Scaling Up Event and Pattern Detection to Big Data

Daniel B. Neill
H.J. Heinz III College
Carnegie Mellon University
E-mail: neill@cs.cmu.edu

We gratefully acknowledge funding support from the National Science Foundation, grants IIS-0916345, IIS-0911032, and IIS-0953330.
My research is focused at the intersection of **machine learning** and **public policy**.

Increasingly critical importance of addressing global policy problems (disease pandemics, crime, terrorism…) and Continuously increasing size and complexity of policy data, and rapid growth of new and transformative technologies.

Machine learning has become increasingly essential for data-driven policy analysis and for the development of new, practical information technologies that can be directly applied for the public good (e.g. public health, safety, and security).

My research in this area has two main goals:

1) Develop new machine learning methods for better (more scalable and accurate) detection and prediction of events and other patterns in massive datasets.

2) Apply these methods to improve the quality of public health, safety, and security.
Daniel B. Neill (neill@cs.cmu.edu)
Associate Professor of Information Systems, Heinz College, CMU
Director, Event and Pattern Detection Laboratory
Courtesy Associate Professor of Machine Learning and Robotics

Disease Surveillance:
Very early and accurate detection of emerging outbreaks.

Law Enforcement:
Detection, prediction, and prevention of “hot-spots” of violent crime.

Medicine:
Discovering new “best practices” of patient care, to improve outcomes and reduce costs.

Our disease surveillance methods are currently in use for deployed systems in the U.S., Canada, India, and Sri Lanka.

Our “CrimeScan” software has been in day-to-day operational use for predictive policing by the Chicago PD.

“CityScan” is being evaluated for prediction and prevention of rodent infestations using 311 call data.
Pattern detection by subset scan

One key insight that underlies much of my work is that pattern detection can be viewed as a search over subsets of the data.

**Statistical challenges:**
- Which subsets to search?
- Is a given subset anomalous?
- Which anomalies are relevant?

**Computational challenge:**
- How to make this search over subsets efficient for massive, complex, high-dimensional data?

New statistical methods enable more timely and more accurate detection by integrating multiple data sources, incorporating spatial and temporal information, and using prior knowledge of a domain.

New algorithms and data structures make previously impossible detection tasks computationally feasible and fast.

New machine learning methods enable our systems to learn from user feedback, modeling and distinguishing between relevant and irrelevant types of anomaly.
1) Multivariate event detection

Spatial time series data from spatial locations $s_i$ (e.g. zip codes)

Time series of counts $c_{i,m}^t$ for each zip code $s_i$ for each data stream $d_m$.

Outbreak detection
- $d_1 =$ respiratory ED
- $d_2 =$ constitutional ED
- $d_3 =$ OTC cough/cold
- $d_4 =$ OTC anti-fever (etc.)

Main goals:
- **Detect** any emerging events.
- **Pinpoint** the affected subset of locations and time duration.
- **Characterize** the event by identifying the affected streams.

Compare hypotheses:
- $H_1(D, S, W)$
  - $D =$ subset of streams
  - $S =$ subset of locations
  - $W =$ time duration
- vs. $H_0$: no events occurring
Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)

We search for spatial regions (subsets of locations) where the recently observed counts for some subset of streams are significantly higher than expected.

We perform **time series analysis** to compute expected counts ("baselines") for each location and stream for each recent day.

We then compare the actual and expected counts for each subset (D, S, W) under consideration.
Expectation-based scan statistics

(Kulldorff, 1997; Neill and Moore, 2005)

We find the subsets with highest values of a likelihood ratio statistic, and compute the $p$-value of each subset by randomization testing.

$$F(D, S, W) = \frac{\Pr(\text{Data} \mid H_1(D, S, W))}{\Pr(\text{Data} \mid H_0)}$$

To compute $p$-value

Compare subset score to maximum subset scores of simulated datasets under $H_0$.

Maximum subset score = 9.8

2nd highest score = 8.4

Not significant ($p = .098$)

Significant! ($p = .013$)

F1* = 2.4

F2* = 9.1

F999* = 7.0

F1* = 2.4

F2* = 9.1

F999* = 7.0
Likelihood ratio statistics

For our expectation-based scan statistics, the null hypothesis $H_0$ assumes “business as usual”: each count $c_{i,m}^t$ is drawn from some parametric distribution with mean $b_{i,m}^t$. $H_1(S)$ assumes a multiplicative increase for the affected subset $S$.

**Expectation-based Poisson**

- $H_0$: $c_{i,m}^t \sim \text{Poisson}(b_{i,m}^t)$
- $H_1(S)$: $c_{i,m}^t \sim \text{Poisson}(qb_{i,m}^t)$

Let $C = \sum_S c_{i,m}^t$ and $B = \sum_S b_{i,m}^t$.

Maximum likelihood: $q = C / B$.

$F(S) = C \log (C/B) + B - C$

**Expectation-based Gaussian**

- $H_0$: $c_{i,m}^t \sim \text{Gaussian}(b_{i,m}^t, \sigma_{i,m}^t)$
- $H_1(S)$: $c_{i,m}^t \sim \text{Gaussian}(qb_{i,m}^t, \sigma_{i,m}^t)$

Let $C' = \sum_S c_{i,m}^t b_{i,m}^t / (\sigma_{i,m}^t)^2$ and $B' = \sum_S (b_{i,m}^t)^2 / (\sigma_{i,m}^t)^2$.

Maximum likelihood: $q = C' / B'$.

$F(S) = (C')^2 / 2B' + B'/2 - C'$

Many possibilities: exponential family, nonparametric, Bayesian…
Which regions to search?

Typical approach: “spatial scan” (Kulldorff, 1997)

Each search region $S$ is a sub-region of space.

- Choose some region shape (e.g. circles, rectangles) and consider all regions of that shape and varying size.
- Low power for true events that do not correspond well to the chosen set of search regions (e.g. irregular shapes).

Our approach: “subset scan” (Neill, 2012)

Each search region $S$ is a subset of locations.

- Find the highest scoring subset, subject to some constraints (e.g. spatial proximity, connectivity).
- For multivariate, also optimize over subsets of streams.
- Exponentially many possible subsets, $O(2^N \times 2^M)$: computationally infeasible for naïve search.
2) General pattern detection

Set of data records \( R_i \) (e.g., container shipments)

Main goals:

Detect any anomalous patterns.

Pinpoint the affected subset of data records.

Characterize the pattern by identifying the affected subset of attributes.

Compare hypotheses:

\[ H_1(R, A) \]

\( R = \) subset of records
\( A = \) subset of attributes

vs. \( H_0: \) no events occurring

Attribute value \( v_{ij} \) for each attribute \( A_j \) for each record \( R_i \).
2) General pattern detection

Set of **data records** $R_i$ (e.g., container shipments)

<table>
<thead>
<tr>
<th>DATE</th>
<th>F PORT</th>
<th>US PORT</th>
<th>COUNTRY</th>
<th>LINE</th>
<th>CARGO</th>
<th>SIZE</th>
<th>WEIGHT</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Jan</td>
<td>TOKYO</td>
<td>SEATTLE</td>
<td>JAPAN</td>
<td>CSCO</td>
<td>EMPTY</td>
<td>20</td>
<td>5.6</td>
<td>27579</td>
</tr>
<tr>
<td>1-Jan</td>
<td>TOKYO</td>
<td>SEATTLE</td>
<td>JAPAN</td>
<td>CSCO</td>
<td>TIRES</td>
<td>40</td>
<td>13.43</td>
<td>9497</td>
</tr>
<tr>
<td>1-Jan</td>
<td>TOKYO</td>
<td>SEATTLE</td>
<td>JAPAN</td>
<td>CSCO</td>
<td>IODINE</td>
<td>20</td>
<td>17.68</td>
<td>251151</td>
</tr>
</tbody>
</table>

**Attribute value** $v_{ij}$ for each attribute $A_j$ for each record $R_i$.

**Fast Generalized Subset Scan (McFowland et al., 2013):**

1) Learn Bayesian network structure and parameters from data.
2) Compute conditional probability of each attribute value.
3) Convert to empirical $p$-values (uniform on $[0,1]$ under $H_0$)
4) Find subsets of records and attributes with higher than expected numbers of low (significant) empirical $p$-values.

$O(2^N \times 2^M)$ subsets: computationally infeasible for naïve search!
Question: Why search over subsets?
Answer: Simpler approaches can fail.

**Top-down detection approaches**

Are there any globally interesting patterns? If so, recursively search the most interesting sub-partition.

Two examples: bump hunting; “cluster then detect”.

Top-down fails for **small-scale patterns** that are not evident from the global aggregates.

**Bottom-up detection approaches**

Find individually (or locally) anomalous data points, and optionally, aggregate into clusters.

Two examples: anomaly/outlier detection; density-based clustering.

Bottom-up fails for **subtle patterns** that are only evident when a group of data records are considered collectively.
Question: Why search over subsets?
Answer: Simpler approaches can fail.

Top-down detection approaches
Are there any globally interesting patterns? If so, recursively search the most interesting sub-partition.

Bottom-up detection approaches
Two examples: bump hunting; “cluster then detect”.
Find individually (or locally) anomalous data points, and optionally, aggregate into clusters.
Two examples: anomaly/outlier detection; density-based clustering.

So here’s where we are so far:
Treating pattern detection as a subset scan problem is statistically desirable for maximizing detection power…

but computationally infeasible (for exhaustive search at least).

Top-down fails for 
**patterns** that are not evident from the global aggregates.

Bottom-up fails for **subtle patterns** that are evident when a group of data records are considered collectively.
Fast subset scan

• In certain cases, we can optimize $F(S)$ over the exponentially many subsets of the data, while evaluating only $O(N)$ rather than $O(2^N)$ subsets.

• Many commonly used scan statistics have the property of **linear-time subset scanning**:
  • Just sort the data records (or spatial locations, etc.) from highest to lowest priority according to some function…
  • … then search over groups consisting of the top-$k$ highest priority records, for $k = 1..N$.

  The highest scoring subset is **guaranteed** to be one of these!

**Sample result**: we can find the **most anomalous** subset of Allegheny County zip codes in **0.03 sec** vs. **$10^{24}$ years**.
Linear-time subset scanning

• Example: Expectation-Based Poisson statistic
  • Sort data locations $s_i$ by the ratio of observed to expected count, $c_i / b_i$.
  • Given the ordering $s_{(1)} \ldots s_{(N)}$, we can prove that the top-scoring subset $F(S)$ consists of the locations $s_{(1)} \ldots s_{(k)}$ for some $k$, $1 \leq k \leq N$.
  • Key step: if there exists some location $s_{\text{out}} \notin S$ with higher priority than some location $s_{\text{in}} \in S$, then we can show that $F(S) \leq \max(F(S \cup \{s_{\text{out}}\}), F(S \setminus \{s_{\text{in}}\}))$.

• Theorem: LTSS holds for convex functions of two additive sufficient statistics.

• Theorem: LTSS holds for all expectation-based scan statistics in any separable exponential family.
Constrained fast subset scanning

LTSS is a new and powerful tool for **exact** combinatorial optimization (as opposed to approximate techniques such as submodular function optimization). But it only solves the “best unconstrained subset” problem, and cannot be used directly for **constrained** optimization.

Much of our recent work has focused on how LTSS can be extended to the many real-world problems with (hard or soft) constraints on our search.

- Proximity constraints → Fast spatial scan (irregular regions)
- Multiple data streams → Fast multivariate scan
- Connectivity constraints → Fast graph scan
- Group self-similarity → Fast generalized subset scan
Fast subset scan with spatial proximity constraints

• Maximize a likelihood ratio statistic over all subsets of the “local neighborhoods” consisting of a center location $s_i$ and its $k-1$ nearest neighbors, for a fixed neighborhood size $k$.

• Naïve search requires $O(N \cdot 2^k)$ time and is computationally infeasible for $k > 25$.

• For each center, we can search over all subsets of its local neighborhood in $O(k)$ time using LTSS, thus requiring a total time complexity of $O(Nk) + O(N \log N)$ for sorting the locations.

• In Neill (2012), we show that this approach dramatically improves the timeliness and accuracy of outbreak detection for irregularly-shaped disease clusters.
Multivariate fast subset scan

(Neill, McFowland, and Zheng, 2013)

• The LTSS property allows us to efficiently optimize over subsets of spatial locations for a given subset of data streams.
• But it also allows us to efficiently optimize over subsets of streams for a given subset of locations…
• So we can jointly optimize over subsets of streams and locations by iterating between these two steps!
• Convergence to local (conditional) maximum \(\Rightarrow\) need to do multiple restarts to approach the global maximum.
• For general pattern detection problems, a similar approach can be used to jointly optimize over subsets of data records and attributes in our FGSSS approach.
Incorporating connectivity constraints

Proximity-constrained subset scans may return a disconnected subset of the data. In some cases this may be undesirable, or we might have non-spatial data so proximity constraints cannot be used.

Example: tracking disease spread from person-to-person contact.

Example: identifying a connected subset of zip codes (Allegheny County, PA)
Incorporating connectivity constraints

Proximity-constrained subset scans may return a disconnected subset of the data.

In some cases this may be undesirable, or we might have non-spatial data so proximity constraints cannot be used.

Our GraphScan algorithm* can efficiently and exactly identify the highest-scoring connected subgraph:

- Can incorporate multiple data streams
- With or without proximity constraints
- Graphs with several hundred nodes

We can use the LTSS property to rule out subgraphs that are provably suboptimal, dramatically reducing our search space.

* Speakman, McFowland, and Neill, 2013 (submitted)
Evaluation: run times

Average Runtime for a Single Day of ED data

- GraphScan
- GraphScan w/ no Branch & Bounding
- Flexscan

seconds vs. Neighborhood Size (k)
Evaluation: detection power

Comparison of detection power for outbreaks along highways

- GraphScan
- circles
- All subsets
Incorporating soft constraints

(Speakman, Somanchi, McFowland, and Neill, 2014, submitted)

• So far we have talked about **hard** constraints (i.e., restrictions on the search space, ruling out some subsets).
• What about **soft** constraints?
  • We would like to search over all subsets, but reward more likely subsets and penalize those that are less likely.

For functions satisfying the **Additive Linear Time Subset Scanning** property, conditioning on the relative risk, \( q \), allows the function to be written as an *additive* set function over the data elements \( s_i \) in \( S \).

Expectation-based scan statistics in a one-parameter exponential family

(not just separable exponential family!)

\[
F(S) = \max_{q > 1} \log \frac{P(Data \mid H_1(S))}{P(Data \mid H_0)}
\]

\[H_0 : x_i \sim \text{Dist}(\mu_i)\]

\[H_1 : x_i \sim \text{Dist}(q\mu_i)\]
Penalized Fast Subset Scanning

For functions satisfying the **Additive Linear Time Subset Scanning** property, conditioning on the relative risk, $q$, allows the function to be written as an *additive* set function over the data elements $s_i$ in $S$.

**Consequence #1**: Extremely easy to maximize $F(S)$ over subsets, for a given $q$, by including all “positive” elements and excluding “negative”.

**Consequence #2**: Additional, element-specific penalty terms may be added to the scoring function while maintaining the additive property.

**Expectation-based Poisson:**

$$F(S) = \max_{q>1} \sum_{s_i \in S} x_i (\log q) + \mu_i (1 - q)$$
Penalized Fast Subset Scanning

For functions satisfying the Additive Linear Time Subset Scanning property, conditioning on the relative risk, $q$, allows the function to be written as an additive set function over the data elements $s_i$ in $S$.

Consequence #1: Extremely easy to maximize $F(S)$ over subsets, for a given $q$, by including all “positive” elements and excluding “negative”.

Consequence #2: Additional, element-specific penalty terms may be added to the scoring function while maintaining the additive property.

"Total Contribution" $\gamma_i$ of record $s_i$ for fixed risk, $q$

Expectation-based Poisson:

$$F_{\text{penalized}}(S) = \max_{q > 1} \sum_{s_i \in S} \left[ x_i (\log q) + \mu_i (1 - q) + \Delta_i \right]$$
Penalized Fast Subset Scanning

For functions satisfying the **Additive Linear Time Subset Scanning** property, conditioning on the relative risk, $q$, allows the function to be written as an *additive* set function over the data elements $s_i$ in $S$.

**Consequence #1:** Extremely easy to maximize $F(S)$ over subsets, for a given $q$, by including all “positive” elements and excluding “negative”.

**Consequence #2:** Additional, element-specific penalty terms may be added to the scoring function while maintaining the additive property.

How to optimize efficiently over all values of $q$, not just a given $q$???

**Theorem:** the optimal subset $S^* = \arg \max_S F_{\text{pen}}(S)$ for a penalized expectation-based scan statistic satisfying the ALTSS property may be found by evaluating only $O(N)$ of the $2^N$ subsets of data records.
“Proof by picture”

\[ \gamma_i(q) = 0 \]  

Relative Risk \( q > 1 \)
“Proof by picture”

\[ \gamma_1 = 130 \log q + 110(1 - q) + 0 \]

At most 2 zeros

Relative Risk \( q > 1 \)

Record 1

\[ x_1 = 130 \]
\[ \mu_1 = 110 \]
\[ \Delta_1 = 0 \]
Proof by picture

\[ x_1 = 130 \]
\[ \mu_1 = 110 \]
\[ \Delta_1 = 0 \]

Relative Risk \( q > 1 \)

\[ x_2 = 26 \]
\[ \mu_2 = 20 \]
\[ \Delta_2 = 0.5 \]

Record 1
Record 2
\[
\begin{align*}
\gamma_i(q) &\quad 0 \\
\Delta_1 &= 0 \\
x_1 &= 130 \\
\mu_1 &= 110 \\
\Delta_2 &= 0.5 \\
x_2 &= 26 \\
\mu_2 &= 20 \\
\Delta_3 &= -1 \\
x_3 &= 40 \\
\mu_3 &= 30
\end{align*}
\]
Proof by picture

At most 2N intervals

Relative Risk \( q > 1 \)

\( \gamma_i(q) \)
“Proof by picture”

\[
\gamma_i(q) = \begin{cases} 
1 & q > 1 \\
0 & q = 1 \\
-1 & q < 1 
\end{cases}
\]

Relative Risk \( q > 1 \)
“Proof by picture”

\[
\gamma_i(q)
\] vs Relative Risk \(q > 1\)

- **Record 1**
- **Record 2**
- **Record 3**
“Proof by picture”

Relative Risk $q > 1$

$\gamma_i(q)$

- $\gamma_i(q)$ vs. Relative Risk $q > 1$
- $I_1$, $I_2$, $I_3$, $I_4$
- Record 1, Record 2, Record 3
“Proof by picture”

The graph shows three records labeled \( \gamma_i(q) \) for different values of \( q > 1 \). The records are distinguished by different colors:
- **Record 1**: Blue
- **Record 2**: Red
- **Record 3**: Green

Each record is plotted against the relative risk \( q \), with the \( \gamma_i(q) \) values on the vertical axis and \( q > 1 \) on the horizontal axis. The graph illustrates the relationship between the relative risk and the \( \gamma_i(q) \) values for each record.
Penalized Fast Subset Scanning is a general framework for scalable pattern detection with soft constraints.

- **Exactness**: The most anomalous (highest scoring) subset is guaranteed to be identified.

- **Efficiency**: Only $O(N)$ subsets must be scanned in order to identify the most anomalous penalized subset in a dataset containing $N$ elements.

- **Interpretability**: Soft constraints may be viewed as the prior log-odds for a given record to be included in the most anomalous penalized subset.
Detecting and Tracking Dynamic Patterns

Most subset scan methods have difficulty dealing with dynamic patterns, where the affected subset changes over time.

Optimizing each time step independently fails, as does neglecting event dynamics.

Our solution, Dynamic Subset Scan, uses soft constraints on temporal consistency to pass information between time steps.
Detecting and Tracking Dynamic Patterns

Dynamic Subset Scan algorithm

1) Identify subsets $S_t$ independently for each time step $t$, using unpenalized fast subset scan.

2) Repeat until convergence:
   a) Choose a time step $t$.
   b) Compute $\Delta_{i}^t$ for each location $s_i$, given subsets $S_{t-1}$ and $S_{t+1}$.
   c) Find new optimal subset $S_t$ using penalized fast subset scan with the given $\Delta_{i}^t$.

Generative model

$$\log\left(\frac{p_i^t}{1 - p_i^t}\right) = \beta_0 + \beta_1 X_{i}^{t-1} + \beta_2 \frac{n_i^{t-1}}{k_i}$$

Prior log-odds that location $s_i$ affected on time step $t$.

Equals 1 if location $s_i$ affected on time step t-1.

Fraction of neighbors affected on time step t-1.

Computing $\Delta_{i}^t$ is complicated, since we must incorporate both $\text{Pr}(S_t$ generated from $S_{t-1})$ and $\text{Pr}(S_t$ generates $S_{t+1})$. 

$$\Delta_i^t \approx \beta_1 (x_i^{t-1} + x_i^{t+1} - 1) + \beta_2 \left( \frac{n_i^{t-1}}{k_i} + \sum_{\text{neighbors } j \in S_{t+1}} \frac{1}{k_j} - \frac{1}{2} \sum_{\text{neighbors } j} \left( \frac{1}{k_i} + \frac{1}{k_j} \right) \right)$$
Tracking Contaminant Plumes

Dynamic Subset Scan improves event tracking, as measured by overlap coefficient between the true and detected regions.
Detecting Contaminant Plumes

Dynamic Subset Scan improves event detection, as measured by average number of hours needed to detect.
Dynamic Subset Scan improves accuracy for locating the source of the event, as measured by overlap between true and detected regions.
Scaling up to even **bigger** data…

Currently the fast subset scan scales to datasets with **millions** of records.

But enforcing certain hard constraints (e.g., graph connectivity) dramatically impacts scalability.

Spatial constraints (FSS)
Similarity constraints (FGSS)
Soft constraints (PFSS)

GraphScan: 250 nodes
Additive Graphscan: 25K nodes

How to scale up to larger graphs with millions of nodes?

ongoing EPD Lab research

How to scale up to datasets with billions or trillions of records?

Many possible answers!

Sampling
Parallelization
Problem Partitioning
Randomization

Locality-Sensitive Hashing
Sublinear-Time Algorithms
Summarization
Hierarchy
Idea #1: Massive parallelization

For example, what if we have a trillion records but a million processors?

Certain aspects of fast subset scan are **trivially parallelizable:**
- Randomization testing, to determine statistical significance.
- Scanning over many local neighborhoods (with proximity constraints).
- Scoring many subsets (but not exponentially many!).

For **unconstrained subset scan**, we have the necessary pieces:
- Parallel sorting (merge sort, sample sort): $O(\log N)$ with $N$ processors.
- “Scan” (accumulate sums of top-$k$ elements by priority): $O(\log N)$.

To incorporate **spatial proximity** or more general **similarity** constraints:
- **Locality-sensitive hashing** $\rightarrow$ neighborhoods of similar elements.

With more general constraints (e.g., graphs), we must develop new ways to partition the search space and merge solutions to sub-problems.
Idea #2: Incorporate hierarchy

**Subsampling** the raw data can miss a arbitrarily strong signal that affects a small enough proportion of the dataset.

Possible solution: *summarization.*

Represent the data *hierarchically*, maintain summary statistics at each level of hierarchy, and search over coarse and fine resolutions.

**Goal:** find the most interesting subsets while only looking at a small fraction of the raw data.

**Challenge 1:** building the hierarchy may be expensive (though parallelizable).

**Challenge 2:** how to search the hierarchy, so that we are unlikely to miss small areas?

*Example: image data digital pathology slides, satellite images, etc.*

*Hierarchical Linear-Time Subset Scanning*  
(Somanchi & Neill, DMHI 2013)
Idea #2: Incorporate hierarchy

Example: image data
digital pathology slides, satellite images, etc.

HLTSS has been successfully applied to detect regions of interest in digital pathology slides, and works surprisingly well to detect prostate cancer!

Hierarchical Linear-Time Subset Scanning
(Somanchi & Neill, DMHI 2013)
Idea #2: Incorporate hierarchy

HLTSS improves both the accuracy of detecting which pixels within a slide are cancerous (left panel) and the ability to differentiate cancerous from non-cancerous slides (right panel).
Current application domains

**Biosurveillance:** deployed systems in Ottawa, Grey-Bruce, Sri Lanka, India.

In progress: deployments in Canada for monitoring hospital-acquired illness, and patterns of harm related to drug abuse.

**Crime prediction in Chicago:**
Able to predict about 83% of “clustered” violent crimes and 57% of all violent crimes, with 15% false positive rate.

**Detecting anomalous patterns of care in UPMC hospitals:**
Our goal is to find atypical treatment conditions that improve patient outcomes (“best practices”) or harm patients (systematic errors, improper hygiene, etc.).

Many more applications:

- Illicit container shipments
- Clusters of water pipe breaks
- Spreading water contamination
- Network intrusion detection
- Economic growth “outbreaks”
- Conflict, violence, human rights (predicting civil unrest using Twitter)
References


Interested?

More details on our web site:
http://epdlab.heinz.cmu.edu

Or e-mail me at:
neill@cs.cmu.edu