could be modified and deployed to identify the formation patterns of severe weather phenomena, such as floods, droughts, or earthquakes. With our predictive model, people could be given early warnings so that the damage caused by these natural disasters might be mitigated. The conclusions and future works of our research are presented in Section 6.

1.3 Contribution

In summary, our contributions are:

- We introduce the novel concept of underlying ST pattern, named **Distributed Spatio-Temporal Pattern (DSTP)**. DSTPs are learned at different spatial scales and different temporal spaces. We then use the combinations of the DSTPs to represent the more complex ST pattern, named **Ensemble Spatio-Temporal Pattern (ESTP)**, as the second hierarchy. Finally, to consider different scenarios of the occurrences of an ST event, we incorporate different ESTPs to construct a hierarchical model.
- We propose the **Cluster-Confidence-Rate-Boosting (CCRBoost)** approach which starts with multi-clustering followed by local feature learning processes to discover all possible DSTPs from distributions of different shapes, sizes, and time periods. Through these processes, we extract the most suitable indicators for each DSTP to represent the underlying factors of a pattern. Next, we adopt a gradient descent boosting approach embedded with a greedy search algorithm to perform pattern selection in forming ESTPs hierarchically and then build a layered predictive model at the same time.
- We introduce **Negative-Sample-Trimming**, which is implemented in our pattern selection to significantly reduce the computational complexity in finding the best DSTP combination to form an ESTP. We also provide the theoretical analysis and proof of this theorem.
- Using the real-world crime data, we have applied our approach to predicting the occurrence of residential burglary incidences. The results show that our proposed predictive model is able to obtain 80% accuracy. Meanwhile, through the visualization of the chosen DSTPs used in the final predictive model, we verify that our approach does identify the existing crime patterns and provides better interpretation of the characteristics of the crime.

TABLE 1
Mathematical Notations

f_i	An indicator
$\vec{x}$	The feature vector of an instance
y	The class label of an instance
X	The training dataset
K	The clustering levels
M	The total number of time periods
T	The learning layers
r	The rule-based classifier of a DSTP
D_t	The weight distribution of X at layer t
W_+	The total weight of the true positive instances
W_-	The total weight of the false positive instances
R_t	The ESTP constructed at layer t
$\hat{C}_R$	The confidence value of R. (Eq. 15)
Z	The normalization factor (Eq. 7)
$\hat{Z}$	The factor of the minimum Z (Eq. 16)
$h_t(\vec{x})$	The hypothesis of R_t
$H(\vec{x})$	The final strong hypothesis
α	The user-defined threshold

2 Problem Formulation and Definitions

Shown in Figure 3, one data instance is generated from a spatio-temporal cube ranged in a location (grid cell) and a timespan. A feature f_i of an instance represents one characteristic showing in this instance's spatio-temporal context. f_i can be a real number, a boolean, or an ordinal value. Let $\vec{x} = \{f_1, f_2, \ldots, f_n\}$ be the feature vector and $y = \{1, -1\}$ be the class label of an instance. A positive class indicates the problem of interest, e.g. a crime hotspot. A spatio-temporal dataset X is the collection of the instances obtained from the studied locations and time periods. We formulate the problem of spatio-temporal predictive modeling as follows.

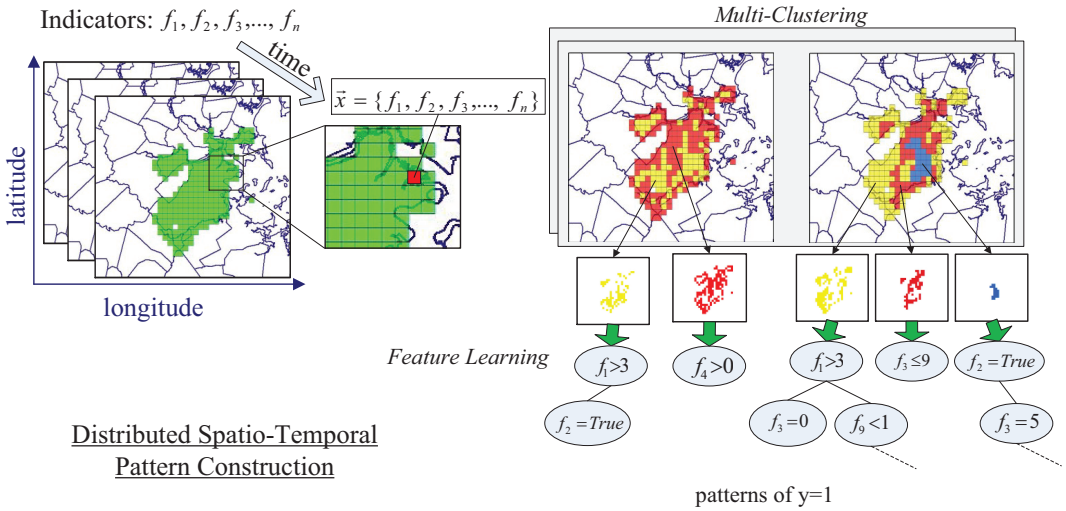


Fig. 3. An illustration of the DSTP construction. $\vec{x}$ is a feature vector of an instance of one location (grid cell) consisting of the indicators $f_1, f_2, \ldots, f_n$. Each instance has a class label y as the ground truth. Through the multi-clustering process, the instances with similar feature vectors in the same time period are segmented into non-mutually-excursive sub-partitions. By projecting a sub-partition on the map, we can obtain a spatial distribution (the grid cells filled with the same color). We learn a rule-based classifier from this type of spatial distribution to represent a Distributed Spatio-Temporal Pattern. (Best viewed in color)

- **Given:** a spatio-temporal dataset X; the training and test instances drawn from X.
- **Find:** a classification model based on the training instances.
- **Objective:** minimize classification errors generated by evaluating the model on the test instances.

Our goal is to construct a predictive model, denoted as H, which takes an instance's feature vector $\vec{x}$ as its input and then outputs $H(\vec{x})$ as this instance's predicted class label, as shown in Figure 2. When $H(\vec{x}) = y$, we consider the model to correctly predict the class label of this instance. Therefore, we model the problem as a classification problem and validate the model using the classification accuracy.

2.1 The Ensemble Distributed Spatio-Temporal Pattern

We group the instances with similar feature vectors at the same time period into the same cluster. We assume that there exists a pattern within each cluster. We learn a rule-based binary classifier from this type of cluster to represent an underlying local ST pattern of the class label $y = 1$. We call this type of local spatio-temporal pattern a **Distributed Spatio-Temporal Pattern (DSTP)** because this pattern exists in the spatial distribution consisting of the locations with similar characteristics. Since the DSTP is a rule-based pattern, it is discriminative and interpretable by design. For example, in Figure 3, if f_1 is the number of bars, f_2 indicates the nighttime hours, and $y = 1$ means the occurrence of the drunk-driving incident, then the leftmost DSTP represents the drunk-driving pattern occurring at the locations with more than three bars during the nighttime hours. Therefore, only the most appropriate features are involved in a DSTP, not all of them. We then formally define a DSTP as follows.

Definition 2.1. ***Distributed Spatio-Temporal Pattern (DSTP):*** *Let a group of instances with similar feature vectors be denoted as C. A DSTP, denoted as r, is a rule-based binary classifier learned from C. When a DSTP r is used to classify a feature vector $\vec{x}$, we denote $r(\vec{x}) = 1$ if $\vec{x}$ is classified as a positive class and $r(\vec{x}) = -1$ if $\vec{x}$ is classified as a negative class.*

DSTPs can be learned from the sub-partitions at different similarity degrees to capture the local ST patterns of different granularity levels. We further proposed the **Ensemble Spatio-Temporal Pattern (ESTP)** to represent the global ST patterns with various regularities. An ESTP is a combination of multiple DSTPs, which represents a scenario under certain circumstances. For example, in Figure 1, the leftmost ESTP (Scenario 1) represents a global pattern of drunk-driving incidents near bars during game nights on the weekends. By considering DSTP's classification result as Boolean variable, we define an ESTP as the conjunction of the DSTPs. Formally, an ESTP is defined as follows.

Definition 2.2. ***Ensemble Spatio-Temporal Pattern (ESTP):*** *An ESTP R is the conjunction of a group of DSTPs, $r_1, r_2, \ldots, r_n$, with the following form:*

$$R(\vec{x}) = \{r_1(\vec{x}) \wedge r_2(\vec{x}) \wedge \cdots \wedge r_n(\vec{x})\}. \quad (1)$$

$R(\vec{x}) = 1$, if and only if $r_1(\vec{x}) = 1$ and $r_2(\vec{x}) = 1$ and ... and $r_n(\vec{x}) = 1$. Otherwise, $R(\vec{x}) = -1$.

2.2 Theoretical Background

To construct ESTPs and then collectively utilize the identified ESTPs, we incorporate a hierarchical learning process based on the boosting algorithm. This process iteratively chooses the best ESTP, denoted as R_t, at each layer t and builds a predictive model with T layers of ESTPs at the end of T iterations. From each R_t, we learn a hypothesis h_t where $h_t(x) \in \mathbb{R}$, x is a feature vector of an instance, and $\mathbb{R}$ is the domain of real numbers. The sign of $h_t(\vec{x})$ is

regarded as the predicted label of the input instance. Thus, if $h_t(\vec{x}) > 0$ then the predicted class label is 1; if $h_t(\vec{x}) < 0$ then the predicted class label is -1. We then use $|h_t(\vec{x})|$ as the prediction confidence. Let a training instance's weight at iteration t be $D_t(i)$, feature vector be $\vec{x_i}$, and class label be y_i. The objective is to find the best h_t which yields the least prediction error on training instances. In particular, we find the best combination of DSTPs to construct an ESTP which gives the best h_t.

We normalize the weight distribution of the training instance at each round t so that $\sum_i D_t(i) = 1$ which is the probability distribution in the training space. We then define the expected confidence value m_t of hypothesis h_t as:

$$m_t = E_{i\sim D_t}[y_i h_t(\vec{x_i})] = \sum_i D_t(i) y_i h_t(\vec{x_i}). \quad (2)$$

When $h_t(\vec{x_i})$ has the correct prediction (a true positive or a true negative), $y_i h_t(\vec{x_i}) > 0$. Otherwise, $y_i h_t(\vec{x_i}) < 0$. Therefore, m_t measures the overall prediction performance of h_t in the training space. The higher the m_t the better the h_t. This hierarchical learning approach essentially uses the same principle of confidence-rate boosting approach [15], [16], which allows indecisive prediction as zero confidence ($h_t(\vec{x_i}) = 0$). According to Schapire and Singer [15], when h_t has the range of $\{-1, +1\}$, the training error of the final strong hypothesis H has the upper boundary of $\prod_t \sqrt{1-(m_t)^2}$. Schapire and Singer also prove that

$$Z_t \leq \sqrt{1-(m_t)^2}. \quad (3)$$

Thus, the upper bound of the training error can be replaced by $\prod_t Z_t$. Z_t is the normalization factor used to reweigh the instances for next round $t+1$ and is defined as:

$$Z_t = \sum_i D_t(i) \mathrm{e}^{-\beta_t y_i h_t(\vec{x_i})}, \quad (4)$$

where $\beta_t \in \mathbb{R}$, a weight given to h_t. Minimizing Z_t at each round t leads to a lower error upper bound, which also suggests the smallest training error. Let $C_R = \beta_t h_t(\vec{x_i})$ and ignore the round t; Z is our loss function:

$$Z = \sum_i D(i) \mathrm{e}^{-C_R y_i}. \quad (5)$$

We intend to find the C_R value that produces the smallest Z, such that we have the minimum training error. We let $C_R = 0$ when $R(\vec{x_i}) \neq 1$. Here, $R(\vec{x_i}) \neq 1$ means that $\vec{x_i}$ is not classified as a positive instance by an ESTP R (Definition 2.2). Therefore, C_R is a real-valued confidence for the positive prediction generated by $h_R()$ which is the hypothesis learned from R using training instances. By separating the training instances into two groups, $R(\vec{x_i}) = 1$ and $R(\vec{x_i}) \neq 1$, from Equation (5), we obtain

$$Z = \sum_{i|R(\vec{x_i})\neq 1} D(i) + \sum_{i|R(\vec{x_i})=1} D(i) \mathrm{e}^{-C_R y_i} \quad (6)$$

since $C_R = 0$ when $R(\vec{x_i}) \neq 1$. We have rewritten Equation (6) as

$$Z = W_0 + W_+ \mathrm{e}^{-C_R} + W_- \mathrm{e}^{C_R} \quad (7)$$

by dividing and then summarizing the weights of the instances into three groups, W_0, W_+, and W_-. $W_0 = \sum_{i|R(\vec{x_i})\neq 1} D(i)$, so W_0 is the total weight of the instances predicted as true negatives or false negatives.

$$W_+ = \sum_{i|R(\vec{x_i})=1 \; and \; y_i=1} D(i), \quad (8)$$

$$W_- = \sum_{i|R(\vec{x_i})=1 \; and \; y_i=-1} D(i). \quad (9)$$

W_+ is the total weight of the instances correctly predicted as true positives, and W_- is the total weight of the instances wrongly predicted, otherwise known as false positives.

With Equation (7), we want to find the value of C_R which minimizes Z. This can be done by taking the first derivative of Z with respect to C_R and let $\frac{dZ}{dC_R} = 0$, which is

$$\begin{aligned}
\frac{dZ}{dC_R} &= -W_+ \mathrm{e}^{-C_R} + W_- \mathrm{e}^{C_R} = 0 \\
&\Longrightarrow W_- \mathrm{e}^{C_R} = W_+ \mathrm{e}^{-C_R} \\
&\Longrightarrow \ln(W_- \mathrm{e}^{C_R}) = \ln(W_+ \mathrm{e}^{-C_R}) \\
&\Longrightarrow \ln(W_-) + C_R = \ln(W_+) - C_R \\
&\Longrightarrow 2C_R = \ln(W_+) - \ln(W_-) \\
&\Longrightarrow C_R = \frac{1}{2}\ln(\frac{W_+}{W_-}). \quad (10)
\end{aligned}$$

Next, we take the second derivative of Z, which is

$$\frac{dZ}{dC_R^2} = W_+ \mathrm{e}^{-C_R} + W_- \mathrm{e}^{C_R} > 0.$$

Since the second derivative of Z is greater than zero,

$$min(Z) = W_0 + 2\sqrt{W_+ W_-}, \quad (11)$$

when

$$C_R = \frac{1}{2}\ln(\frac{W_+}{W_-}). \quad (12)$$

Because the weight distribution D is a probability distribution, $W_0 + W_+ + W_- = 1$. As a result, the minimum value of Z can be rewritten as:

$$\begin{aligned}
min(Z) &= 1 - (W_+ - 2\sqrt{W_+ W_-} + W_-) \\
&= 1 - (\sqrt{W_+} - \sqrt{W_-})^2. \quad (13)
\end{aligned}$$

Also, from Equation (7), we obtain

$$Z = (1 - W_+ - W_-) + W_+ \mathrm{e}^{-C_R} + W_- \mathrm{e}^{C_R}. \quad (14)$$

Therefore, W_0 is eliminated from the equation to calculate Z. To prevent the division by zero, C_R is adjusted as:

$$\hat{C}_R = \frac{1}{2}\ln(\frac{W_+ + \frac{1}{2v}}{W_- + \frac{1}{2v}}), \quad (15)$$

where v is the total number of instances.

2.3 Ensemble Spatio-Temporal Pattern Construction

Based on the theoretical analysis given in Section 2.2, we further propose $BuildChain()$ function to construct the best ESTP R_t which leads to the smallest classification error at each layer t. In other words, R_t gives the minimum Z_t value described in Equation (13). With Equation (15), we can calculate the confidence $\hat{C}_{R_t}$ of R_t which is used as the key measurement to find the best combinations of the DSTPs. Furthermore, we propose $PruneChain()$ function to reevaluate the ESTP in order to prevent the hypothesis h_t from overfitting.

Algorithm 1 BuildChain(**GrowSet**, l)

Input:
GrowSet: The training dataset.
l**:** The collection of all DSTPs.
Output:
R**:** The LIFO queue initialized as an empty queue.
$\tilde{Z}_{max} = -\infty$.
repeat
  Find a DSTP r from l which maximizes $\tilde{Z}$. $\tilde{Z}$ is calculated using **GrowSet** and Equation (16).
  if $\tilde{Z} > \tilde{Z}_{max}$ **then**
    Set $\tilde{Z}_{max} = \tilde{Z}$.
    Push r into R.
  end if
until ($\tilde{Z}_{max} \leq \tilde{Z}$)
Return: R

Algorithm 2 BuildChain(**GrowSet**, l) with Negative-Sample-Trimming

Input:
GrowSet: The training dataset.
l**:** The collection of all DSTPs.
Output:
R**:** The LIFO queue starting with an empty queue.
G= A copy of **GrowSet**.
$\tilde{Z}_{max} = -\infty$.
repeat
  Find a DSTP r from l which maximizes $\tilde{Z}$. $\tilde{Z}$ is calculated using G and Equation (16).
  if $\tilde{Z} > \tilde{Z}_{max}$ **then**
    Remove the instances with the feature vectors $r(\vec{x}) \neq 1$ from G.
    Set $\tilde{Z}_{max} = \tilde{Z}$.
    Push r into R.
  end if
until ($\vec{x} \in G \mid \forall r(\vec{x}) = 1$) or ($\tilde{Z}_{max} \leq \tilde{Z}$)
Return: R

2.3.1 The BuildChain Function

To begin, we divide the training dataset into two subsets, **GrowSet** and **PruneSet**. We then define $\tilde{Z}$ as follows:

$$\tilde{Z} = \sqrt{W_+} - \sqrt{W_-} \tag{16}$$

By Equation (13), having the maximum $\tilde{Z}$ value also gives the minimum Z value. Thus, we design $BuildChain()$ as a greedy algorithm which keeps finding a DSTP r from the pool of DSTPs to add into R until this R gives the maximum $\tilde{Z}$ value using **GrowSet** and Equation (16), shown in Algorithm 1. In our setting, $\tilde{Z} > 0$ is preferred because predicting more true positive samples is favorable.
However, the search function becomes very computationally expensive when the number of DSTPs included in R keeps growing because evaluating R is equal to evaluating each $r_i \in R$ using **GrowSet**. To increase the efficiency of calculating $\tilde{Z}$ values in this function, we give a new theorem, **Negative-Sample-Trimming**. According to Equation 7, the weight of an instance which is predicted as a negative instance by a pattern r_i included in R is accumulated into W_0. Since W_0 is not involved in Equation 16, the weights of the instances predicted as negative classes have no impact in the calculation of $\tilde{Z}$. Originally, the $\tilde{Z}$ value is calculated by evaluating $R \wedge r$ using **GrowSet**. With **Negative-Sample-Trimming**, we remove the instances which are predicted as negative classes by R from **GrowSet** to obtain a trimmed dataset, G, before adding r into R. The same $\tilde{Z}$ value is obtained by evaluating r using G. This theorem is formally defined and proved as follows.

Theorem 2.3. (**Negative-Sample-Trimming**) *Let $\bar{G}$ be a subset of dataset S, $\bar{G} \subseteq S$, where $\bar{G} = \{\vec{x_j} \in \bar{G} \mid \forall R(\vec{x_j}) \neq 1\}$ and R is an ESTP. Let G be a subset of S, where $G = S - \bar{G}$. $\tilde{Z}(r, G)$ is a function which calculates the $\tilde{Z}$ value using Equation (16) with r and G as its inputs. Then $\tilde{Z}(r, G) = \tilde{Z}((R \wedge r), S)$, where r is a DSTP.*

Proof. By the definitions in Equation (8),

$$W_{S+} = \sum_{i|R(\vec{x_i})=1 \text{ and } r(\vec{x_i})=1 \text{ and } y_i=1} S(i),$$

where $S(i)$ is the weight of $\vec{x_i}$ and $S()$ is the weight distribution of S. Let

$$\bar{G} = \{\vec{x_j} \in \bar{G} \mid \forall R(\vec{x_j}) \neq 1\}$$
$$G = S - \bar{G}.$$

Since $R(\vec{x_i}) = 1$ and $r(\vec{x_i}) = 1$, we know that $\vec{x_i} \notin \bar{G}$ so

$$W_{S+} = \sum_{i|R(\vec{x_i})=1 \text{ and } r(\vec{x_i})=1 \text{ and } y_i=1} \textbf{G(i)},$$

where $G()$ is the weight distribution of G.
Besides, every $\vec{x_i}$ in G must satisfy $R(\vec{x_i}) = 1$. Therefore,

$$W_{S+} = \sum_{i|r(\vec{x_i})=1 \text{ and } y_i=1} G(i)$$
$$= W_{G+}$$

Following the same steps and Equation (9), we can also prove that $W_{S-} = W_{G-}$.
From Equation (16), we have

$$\tilde{Z}(r, G) = \sqrt{W_{G+}} - \sqrt{W_{G-}}$$
$$\tilde{Z}((R \wedge r), S) = \sqrt{W_{S+}} - \sqrt{W_{S-}}$$

Thus, we conclude that

$$\tilde{Z}(r, G) = \tilde{Z}((R \wedge r), S)$$

□

The improved $BuildChain()$ function, shown in Algorithm 2, uses a dataset G, duplicated from **GrowSet** initially, to evaluate one single r instead of R. Repeatedly, this function finds the best r to join R and removes the instances predicted as negative classes by r from G until there does not exist any r which can increase the $\tilde{Z}$ values, or until every instance in G is predicted as positive classes by r. At the end, this function returns an ESTP R which gives the minimum classification error.

Algorithm 3 PruneChain(**GrowSet**, **PruneSet**, R)

Input:
GrowSet: The training dataset.
PruneSet: The validation dataset.
R: The output from the $BuildChain()$.
Output:
R: The final ESTP.
$Stop = False$.
Calculate $\hat{C}_R$ using Equation (15) and **GrowSet**.
Calculate Z using Equation (14) and **PruneSet**.
Set $Z_{min} = Z$.
repeat
  Pop the top r from the queue of R.
  Calculate $\hat{C}_R$ using Equation (15) and **GrowSet**.
  Calculate Z using Equation (14) and **PruneSet**.
  if $Z < Z_{min}$ **then**
    Set $Z_{min} = Z$.
  else
    Push r back to R
    Set $Stop = True$
  end if
until $(Stop = True)$ or (only one pattern left in R)
Return: R

2.3.2 The PruneChain Function

In $PruneChain()$ function, shown in Algorithm 3, we reevaluate the ESTP R returned by $BuildChain()$ using Equation (14) and **PruneSet** to solve a possible overfitting problem. The Z in Equation (14) can be used to represent the error rate when evaluating R using **PruneSet**. We first calculate the $\hat{C}_R$ value of R using **GrowSet** and Equation (15). We then calculate the Z value using **PruneSet** and $\hat{C}_R$ as the C_R in Equation (14). This function repeatedly removes an r from R until the minimum value of Z is reached or there is only one pattern left in R. At the end, the R with the minimum error rate is returned.

3 The CCRBoost Approach

Our CCRBoost approach implements the theoretical framework discussion in Section 2. It performs (1) DSTP discovery: To identify all the DSTP candidates at various granularity levels and different time periods; (2) Pattern selection: To select the most suitable DSTPs to be assembled for constructing an ESTP; (3) Predictive modeling: To incorporate the identified ESTPs into one predictive model. The steps of the proposed CCRBoost approach are given in Algorithm 4.

3.1 DSTP Discovery

We first divide the data into M subsets by certain length of time interval. For example, if we divide the time series data using a window of one month, then there are 12 subsets ($M = 12$) when one year worth of data is involved. Next, we identify the spatial distributions consisting of the locations with similar indicators. We also consider the spatial distributions under various resolutions. Then, we craft an unsupervised multi-level clustering to find all the possible distributions. K-Means is chosen to identify these distributions. However, our approach is not limited to using

Algorithm 4 CCRBoost(X, K, M, cls)

Input:
X: The set of training instances.
K: The clustering level.
M: The total number of time periods.
cls: The base classifier used to extract a DSTP.
Output:
H: The strong hypothesis of the final model.
for $k = 1 \dots K$ **do**
  for $m = 1 \dots M$ **do**
    Run K-Means using the instances in period m to generate k clusters from which k DSTPs are extracted by cls and stored in l.
  end for
end for
Balance the weights of the dataset.
for $t = 1 \dots T$ **do**
  Normalize the weights, let D_t be a probability distribution.
  Divide data into two sets, **GrowSet** and **PruneSet**.
  R_t = $BuildChain$(**GrowSet**, l)
  R_t = $PruneChain$(**GrowSet**, **PruneSet**, R_t)
  Calculate $\hat{C}_{R_t}$ using entire dataset and Equation (15).
  Update D_t based on Equation (17).
end for
The strong hypothesis of the final global predictive model is defined as:
$H(\vec{x}) = \begin{cases} Positive & \sum_{t=1}^{T} \hat{C}_{R_t} h_t(\vec{x}) > \alpha \\ Negative & otherwise \end{cases}$
where $h_t(\vec{x}) = 1$ when $R_t(\vec{x}) = 1$,
$h_t(\vec{x}) = 0$ when $R_t(\vec{x}) \neq 1$,
and α is a user-defined threshold.

K-Means for multi-level clustering. We perform multi-level clustering, by running K-Means from 1 to K clusters, to obtain $1 + 2 + \cdots + K$ distributions. As a result, there are total $M \times (1+2+\cdots+K)$ distributions acquired from these M subsets.

From each identified distribution, we extract a DSTP through decision-tree learning which generates a rule-based classifier and provides an interpretable representation of a pattern as well as feature selection. Thus, each DSTP has different indicators as its features to represent a pattern at the local level and the underlying factors of the hierarchical ST patterns. In our approach, we intend to embed the spatio-temporal dimension into this pattern discovery so we are able to identify all the possible DSTPs at different granularity levels and different time periods.

3.2 Pattern Selection

The two remaining tasks are how to select the most suitable DSTPs to form the ESTPs and how to build an effective predictive model based on multiple ESTPs. We achieve these two tasks by adopting the hierarchical learning model discussed in Section 2.2. We first balance the training dataset X by making the total weight of positive instances equal to the total weight of negative instances. This ensures that every instance of every location is included in our learning process and avoids possible biased results caused by the imbalanced training data [17]. The weight plays a key role

in our hierarchical learning because the evaluation measurements, such as Z and C_R, in our model are calculated based on the weights of the training instances. Next, the weights of the entire training dataset are set to be in a probability distribution which makes the total weight equals to 1. We then use the $BuildChain()$ followed by $PruneChain()$, described in Section 2.3.1 and Section 2.3.2, to find the best ESTP candidate as another hierarchy of our model.

3.3 Predictive modeling

To find the different scenarios of a hierarchical spatio-temporal pattern, we repeat the pattern selection for a given number of iterations T, where T is a user-defined variable. At each round t, the ESTP returned by $PruneChain()$ is then added to the hierarchical model as the layer t along with its confidence-rate $\hat{C}_{R_t}$ which is calculated by Equation (15) using R_t and the entire training dataset. To give the data instances which are not recognized by R_t more attention in the next iteration, we exponentially lower the weights on those instances which are classified by R_t as positive classes. This weight update is based on $\hat{C}_R$ and we define the update function as follows:

$$D_{t+1}(i) = \frac{D_t(i)}{e^{y_i \hat{C}_{R_t}}},\ if\ R_t(\vec{x_i}) = 1. \tag{17}$$

At the end, T ESTPs, $R_1, R_2, \ldots, R_T$, and T confidence-rates, $\hat{C}_{R_1}, \hat{C}_{R_2}, \ldots, \hat{C}_{R_T}$ are produced. The strong hypothesis of the final model is defined as:

$$H(\vec{x}) = \begin{cases} Positive & \sum_{t=1}^{T} \hat{C}_{R_t} h_t(\vec{x}) > \alpha \\ Negative & otherwise \end{cases}, \tag{18}$$

where $h_t(\vec{x}) = 1$ when $R_t(\vec{x}) = 1$, $h_t(\vec{x}) = 0$ when $R_t(\vec{x}) \neq 1$, and α is the user-defined threshold.

By taking an input $\vec{x}$, this strong hypothesis evaluates $\vec{x}$ over each ESTP R_t. If x is classified by R_t as a positive class, then $\hat{C}_{R_t}$ is added to the total confidence score. $\vec{x}$ is predicted as a positive class if the total confidence score is greater than the threshold α after the evaluation. Otherwise, $\vec{x}$ is predicted as a negative class. The threshold α is usually set to zero.

4 Case Study: Forecasting Residential Burglary

4.1 Crime Data

Residential Burglary, defined as an illegal entry to a dwelling to commit a felony [18], is of particular interest to study from a machine learning prediction perspective since the near repeat hypothesis suggests that proximity to a burgled residence increases the likelihood of victimization of other domiciles in the neighborhood [19].

Collaborating with the police department of a Northeastern city in the United States[1], we obtained 4-years of residential burglary report from January 1st, 2006 to December 31st, 2009. Six categories of related events were identified as having the highest correlation with residential burglary. Selection of those crime explanatory variables is also in-line with the criminology literature [19], [20], [21], [22], [23]. These six categories, used as crime indicators in our case study, are Overall Arrest, Residential Burglary, Commercial Burglary, Motor Vehicle Larceny, Street Robbery, and Foreclosure. Clearly, these indicators are related, as evidence suggests that offenders do not specialize in their criminal activities meaning that they will engage in a diversity of criminal opportunities depending upon opportunities and personal circumstances [24]. Arrest indicates that there is someone taken into custody for a crime. Residential Burglary is the reported record of illegal intrusion into a private or personal property with the intent of taking another's possession whereas Commercial Burglary indicates the record of intrusion happening in a commercial or business buildings. Vehicle Theft is categorized as Motor Vehicle Larceny. These are all property crimes and do not involve violence against individuals. Street Robbery, however, is a different crime that involves a victim who is threatened with force to give this person's possession to the offender. While not indicative of specific crime, an indicator of Foreclosures is included in order to consider the economic factors associated with criminal activity. The total number of records of each crime category, including population, size, income, and households used in our study, are listed in Table 2.

TABLE 2
Crime data sheet.

Crime Incidents (Jan 1st, 2006 ∼ Dec 31st, 2009)

Type	Incidents
Arrest	254,309
Commercial Burglary	3,059
Foreclosed Houses	11,671
Motor-Vehicle Larceny	29,633
Residential Burglary	11,056
Street Robbery	8,217

Societal Factors

Population	630,000
Size	232 km^2
Median household income	$53,136
Households	248,704

Data Grid Resolutions

Size	Blocks	Hotspots	Coldspots
800m	235	3,540	4,920
600m	377	4,466	9,106
450m	619	5,441	16,843

The data instances of our case study are constructed by geographically dividing the city into a chessboard-like grid cells/blocks. Each block is considered one location. The counts of same type events are aggregated by month and by location. In our case study, the ground truth y of each instance is labeled as a crime hotspot (positive class) if at least one Residential Burglary occurred at the location of $\vec{x}$ in the next month. Otherwise, it is labeled as a coldspot (negative class). Such a setting predicts emerging residential burglaries one month in advance by evaluating crime indicators in the current month. To find the optimal spatial resolution, three different grid resolutions have been applied to generate three data sets from the original crime records. These three resolutions have the square cell/block with edge lengths of 800, 600, and 450 meters, respectively. The finer resolution with the cell size smaller than 450 meters is not used in our study because there are too

1. Due to an agreement with this police department, we cannot disclose the name of our study city in this paper.

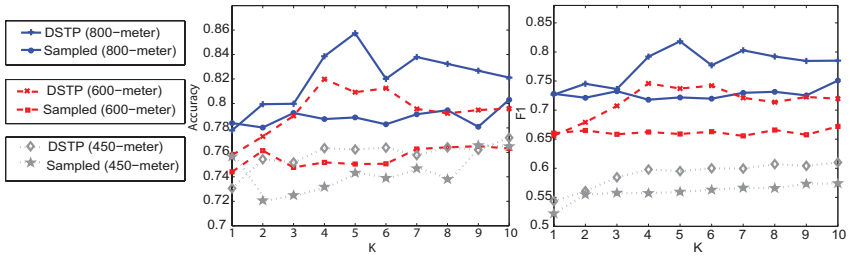


Fig. 4. The prediction results of the model using DSTPs and the model using randomly sampled patterns on three datasets of different resolutions.

many crime indicators with zero values. Three years' crime records from January 1st, 2006 to December 31st, 2008 are used as training data, and one year's crime records from January 1st, 2009 to December 31st, 2009 are used as test data. To extract DSTPs, LADTree [25] is chosen as the base classifier for ensemble learning due to its good performance on rule-based classifiers and weighting system similar to our proposed approach. In Table 3, we show the experimental results of using different base classifiers in our model. Our approach is not limited to using LADTree as the base classifier because an ST pattern can be represented in any proper model as long as the model can classify whether an instance is a hotspot or coldspot. We will discuss the performance measurements for the classification models in Section 4.2. In our experiments, α is always set to be zero. However, we vary the α and study its impact in Section 4.8.

TABLE 3
The results of CCRBoost using different base classifiers, where $T = 500$ and $K = 5$.

Base Classifier	*Accuracy*	*F1*
OneR	0.774	0.713
C4.5	0.771	0.715
NaiveBayes	0.782	0.691
SVM	0.827	0.771
LADTree	**0.857**	**0.818**

4.2 Performance Measurements

To evaluate the prediction performance of the proposed approach as well as compare it with the state-of-the-art approaches, we use Accuracy, Precision, Recall, and F1-score, as our prediction measurements [26]. Accuracy measures the correctness of the overall prediction while Precision shows the correctness of the hotspot prediction. What fraction of the actual hotspots is predicted is measured by Recall. F1-score is the harmonic mean of Recall and Precision.

4.3 Comparison with Randomly Sampled Patterns, the Baseline Approach

The baseline approach in our comparative study is designed as follows. In this experiment, the variable K used as the level of clustering in our DSTP discovery is also used to decide the number of bags for random sampling. We randomly select 50% of the instances from one monthly dataset for $\frac{K(1+K)}{2}$ times without replacement, which means that there are no duplicated instances within each bag. This method constructs $\frac{M \times K(1+K)}{2}$ subsets of the same size from M monthly datasets. Then, $\frac{M \times K(1+K)}{2}$ patterns are learned from these random datasets using the LADTree classifiers. As a result, there are the same number of local patterns generated by our multi-level clustering and by random sampling. Next, we use the hierarchical learning approach with the $BuildChain()$ and $PruneChain()$ functions to construct ESTPs from those randomly sampled patterns to build a predictive model. As illustrated in Figure 4, our method consistently performs better than random sampling regardless of the resolution of the dataset due to its ability to learn discriminative multi-dimensional ST patterns.

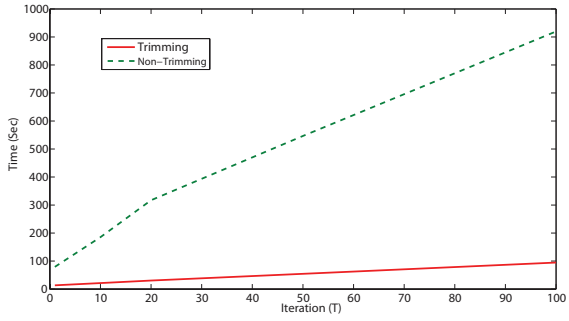


Fig. 5. The execution time of the CCRBoost approach without Negative-Sample-Trimming (Non-Trimming) and with Negative-Sample-Trimming (Trimming).

4.4 Complexity Analysis

To articulate the best ESTP, the $BuildChain()$ function has to search and evaluate all possible combinations of DSTPs. Therefore, this evaluation process is computationally expensive. Let's assume that there is an ESTP R consisting of n classifiers. We need to run the classification n times on the training dataset to obtain the evaluation results. Furthermore, in order to choose the next appropriate DSTP r from a pool of m candidates, we need to run $m(n+1)$ classifications.

According to the Negative-Sample-Trimming theorem, proved in Theorem 2.3, the $BuildChain()$ function only needs to run the classification m times using the trimmed dataset to find the best r to be added into R. Figure 5 shows that the execution time of CCRBoost is significantly reduced with Negative-Sample-Trimming while same Accuracy and

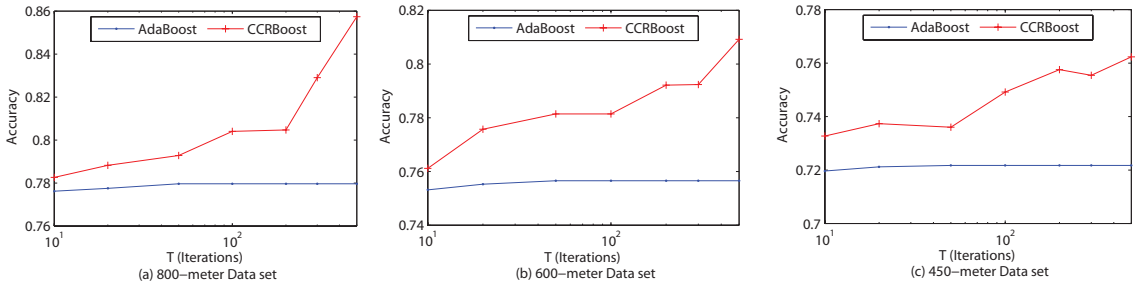


Fig. 6. Comparing CCRBoost with AdaBoost under different iterations T and different grid sizes.

F1-score are achieved as the function without Negative-Sample-Trimming.

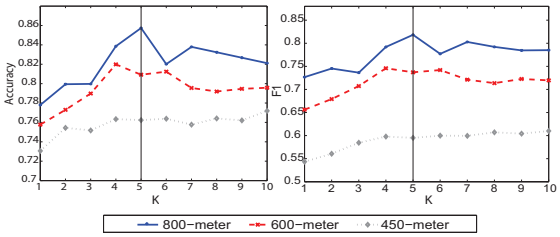


Fig. 7. The prediction results in Accuracy and F1-score on the datasets of different grid sizes using K as the control variable.

4.5 The Impact of Multi-clustering Level, K

In this section, we evaluate the impact of K. The efficiency of $BuildChain()$ is affected the total number of DSTP candidates. This number is controlled by the user-defined clustering variable, K, which is usually decided based on application domain and the amount of data. During these experiments, we set the other two user-defined variables $T = 500$ (The number of layers) and $\alpha = 0$ (The threshold for $H(\vec{x})$). K is evaluated using the three datasets with different grid cell sizes. The results of these experiments are shown in Figure 7. When $K = 1$, there is essentially no clustering and the entire monthly data set is the only cluster. Therefore, the results obtained from the settings of $K = 1$ are then used as the baseline to compare with other setting of K. We observe that using clustering yields better overall accuracy and the F1-score. This is because using clustering to find spatial distribution at local levels successfully captures the contextual patterns in its spatial and temporal dimensions. Moreover, we also find that the performance converges at a certain level when $K = 4$ and then maintains this level when $K \geq 5$. This shows that the patterns lose the true representation of local distributions when the size of cluster is too small.

In addition, the $BuildChain()$ function uses the greedy algorithm. Therefore, when the cluster size is too small, the ESTP tentatively fits the training data too precisely which may cause overfitting. Therefore, when evaluating the ensemble pattern using the test data, setting with $K > 5$ is not necessarily better than that setting of $K = 5$ due to the overfitting effect. Thus, K is set to be 5 in the rest of our empirical study.

4.6 Convergence Analysis

By considering CCRBoost as a specially designed boosting approach, we compare our approach with AdaBoost [27] using the number of iterations, T, as the control variable for convergence analysis. However, there are three fundamental differences between our proposed approach and AdaBoost. First, our approach uses ESTPs, hierarchically constructed from local level DSTPs, as the layer of our model, while AdaBoost directly uses the classifiers learned from globally or randomly selected instances as its weak learners. Secondly, we adopt unsupervised learning on the training data to select the most suitable features in advance, while AdaBoost depends on weak learners for feature selection. Lastly, we take spatio-temporal dimensions into consideration, while AdaBoost is regarded as a general ensemble classifier.

In this experiment, the convergences of these two approaches are indicated by the number of iterations needed to reach certain prediction accuracy. We chose LADTree as the weak classifier for AdaBoost, which is also the base classifier of a DSTP in our approach. Shown in Figure 6, the accuracy obtained from the AdaBoost reaches its ceiling when $T > 50$. However, our CCRBoost approach not only obtains better accuracy but also consistently achieves better convergences throughout three datasets.

TABLE 4
The results of comparing CCRBoost with the other approaches.

Dataset	800-meter		600-meter		450-meter	
Approach	Accuracy	F1	Accuracy	F1	Accuracy	F1
C4.5	0.500	0.667	0.500	0.667	0.500	0
NaiveBayes	0.730	0.675	0.703	0.647	0.667	0.592
LADTree	0.772	0.757	0.728	0.702	0.644	0.487
ANN	0.626	0.757	0.501	0.668	0.506	0.672
Voted Perceptron	0.780	0.792	0.733	0.721	0.576	0.335
K*	0.755	0.724	0.661	0.548	0.551	0.239
LogitBoost	0.779	0.782	0.737	0.741	0.715	**0.726**
Random Forests	0.754	0.752	0.714	0.700	0.670	0.641
SVM	0.817	0.801	0.776	0.742	0.651	0.489
CCRBoost	**0.857**	**0.818**	**0.820**	**0.746**	0.772	0.610
GWR	0.785	0.678	0.783	0.529	**0.807**	0.329

GWR: Geographically Weighted Regression

4.7 Comparing with The Other Approaches

Using the same crime datasets, we compare our proposed approach with the other state-of-the-art classification methods: (1) Support Vector Machine (SVM) [28] with a linear

kernel; (2) C4.5 [29] using confidence factor of 0.25; (3) Naive Bayes classifier; (4) LADTree [25]; (5) Artificial Neural Network (ANN) with one hidden layer; (6) Voted Perceptron [30] using $10,000$ as the maximum number of alterations; (7) K*, which is an instance-based classifier [31]; (8) LogitBoost [32] using 100 as its weight threshold; and (9) Random Forests [33] with 10 trees, are chosen in our comparison. As shown in Table 4, our proposed CCRBoost approach consistently has the best accuracy over the other methods across all three datasets. This is because our approach hierarchically incorporates the spatio-temporal patterns from a local level to a global level, from DSTP to ESTP, into the modeling processes while the other approaches learn the patterns only at a global level or without considering spatio-temporal context. Another observation is that using 800-meter grid yields better accuracy than using other two grid sizes. Since the indicators are the monthly aggregations of the crime events, finer resolution generates smaller values of the indicators. Our explanation for this observation is that the finer-sampled data introduces more noise as suggested by the Nyquist-Shannon Sampling Theorem [34].

TABLE 5
10-fold cross-validated paired significance test on CCRBoost.

Size	CCRBoost		SVM		Z	Result
	$\hat{K}$	$Var(\hat{K})$	$\hat{K}$	$Var(\hat{K})$		
800m	0.557	2.60*10^{-5}	0.540	3.17*10^{-5}	2.19	Significant
600m	0.429	2.26*10^{-5}	0.580	4.13*10^{-5}	3.60	Significant
450m	0.344	3.49*10^{-5}	0.160	1.80*10^{-5}	25.3	Significant

(a) CCRBoost v.s. SVM

Size	CCRBoost		LogitBoost		Z	Result
	$\hat{K}$	$Var(\hat{K})$	$\hat{K}$	$Var(\hat{K})$		
800m	0.557	2.60*10^{-5}	0.489	4.93*10^{-5}	7.73	Significant
600m	0.429	2.26*10^{-5}	0.460	4.55*10^{-5}	3.81	Significant
450m	0.344	3.49*10^{-5}	0.434	2.81*10^{-5}	11.3	Significant

(b) CCRBoost v.s. LogitBoost

We further perform the 10-fold cross-validated paired significance Z test [35] on CCRBoost with SVM and LogitBoost. Adopting the test setting used in [36], we calculate the $\hat{K}$ scores and set the significance threshold at 5%. Therefore, when the Z-score is greater than 1.96, CCRBoost is statistically significant. The significance test results listed in Table 5 show that CCRBoost is significant while comparing with both SVM and LogitBoost.

In addition to the classification approaches, we also compare CCRBoost with the Geographically Weighted Regression (GWR) approach [37], which involves using a spatial linear regression to model spatially varying relationships of the variables. To apply GWR, we generate twelve raster maps from the test data by month. Next, we let the GWR model's dependent variable be the class label, hotspot=1, and coldspot=-1. The six features are then used as the explanatory variables. The sign of the regression result of each test instance is considered as its predicted class. We obtain the performance measurements based on the prediction results generated by the GWR on all the test instances. With the grid side smaller than 450 meters, we have observed that GWR predicts almost all instances as the coldspots. This is because that the unbalanced data results in a nearly singular Hessian matrix for the model's log-likelihood function in most of the locations [38]. The performance of GWR shown in Table 4 indicates that GWR tends to be biased toward the coldspots on 450-meter dataset because of the low F1-score and does not perform better than the CCRBoost on the other two datasets.

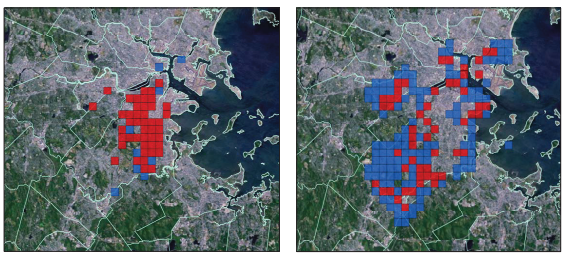
TABLE 6
The results of setting different α in our approach when $T = 100$ and $K = 5$.

α	*Accuracy*	*F1*	*Precision*	*Recall*
0	0.78787	0.73905	0.74631	**0.73194**
0.01	0.79710	0.74524	0.76877	0.72311
0.02	0.80566	0.74914	0.79656	0.70706
0.05	**0.80961**	**0.75693**	0.79505	0.72231
0.1	0.79710	0.74290	0.77391	0.71428
0.2	0.79216	0.72091	0.80295	0.65409
0.5	0.78985	0.71000	0.81865	0.62680
1	0.760540	0.60723	0.92892	0.45104
2	0.66040	0.29528	0.99539	0.17335
5	0.59782	0.03933	**1.00000**	0.02006

4.8 The Threshold α

Threshold α, used in Equation (18), determines whether an instance is predicted as a hotspot or not. α is always set to zero in our other experiments. To understand how this threshold affects the prediction performance, we use different values of α between 0 to 5 in this experiment. The other variables T and K are set to 100 and 5, respectively. According to the definition of confidence rate in Equation (15), the greater α is, the fewer false positive instances we should expect. From the results shown in Table 6, the precision of predicting hotspots increases when a larger α is used. On the other hand, the recall drops with increasing α. This means that our model predicts fewer hotspots. However, these hotspots are more likely to be true hotspots since they have high confidence rates. We obtain the same observation across datasets of different resolutions so we display only the results of the 800-meter dataset in Table 6. Thus, one of the advantages of our predictive model is that our model is able to detect high potential hotspots at which crime will have the highest chances of being committed in the future by setting a larger α.

4.9 The Final Global Spatio-Temporal Pattern



(1) September 2007 (2) August 2009

Fig. 8. The first two local patterns used in the final global spatio-temporal pattern result from the 800-meter grid dataset. The red blocks are hotspots and blue blocks are coldspots. (Better viewed in color)

The final product of the CCRBoost approach is an ensemble hierarchical pattern which is a global spatio-temporal

pattern and a physically interpretable rule-based pattern. We visualize the first two DSTPs of the final global pattern on the actual city map. Shown in Figure 8, we use the red grid blocks to represent hotspots and blue blocks for coldspots in both patterns. We investigate the first DSTP which is extracted from the monthly dataset of September 2007 and find that the locations of the first pattern are consistent with the known crime patterns in the target city. Verified by our domain experts, almost all the red blocks indicate the locations with above-average crime rates. The second DSTP is from August 2009 and identifies crime hotspots that are excluded from the first pattern. More importantly, the footprint of the second pattern is useful for pinpointing coldspots that have some protective factors against residential burglary and other crimes.

As a result, the first two DSTPs are complementary in identifying locations where we would expect residential burglary across the entire city as well as areas that were coldspots. Interestingly and consistent with the criminological literature, both patterns occur in the months when children are out of school and individuals may take vacations causing them to be less vigilant about protecting their property [20]. It may be that there is an increased likelihood of residential burglary in this city during this time period [21], [22].

Based on the consistency of our resulting patterns as compared to actual crime patterns, our approach does find the global pattern which recognizes not only the spatial but also the temporal factors that are useful for criminal justice professionals in predicting the incidents of future crime.

5 Related Work

From a predictive modeling point of views, we categorize the existing approaches into four categories, statistic mapping, mathematical modeling, clustering and classification, and association rule mining. In addition, we analyze the spatio-temporal patterns used in each of these approaches based on [39].

5.1 Statistic Mapping

Statistic mapping uses historical statistics to forecast crime occurring at the same location. In [40], histograms are used to present the "Additive Seasonal Factor" of the crime, which is the coefficient between a factor and a time period and a temporal pattern of one type of crime. Each bar in the histograms represents the seasonality of the respective factor at a location. For example, the police departments might use last September's crime summary as this September's crime forecast. This statistical model relies on the seasonality to predict future crime. However, it might miss a crime pattern with various regularities. Our proposed predictive model is intentionally designed to fit the ST patterns with various regularities.

5.2 Mathematical Modeling

Geographically Weighted Regression (GWR) is another model used to study crime patterns [41]. It involves one type of spatial regression model used to incorporate important spatial relationships among different variances. This type of pattern does not take temporal factors into consideration and is referred as a spatial pattern [39].

In [42], mathematical modeling is used to simulate the formation of a crime hotspot. This approach builds models based on two types of features: offender behaviors and relative locations. Each relative location is assigned with a density based on the crime frequency collectively obtained from the statistical models of individual offenders. This density is called the attractiveness value, which is calculated by a mathematical formula based on time length and linear stability. These hotspots overlap with each other, so a suppression process is needed to filter out the local maximum density as the true hotspots. This type of crime pattern is referred as the frequent spatio-temporal pattern [39] and does not incorporate seasonality or nonstationarity.

Later in [43], Mohler proposes a point-based model which eliminates the suppression step. Using the same concept in predicting aftershock, this model simulates how the crime spreads out, like diseases, from the initial background events. This approach has to identify which events are the initial background events and which are the aftershock events. Based on the initial and aftershock events, this model needs to optimize the two key variables used to measure the likelihood of a crime spreading into a neighborhood and how long it is likely to last. The hotspots defined in this paper are those locations covering the highest crime areas. This approach is suited better for capturing the crime patterns of short life cycles or abnormal situations at the local level. This type of patterns are the unusual spatio-temporal pattern as described in [39]. By considering different scenarios as the layers of ST patterns, the proposed approach is different because it includes not only the frequent but also the unusual spatio-temporal pattern in our predictive model.

5.3 Clustering and Classification

Clustering is adopted by Kumar et al. in [44] to define the geographic boundaries of each crime cluster. After the boundaries have been drawn, the crime density in a boxed cluster is regarded as the crime trend of this crime cluster. Local crime patterns identified by Kumar's approach are restricted by geographical distances from a center location. In contrast, our DSTP is designed to fit the nature of true distributions of crime incidents.

Besides clustering, classification models, such as decision tree, Naive Bayes, and SVM, are also used to predict crime in [13], [45], [46], [47]. These types of patterns identified by classification models vary according to the features involved in the modeling. One of the advantages of using classification models is that the societal or environmental factors can be incorporated into the predictive models.

In [46], Malathi adopts clustering not only to fill in the missing values of population sizes but also to obtain four levels of crime trends. This type of crime level indicates the trend of a group of the relative crime at the clustered locations during a period of time. There is common ground between Malathi's concept and our proposed DSTP. Both approaches identify the trend or pattern that the spatial distribution is a group of locations with similar indicators. However, Malathi's approach uses the clustered results directly to train a decision-tree based classifier as the predictive model to forecast the overall crime rate while our approach uses

hierarchical learning to build a layered model at different resolution levels, which is able to predict the occurrence of a target event at one individual location.

5.4 Association Rule Mining

Mohan et al. apply the principle of association rule mining to the discovery of an ST pattern, named cascading spatio-temporal pattern (CSTP) [4]. CSTP is defined as a frequent item set consisting of different types of events occurring within a certain distance and a certain time interval. Therefore, CSTP is also a frequent spatio-temporal pattern. To be qualified as a CSTP, a set of event types must pass two thresholds, the Cascade Participation Ratio (CPR) and the Cascade Participation Index (CPI). A CPR is the conditional probability of one event type included in one CSTP, which is the number of instances of one event type included in a CSTP divided by the total instances of this event type. CPI is defined as the minimum CPR within a CSTP.

The ST pattern identified by association rule mining is interpretable and easy to understand, such as the pattern of drunk-driving incidents after bars close on Saturday nights in locations near bars. However, this approach does not address ST nonstationarity while our proposed approach does. The CPR and CPI can be considered as crime indicators. Adding these two indicators, our approach can identify the DSTPs representing different crime occurring frequencies and then build a model that considers different crime regularities.

6 Conclusions

In summary, we study the hierarchical structure of ST patterns and propose a new approach, CCRBoost, to identify the hierarchical ST pattern and construct a predictive model. We also define the novel concept of DSTP to underlie the hierarchical structure. Next, we discover all the potential DSTPs through unsupervised multi-clustering and then use a greedy searching algorithm along with a pruning algorithm to combine the DSTPs into an ESTP. An ESTP is used to represent one of the scenarios of a real-world ST pattern. We further build a model with layers of weighted ESTPs to incorporate all the possible scenarios. This model is able to predict the occurrence of a target event at a given location. From our case study, we show that the ST patterns discovered by our approach are indicative of the true locations of residential burglaries. This gives concrete evidence that the proposed approach has the significant potential in predicting crime.

Predictive policing [48], [49] is a new trend in modern policing. With the ability to anticipate the emerging crime via our application to crime pattern identification, police will be able to more effectively fight or prevent crime using fewer resources. Through this research, one of our goals is to implement a crime prediction system which will provide timely crime forecasts with high accuracy and will require fewer data inputs. Furthermore, we will explore the potential of deploying our approach to discovering different types of ST patterns within different domains, such as flooding, or drought, or the occurrence of earthquakes. With accurate forecasts from our model, people will be given proper early warnings so that the damages caused by these natural disasters can be mitigated.

Acknowledgments

The authors would like to thank Mina Li for her effort in improving the readability of this paper.
This work was partially funded by the National Institute of Justice (No.2009- DE-BX-K219).

References

[1] H. Yang, S. Parthasarathy, and S. Mehta, "A generalized framework for mining spatio-temporal patterns in scientific data," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 716–721.

[2] H. Cao, N. Mamoulis, and D. W. Cheung, "Mining frequent spatio-temporal sequential patterns," in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 8–pp.

[3] J. Kang and H.-S. Yong, "Mining spatio-temporal patterns in trajectory data." *JIPS*, vol. 6, no. 4, pp. 521–536, 2010.

[4] P. Mohan, S. Shekhar, J. Shine, and J. Rogers, "Cascading spatio-temporal pattern discovery," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 11, pp. 1977–1992, 2012.

[5] J. H. Ratcliffe, "Aoristic signatures and the spatio-temporal analysis of high volume crime patterns," *Journal of Quantitative Criminology*, vol. 18, no. 1, pp. 23–43, 2002.

[6] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *Neural Networks, IEEE Transactions on*, vol. 22, no. 10, pp. 1517–1531, 2011.

[7] G. Villarini, J. A. Smith, and F. Napolitano, "Nonstationary modeling of a long record of rainfall and temperature over Rome," *Advances in Water Resources*, vol. 33, no. 10, pp. 1256–1267, 2010.

[8] Y. Bengio, "Scaling up deep learning," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: ACM, 2014, pp. 1966–1966.

[9] K. C. Wong, "Black's theory on the behavior of law revisited," *International Journal of The Sociology of Law*, vol. 23, pp. 189–232, 1995.

[10] S. N. Durlauf and L. E. Blume, *Social Norms in New Palgrave Dictionary of Economics*. Macmillan, 2011.

[11] C. A. Janzen, A. Deokar, and O. El-Gayar, "Discovering predictive event sequences in criminal careers," in *8th Annual Symposium on Information Assurance (ASIA13)*, 2013, pp. 73–82.

[12] T. C. Pratt and F. T. Cullen, "The empirical status of gottfredson and hirschi's general theory of crime: A meta-analysis," *Criminology*, vol. 38, no. 3, pp. 931–964, 2000.

[13] H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin, and M. Chau, "Crime data mining: a general framework and some examples," *Computer*, vol. 37, no. 4, pp. 50–56, 2004.

[14] I. Fountalis, A. Bracco, and C. Dovrolis, "Spatio-temporal network analysis for studying climate patterns," *Climate Dynamics*, vol. 42, no. 3-4, pp. 879–899, 2014.

[15] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37, no. 3, pp. 297–336, 1999.

[16] W. Cohen and Y. Singer, "A simple, fast, and effective rule learner," in *Proceedings of The Sixteenth National Conference on Artificial Intelligence*, 1999.

[17] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *ICML*, vol. 97. Nashville, USA, 1997, pp. 179–186.

[18] B. Garner, *A dictionary of modern legal usage*. Oxford University Press, 1987. [Online]. Available: http://books.google.com/books?id=2kQEAQAAIAAJ

[19] M. Townsley, R. Homel, and J. Chaseling, "Infectious burglaries. a test of the near repeat hypothesis," *British Journal of Criminology*, vol. 43, no. 3, pp. 615–633, 2003.

[20] T. Bennett, L. Durie, and G. Britain, *Preventing residential burglary in Cambridge: From crime audits to targeted strategies*. Home Office, Policing and Reducing Crime Unit, Research, Development and Statistics Directorate, 1999, no. 108.

[21] G. Farrell and P. PEASE, "Crime seasonality: Domestic disputes and residential burglary in merseyside 1988–90," *British Journal of Criminology*, vol. 34, no. 4, pp. 487–498, 1994.
[22] X. Yang, "Exploring the influence of environmental features on residential burglary using spatial-temporal pattern analysis," Ph.D. dissertation, University of Florida, 2006.
[23] S. D. Johnson and K. J. Bowers, "The stability of space-time clusters of burglary," *British Journal of Criminology*, vol. 44, no. 1, pp. 55–65, 2004.
[24] J. M. McGloin, C. J. Sullivan, A. R. Piquero, and T. C. Pratt, "Local life circumstances and offending specialization/versatility comparing opportunity and propensity models," *Journal of Research in Crime and Delinquency*, vol. 44, no. 3, pp. 321–346, 2007.
[25] G. Holmes, B. Pfahringer, R. Kirkby, E. Frank, and M. Hall, "Multiclass alternating decision trees," in *ECML*. Springer, 2001, pp. 161–172.
[26] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
[27] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory: Eurocolt*. Springer-Verlag, 1995, pp. 23–37.
[28] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[29] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.
[30] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Machine learning*, vol. 37, no. 3, pp. 277–296, 1999.
[31] J. G. Cleary, L. E. Trigg *et al.*, "K*: An instance-based learner using an entropic distance measure," in *ICML*, 1995, pp. 108–114.
[32] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.
[33] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
[34] R. B. Blackman and J. W. Tukey, "The measurement of power spectra from the point of view of communications engineeringpart i," *Bell System Technical Journal*, vol. 37, no. 1, pp. 185–282, 1958.
[35] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, vol. 10, no. 7, pp. 1895–1923, 1998.
[36] Z. Jiang, S. Shekhar, X. Zhou, J. Knight, and J. Corcoran, "Focal-test-based spatial decision tree learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 27, no. 6, pp. 1547–1559, June 2015.
[37] S. Fotheringham, M. Charlton, and C. Brunsdon, "Geographically weighted regression: a natural evolution of the expansion method for spatial data analysis," *Environment and planning A*, vol. 30, no. 11, pp. 1905–1927, 1998.
[38] Y. Wang, K. M. Kockelman, and X. C. Wang, "Anticipating land use change using geographically weighted regression models for discrete response," in *Presented at the 90th Annual Meeting of the Transportation Research Board*, vol. 34, 2011, p. 35.
[39] K. Leong and A. Sung, "A review of spatio-temporal pattern analysis approaches on crime analysis," *International E-journal of Criminal Sciences*, no. 9, 2015.
[40] J. Cohen and W. L. Gorr, *Development of crime forecasting and mapping systems for use by police*. H. John Heinz III School of Public Policy and Management, Carnegie Mellon University, 2005.
[41] M. Cahill and G. Mulligan, "Using geographically weighted regression to explore local crime patterns," *Social Science Computer Review*, vol. 25, no. 2, pp. 174–193, 2007.
[42] M. B. Short, A. L. Bertozzi, and P. J. Brantingham, "Nonlinear patterns in urban crime: Hotspots, bifurcations, and suppression," *SIAM Journal on Applied Dynamical Systems*, vol. 9, no. 2, pp. 462–483, 2010.
[43] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, no. 493, 2011.
[44] M. V. Kumar and C. Chandrasekar, "Spatial clustering simulation on analysis of spatialtemporal crime hotspot for predicting crime activities," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 6, pp. 2867–2864, 2011.
[45] C. Yu, M. W. Ward, M. Morabito, and W. Ding, "Crime forecasting using data mining techniques," in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on Data Mining*. IEEE, 2011, pp. 779–786.
[46] A. Malathi and D. S. Baboo, "An enhanced algorithm to predict a future crime using data mining," *International Journal of Computer Applications*, vol. 21, no. 1, pp. 1–6, May 2011, published by Foundation of Computer Science.
[47] R. Iqbal, M. A. A. Murad, A. Mustapha, P. H. S. Panahy, and N. Khanahmadliravi, "An experimental study of classification algorithms for crime prediction," *Indian Journal of Science and Technology*, vol. 6, no. 3, pp. 4219–4225, 2013.
[48] B. Pearsall, "Predictive policing: The future of law enforcement," *National Institute of Justice Journal*, vol. 266, pp. 16–19, 2010.
[49] S. Greengard, "Policing the future," *Communications of the ACM*, vol. 55, no. 3, pp. 19–21, 2012.

Chung-Hsien Yu Chung-Hsien Yu is a Ph.D. candidate of Department of Computer Science at University of Massachusetts Boston and a member of Knowledge Discovery Lab (KDL) directed by Prof. Wei Ding. His research interests include Spatiotemporal Data Mining, Spatiotemporal Pattern Recognition, and Computer Vision, with application to environmental sciences, geosciences, and human sciences. Currently, his researches focus on developing innovative data mining and pattern recognition techniques to acquire patterns from specific spatiotemporal data, crime and climate.

Wei Ding Wei Ding received her Ph.D. degree in Computer Science from the University of Houston in 2008. She is an Associate Professor of Computer Science in the University of Massachusetts Boston. Her research interests include data mining, machine learning, artificial intelligence, computational semantics, and with applications to astronomy, geosciences, and environmental sciences. She has published more than 85 referred research papers, 1 book, and has 2 patents. She is an Associate Editor of Knowledge and Information Systems (KAIS) and an editorial board member of the Journal of Information System Education (JISE), the Journal of Big Data, and the Social Network Analysis and Mining Journal. Her research projects are currently sponsored by NASA and DOE. She is an IEEE senior member and an ACM senior member.

Melissa Morabito Melissa Schaefer Morabito is an Assistant Professor in the School of Criminology and Justice Studies at the University of Massachusetts, Lowell and an Associate with the Center for Women & Work. Her research interests include the police response to people with mental illness, sexual violence and women and policing, she received her Ph.D in Justice, Law & Society from American University and completed a National Institute of Mental Health postdoctoral fellowship at the Center for Mental Health Services & Criminal Justice Research.

Ping Chen Dr. Ping Chen is an Associate Professor of Computer Science and Director of Artificial Intelligence Lab at the University of Massachusetts Boston. His research interests include Data Mining and Computational Semantics. Dr. Chen has received seven National Science Foundation grants, one grant from Department of Homeland Security, one grant from Veteran Affairs, and published over 60 papers in major Data Mining, Artificial Intelligence, and Computational Linguistics conferences and journals. Dr. Ping Chen received his BS degree on Information Science from Xi'an Jiao Tong University, MS degree on Computer Science from Chinese Academy of Sciences, and Ph.D degree on Information Technology at George Mason University.