An Active Learning Framework Incorporating User Input For Mining Urban Data

A Case Study in Dublin, Ireland

Nikolas Zygouras
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens, Greece
nzygouras@di.uoa.gr

Nikolaos Panagiotou
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens, Greece
npanagiotou@di.uoa.gr

Nikos Zacheilas
Department of Informatics
Athens University of Economics and Business, Greece
zacheilas@aueb.gr

Ioannis Boutsis
Department of Informatics
Athens University of Economics and Business, Greece
mpoutsis@aueb.gr

Vana Kalogeraki
Department of Informatics
Athens University of Economics and Business, Greece
vana@aueb.gr

Dimitrios Gunopulos
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens, Greece
dg@di.uoa.gr

ABSTRACT

Analyzing and detecting events from ubiquitous sensors across the city has been an important goal in recent years. Different techniques that are able to automatically detect events by monitoring urban sensor's data have been efficiently applied in several smart cities to improve the citizens everyday life. However, the analysis of such voluminous data streams often interferes with several constraints that arise in smart cities scenarios. For example it is impossible to hire human oracles that will monitor each data stream continuously to provide knowledge to these models and to annotate past instances. Thus, the development of novel techniques is required in order to build efficient supervised learning models that will be able to cope with urban data deluge. Our approach makes the following contributions: (i) we formulate the problem of building supervised learning models efficiently by incorporating streaming input from urban data, and (ii) we present a novel framework that is able to cope with the restrictions that arise in the event detection of streaming urban data, requiring labels from carefully selected instances.

Keywords

Urban Data; Active Learning;Smart Cities;Event Detection

1. INTRODUCTION

Urban data monitoring has recently been efficiently applied in different smart cities providing immediately useful and accurate information regarding several incidents to the city authorities in order to handle events of emergency. Nowadays, the large development of Internet of Things, with ubiquitous sensors across the city, create voluminous and heterogeneous data streams that need to be automatically monitored. Such data streams may be provided either from moving sensors or from static sensors. Examples of such sensors are buses or taxis that move along the city and report their position, CCTV cameras that capture video at different locations of the city, SCATS sensors that measure the traffic flow at different junctions of the city, cell towers that measure the mobile phone users that interact with the particular cell tower and citizens that move in the city and report several anomalies.

The data streams described above contain useful information and could be used in order to quickly identify several events of emergency (*e.g.*, traffic accidents, traffic congestion, fires, floods, etc.). A typical example of a traffic con-

Figure 1: The traffic control room at Dublin City Council, where different monitors visualize the traffic conditions at different locations in the city of Dublin, captured from CCTV cameras, and allow traffic operators to monitor the traffic conditions

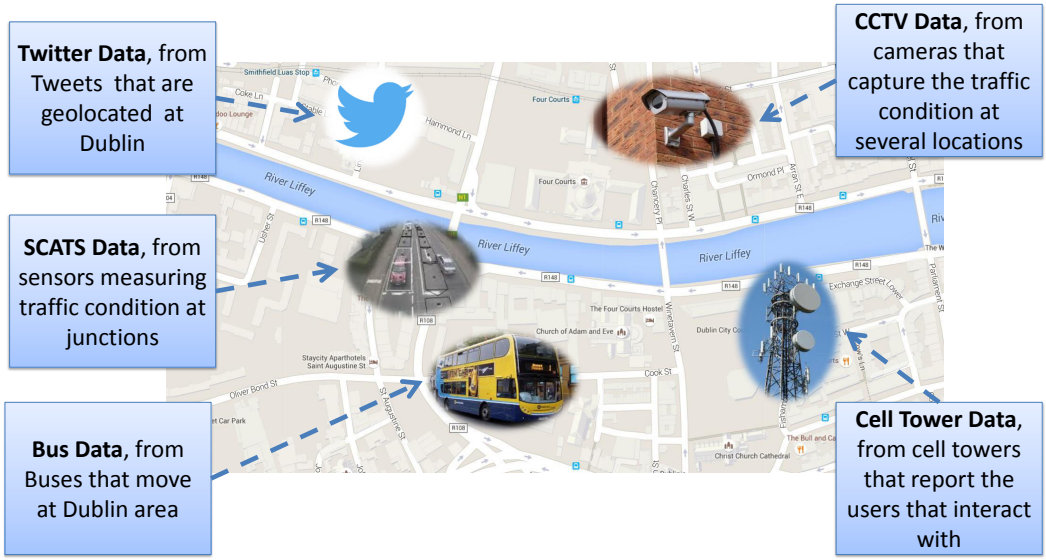


Figure 2: An example of a set of static (SCATS, Cell Tower and CCTV) and moving (Bus and Twitter) data sources generated from sensors that report continuously data to the traffic operators and need to be processed

trol room is the one from Dublin City Council presented in Figure 1 and an example of the data streams that need to be processed and analyzed is illustrated in Figure 2. The traffic control rooms often have to address the following issues: (i) lots of heterogeneous information is received and needs to be processed, (ii) the processing capability is limited thus efficient analysis techniques should be considered, and (iii) human based fusion is necessary. Thus, there is a need for algorithms and techniques in order to automatically extract information and identify patterns in these data streams, as humans are not able to monitor this information explosion. The research community in recent years focuses on the analysis, the monitoring and the event detection of urban data streams. However, there are many *restrictions* that are preventing the creation of accurate prediction models.

The main challenge that arises when data scientists analyze urban data streams is that usually these data do not contain information regarding whether an anomaly is occurring or not at a particular space and time. This lack of information prevents the building of accurate supervised learning models (either classification or regression) that would be able to automatically detect events in the city environment. Commonly this information is provided either from the traffic operators or by exploiting the knowledge of the crowd by posing tasks or queries to the citizens, using crowdsourcing techniques.

Examples of crowdsourcing tasks may be queries to the citizens regarding whether they observe traffic congestion at a particular road or whether they observe a fire at a particular address. Identifying the labels for each data point of the incoming data streams would be tremendously costly and thus unfeasible. Thus queries that request labels for particular instances of the incoming data have to be selected carefully and then posed to the human oracles or to the crowd, without wasting needlessly the available resources.

Active Learning [23] has been proven very efficient in building supervised learning models, with a limited number of labelled training data $\mathcal{L}$. Active Learning techniques are typically characterized from a cyclic procedure in which the learner: (i) carefully selects from a pool of unlabelled data the most informative instances in order to improve its performance and (ii) learns from the provided labels and leverages its new knowledge in order to decide which points to query next.

This cyclic procedure is typically not applicable in models that aim to detect events from urban data streams. In such scenarios a set of constraints exist and should be considered regarding: (i) the **limited** number of queries that could be posed to the annotators (*i.e.*, the learner should request a small number of carefully selected instances to be annotated), (ii) the **predefined** time period in which the model should be ready, as in many cases we have strict deadlines till which the prediction model should be built and (iii) the fact that it is impossible to require labels for past events (*i.e.*, ask the citizens whether a specific road was congested ten weeks ago), thus decisions regarding whether or not to require a particular label should be taken **instantly**. These issues make the existing knowledge extraction techniques not applicable in such urban data analysis scenarios.

In this work we propose a novel framework that is capable to efficiently build prediction models that are able to detect anomalous events in voluminous data streams, exploiting Active Learning techniques. Our approach takes into account the restrictions that arise in streaming urban data monitoring regarding the limited number of queries that could be asked, the fact that we could not request labels for past events and that the model should be ready within a predetermined time period. Finally, the proposed framework is able to automatically decide whether to request the label of a newly received data point setting appropriately a threshold that **dynamically** changes over time and taking into consideration the set of constraints discussed above.

We evaluated our proposed techniques in a real urban data scenario, coming from the city of Dublin. More specifically, we test our techniques building a classification model, that detects whether there is traffic or not at a particular junction

of the city. Our input stream was transmitted from SCATS sensors that are installed at different intersection of the city and measure the traffic. We annotated the SCATS data using input from images captured from CCTV cameras at the junctions of the SCATS sensors.

The main constraints that we deal with in this work are summarized bellow:

- Decide **instantly** whether to request the label for a newly received data point or not, as queries are not allowed for past events.
- Take into account the **limited** number of queries that could be posed to the data annotators, aiming to build accurate model with small budget.
- Build a classification model in a **predefined** time period.
- The created model should **converge** quickly to a batch algorithm, that is trained offline using a larger amount of labelled data points and is trained once.

The main contributions of this work are summarized to the following points:

- We propose a novel streaming Active Learning framework that is able to decide online if a new data-point received is worth labelling. The system effectively exploits the available budget, resulting rapidly to a good predictive model with a minimum number of annotations required.
- We evaluate our system and we show its usefulness under a real urban management scenario where the goal is to automatically train a model that detects traffic events from static SCATS sensors.

The rest of the paper is structured as follows. In Section 2 we discuss the recently proposed research works in urban and traffic data management, Active Learning and crowdsourcing systems. Then, on Section 3, we define (i) the framework's parameters and (ii) the problem that we solve. In Section 4 we describe our proposed framework that satisfies to the set of constraints that were discussed above. Finally, the evaluation of the proposed techniques as well as comparison with alternative techniques is presented in Section 5. The experimental results obtained from a real use case and a summary of the lessons learned from this work are discussed in Section 5.

2. RELATED WORK

Recently many research works focus in the analysis of urban data aiming to automatically detect events of interest. A traffic management system was proposed in [4] where complex events were detected monitoring heterogeneous data streams. This technique uses crowdsourcing in order to resolve possible uncertainties. The authors in [30] proposed an algorithm that extracts the mode of users transportation monitoring their raw gps trajectories. Also in [15] a novel technique that uses a hierarchical Markov model and different levels of abstraction is able to efficiently infer the user's destination or the mode of transportation. The authors in [26] proposed an interactive voting method for matching a raw and sparsely sampled GPS trajectory to roads on a digital map, using information from the road network and information extracted from the trajectory. The authors in [22] proposed a system that monitors SCATS data and a Gauss-Markov model that predicts the evolution of sensor's measurements over time is created based on the historical data. Then using Gaussian processes they estimate the current and the future conditions to the junctions where SCATS sensors are not installed. Similarly, the authors in [18] focus on mining the traffic congestion on the road network examining co-occurring congestion locations. They proposed a tree based algorithm that is able to model the traffic congestion propagation, revealing this way the vulnerabilities of the network. In [9] the authors proposed a method for identifying events in large spatiotemporal datasets and then interactively discovering other similar events using appropriately event group indexes. *Data Polygamy*, a method that aims to discover relationships between different spatio-temporal data sets was proposed in [8], where the users are able to query for statistically significant relationships between the different datasets. Furthermore, the authors in [3] proposed a method for event detection in multiple time series, developing a suite of visual analytics techniques that enables (i) transformations of the original data and (ii) investigation of the events, combining interactive visualizations on time-aware displays and maps with statistical event detection methods. In [10] the authors proposed a method that predicts the travel time for a given origin and destination pair, using methods from Queueing Theory and Machine Learning, that were trained on bus journey logs. In addition an approach that aims to improve the social welfare in a smart city with limited resources (i.e. the roads in a transportation network are characterized by a limited capacity) was presented in [16]. In this work the central agent sends in real time appropriate signals to the agents, that aim to travel across the city, reducing this way the social cost of road's congestion.

In [29] the authors proposed a transfer learning system in order to detect the optimal retail store placement using data from location-based social networks. In [13] the authors proposed a model that identifies the traffic condition monitoring small and sparsely sampled GPS data, creating a Bayesian network that reflects the road network. In [5], [20], [34] and [27] complex event and stream processing techniques have been proposed in order to monitor the traffic conditions in smart cities. Finally, in our work in [33] a novel framework was proposes that is able to detect faulty SCATS sensor measurements using multivariate ARIMA model and taking into consideration how much a sensor deviates from its usual behavior, taking into account the SCATS sensors in its neighborhood.

Active learning is an area of Machine Learning and the key idea is that the learner is able to choose particular instances that will improve its performance. The three main scenarios of Active Learning are: (i) the *membership query synthesis*, in which the learner may generate any instance belonging in the input space and request its label([2]), (ii) *selective sampling*, in which the learner samples a data point and must decide whether to query or not that data point([32, 14]), (iii) *pool based*, in which the learner queries a particular data point that belongs to the dataset([11, 25]). In monitoring urban data streams with the set of restrictions described in the previous section the most appropriate scenario is that of *selective sampling*.

In [24] the authors proposed an Active Learning framework for on-road vehicle recognition and tracking. Initially a recognition model is trained and then the *selective sampling* approach is performed querying the most informative samples. In a similar manner, the authors in [31] propose a *selective sampling* Active Learning framework suitable for high volume streams that utilizes an ensemble of learners. Instead of minimizing the ensemble accuracy they propose to minimize the ensemble variance stating the this performs better in cases of concept drift. A highly relevant system to [31] focusing on road networks is SmartRoad described in [12]. The system receives a stream of data and feeds them to a Random Forest classifier. The system similarly to ours decides about labeling an instance according to a maximum budget and the instance informativeness. As before the informativeness is proportional to the disagreement of the underlying decision trees of the random forest. The authors on [14] state the importance of covering the whole space of a large data stream and propose the use of clustering before querying. Their assumption that each cluster represents a different subspace of the stream that needs to be explored. They initially select the cluster that needs labeling and from this cluster they select the most informative instance. Finally, in [21] the authors used Gaussian Processes to deal with possible disagreements among different annotators.

Crowdsourcing could be very helpful from both the individual and societal viewpoint, as it enables the citizens in a participatory way of contributing to the society [28]. A heat transfer model that estimates the daily mean temperatures from smartphone battery temperatures, exploiting crowdsourcing is presented in [19]. The project that was developed from U.S. Department of Homeland Security (DHS) aims to detect environmental threads exploiting the power of crowd [17]. They equipped mobile phones with chemical-agent detectors and then through security networks the retrieved data were transmitted in order to automatically detect and mitigate threats to urban populations. In our previous work we implemented a mobile application, called CrowdAlert [1], that enables human users to annotate real-world events and we have developed task assignment approaches for crowdsourcing environments that consider user reliability and real-time constraints [6, 7]. This work exploits the feedback, received either from traffic operators or from the citizens, in order to develop an event detection framework that is based on active learning.

3. PROBLEM DEFINITION

We formulate our problem in an Active Learning setting while taking into account the specific constraints that come from the nature of our data. We note that in our problem, a query on a specific data point has to be answered by a human, potentially using a crowdsourcing mechanism in real-time since it is an assessment of the traffic situation at a specific point and time. This creates additional constraints on the total number of queries that can be asked, as well as the rate at which queries can be asked.

The set of parameters that were considered in this work are presented in Table 1. More formally our problem can be formulated in detail as follows:

Given:

- *a set $\mathcal{L}$ of $N_{\mathcal{L}}$ labelled data points, where for each data point $x \in \mathbb{R}^D$ the corresponding label y is provided ($y \in \mathbb{R}$ or $y \in \mathbb{D}$ in case of regression or classification respectively)*
- *a set $\mathcal{U}$ of $N_{\mathcal{U}}$ unlabelled data points, much larger than the labelled set ($N_{\mathcal{U}} >> N_{\mathcal{L}}$), where again each data point $x \in \mathbb{R}^D$*
- *an infinite data stream $\mathcal{S}$ of data points $x \in \mathbb{R}^D$*
- *a time period $\mathcal{T}$ in which the prediction model should have been created*
- *a limited budget of $\mathcal{B}$ queries that the system is available to pose to the annotators*
- *the time t_{slot} required from an annotator to perform the annotation*

Parameter	Description
$\mathcal{L}$	The set of labelled data
$\mathcal{U}$	The set of unlabelled data
$\mathcal{S}$	The infinite stream of data that arrives
$\mathcal{T}$	The time period, in which the model should have been created
$\mathcal{B}$	The number of available questions that is allowed to ask the annotators in order to provide labels
t_{slot}	The time slot that is required from the annotator to provide a label for a particular data point

Table 1: Parameters definitions

Our goal is to create a framework that uses the given budget $\mathcal{B}$ and build a robust supervised learning model in the given period $\mathcal{T}$.

4. OUR METHOD

In this section we describe in detail our proposed Active Learning framework that aims to create an accurate supervised learning model minimizing the human effort and consequently the wasted resources. Our techniques consider the system's constraints described in Section 3.

Initially our framework trains the supervised learning model $\mathcal{M}$ using the labelled dataset L, estimating the model's parameters θ. A wide range of supervised learning models are supported from our framework including Naive Bayes, Support Vector Machine (SVM), Neural Networks and others. In this work we focus, without loss of generality, on classification models that support the probability estimation for a new instance x^* belonging to each class, $P(class = C_i|x^*, \theta)$. Other classification or regression models could be easily supported from our framework using a different query strategy.

An informativeness function $informativeness(x^*, \theta)$ is defined. This function estimates how much informative the knowledge of the label of a particular data point would be for the prediction model $\mathcal{M}$. This function receives as input a new data point x^* and the model's parameters θ and estimates the informativeness of x^*. The most commonly used technique for selecting the informativeness of new instances is the Uncertainly Sampling technique, that favours to select those instances that are closest to the boundaries of the model $\mathcal{M}$. Thus, we decided to use an informativeness function that favors to query the data points for which the

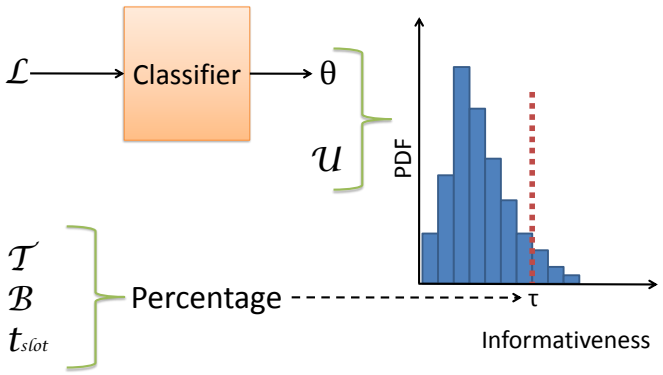


Figure 3: Description of the Active Learning framework learning procedure: The labeled instances $\mathcal{L}$ are used in order to learn the model parameters θ and then using the unlabeled dataset the PDF of each data point in the unlabeled dataset $\mathcal{U}$ is identified, finally the threshold τ of informativeness is calculated using the framework's constraints $\mathcal{T}$, $\mathcal{B}$, t_{slot}

model is least confident. These points are calculated from Equations 1 and 2, which are presented below:

$$x_{LC} = \arg\max_{x} 1 - P_\theta(\hat{y}|x) \quad (1)$$

$$\hat{y} = \arg\max_{y} P_\theta(y|x) \quad (2)$$

When a model aims to learn from a limited set of training data $\mathcal{L}$ and a voluminous data stream $\mathcal{S}$ that arrive sequentially and labels could be assigned only to new instances and not to past events, only the *selective sampling* scenario of Active Learning can be applied. The *pool based* scenario can not be applied as learner don't know in advance the labels of the streaming data that are going to be received in the future. The *membership query synthesis* scenario is not suitable as it would be impossible for the learners to identify the label of multidimensional instances that is not directly human interpretable. In our scenarios the model should automatically decide whether to require the label for the newly received data point. A naive solution in such a case would be to randomly select a set of instances in the streaming data and provide labels for those data points ignoring their informativeness and then use these instances in order to improve the classifiers performance. Another solution would be to set a *static* threshold on the informativeness measurement that remains the same across the learning period. In this case if the threshold is high a small number of instances will be requested in the time period. On the other hand if the threshold is low then a large number of meaningless queries will be posed to the users. Thus a model that is able to *dynamically* adjust the informativeness threshold is required.

Our proposed approach *dynamically* adjusts the threshold exploiting the knowledge provided from the unlabelled dataset $\mathcal{U}$ and the set of constraints $\mathcal{T}$, $\mathcal{B}$ and t_{slot}. More specifically in order to set up the threshold we evaluate the informativeness for every point $x \in \mathcal{U}$ every time that the model $\mathcal{M}$ is updated, as it is illustrated in Figure 3. Then

Algorithm 1 Dynamic Active Learning

1: **Input:** $\mathcal{L}$, $\mathcal{U}$, $\mathcal{S}$, $\mathcal{T}$, $\mathcal{B}$, t_{slot}, $classifier$
2: **Output:** θ
3: $\theta \leftarrow classifier(X_\mathcal{L}, Y_\mathcal{L})$
4: $informativeness_\mathcal{U} \leftarrow calcInformativeness(X_\mathcal{U}, \theta)$
5: $\tau \leftarrow findThreshold(informativeness_\mathcal{U}, t_{slot}, \mathcal{T}, \mathcal{B})$
6: **while** $currentTime < T$ **do**
7: $\quad x_\mathcal{S} \leftarrow getNext(\mathcal{S})$
8: $\quad inf_x \leftarrow calcInformativeness(x_\mathcal{S}, \theta)$
9: $\quad$ **if** $inf_x > \tau$ **then**
10: $\quad\quad y_\mathcal{S} \leftarrow annotate(x_\mathcal{S})$
11: $\quad\quad [X_\mathcal{L}, Y_\mathcal{L}].insert([x_\mathcal{S}, y_\mathcal{S}])$
12: $\quad\quad \theta \leftarrow classifier(X_\mathcal{L}, Y_\mathcal{L})$
13: $\quad\quad informativeness_\mathcal{U} \leftarrow calcInf(X_\mathcal{U}, \theta)$
14: $\quad\quad \tau \leftarrow findThreshold(informativeness_\mathcal{U}, t_{slot}, \mathcal{T}, \mathcal{B})$
15: $\quad\quad \mathcal{B} \leftarrow \mathcal{B} - 1$
16: **return** θ

the calculated informativeness measurements are used in order to create a histogram that illustrates the Probability Density Function (PDF) of informativeness. In order to appropriately set the threshold we take into account the current budget $\mathcal{B}$, the remaining time required in order to build the model T and the time t_{slot} needed to annotate each data point. More specifically we calculate the *percentage* of points that should be asked calculating (i) the total number of queries that could be posed to the user, calculated in Equation 3 and (ii) the percentage as the fraction of the budget of the remaining queries and the total number of queries, estimated in Equation 4. Finally the threshold is calculated as the value of informativeness that splits the histogram in a way that $N_\mathcal{U} \times (1 - percentage)$ points are on the left of the threshold while $N_\mathcal{U} \times percentage$ points are on the right of the threshold. Therefore the threshold considers the knowledge of the historical unlabelled data and the provided restrictions. Finally, it should be mentioned that the parameters $\mathcal{B}$ and $\mathcal{T}$ are updated over time as queries are posed to the users and the time passes, respectively.

$$total = \frac{\mathcal{T}}{t_{slot}} \quad (3)$$

$$percentage = \frac{\mathcal{B}}{total} \quad (4)$$

The steps of the proposed framework are described in Algorithm 1. In line 3 the model $\mathcal{M}$ with parameters θ is calculated using the initial data points $\mathcal{M}$. In lines 5, 6 we calculate informativeness threshold using Equations 1 and 2 on the unlabelled dataset $\mathcal{M}$. Finally the Active Learning cyclic learning procedure is shown in lines $6 - 15$. When the informativeness of the streaming instances exceed the dynamically identified threshold τ (line 9) then queries are posed in the annotators (line 10). Then the labelled dataset is updated with the new data point and its label (line 11). Finally, in lines $12 - 14$ the model's parameters θ and the informativeness threshold are updated.

5. EVALUATION

We performed an extensive experimental evaluation of our proposed framework using a real world scenario with streaming urban data from Dublin City. More specifically, we

evaluated our approach with a traffic anomaly event detection application applied on the SCATS data that are transmitted sequentially from the SCATS sensors to the DCC. The SCATS sensors are located at different junctions across Dublin and sequentially report a variety of informative measurements for the traffic condition of the road. Example measurements include the traffic flow and the degree of saturation of the different lanes in the junction. Setting static thresholds on the measurements (e.g. a static threshold on the degree of saturation) is not an effective approach for detecting traffic events. Examples of such miss-informative situations occur when a car is parked on top of a SCATS sensor, or when particular lanes of the road are closed for maintenance, then the default static thresholds will report falsely traffic events when the road is empty or no traffic will be reported falsely.

The main issue that we had to deal with, in order to perform our evaluation, was that the SCATS data data don't contain labels regarding the existence of traffic events. This information is necessary to build models that are able to monitor these data streams and automatically detect events. In order to resolve this issue and evaluate our technique we provided labels for the SCATS data using images captured from a particular CCTV cameras that captured the SCATS sensors. Dublin City Council (DCC) uploads periodically the images of several CCTV cameras that are distributed in different locations of Dublin city[1]. We annotated some of these images provided from a particular CCTV camera and use the annotations for labelling the sub-stream of the SCATS data that corresponds to the junction covered by the camera. In the rest of this section we describe the approach that we followed in order to generate the different datasets $\mathcal{L}$, $\mathcal{S}$ and $\mathcal{U}$, as well as as the performance of our approach compared against a technique which selects random instances for annotation with a various set of constraints.

5.1 Generating the Datasets

In this section we describe how we integrated the SCATS data and the corresponding labels from the CCTV cameras, in order to generate the datasets $\mathcal{L}$, $\mathcal{S}$ and $\mathcal{U}$ that our framework requires. Initially we selected appropriately a camera that is able to capture suitably a SCATS sensor, deciding to monitor the SCATS sensor 81 and the CCTV camera 31, that their locations are presented in Figure 4. Several images captured from the camera are illustrated in Figure 5. Figures 5a and 5b were annotated as *Traffic* events, while Figures 5c and 5d were labeled as *No Traffic* events.

Initially we created a crawler that downloaded periodically the uploaded images from the different CCTV cameras. Then in order to easily and efficiently provide labels for the images of the CCTV camera 31 we created a web interface, illustrated in Figure 6. The web interface receives as input the id of the camera and a specified time period and returns as output all the images that were downloaded for that camera for this time period. Then the annotator provides labels for each image using his computer's mouse (left click → No Traffic, right click → Traffic). When the user provides an input for a particular image a tuple is stored in a MongoDB database that contains the id of the CCTV camera, the image timestamp and the provided label (*Traffic* or *No Traffic*). Finally the last step was to join the provided labels with the SCATS data. In order to do this for every

[1]https://www.dublincity.ie/dublintraffic/

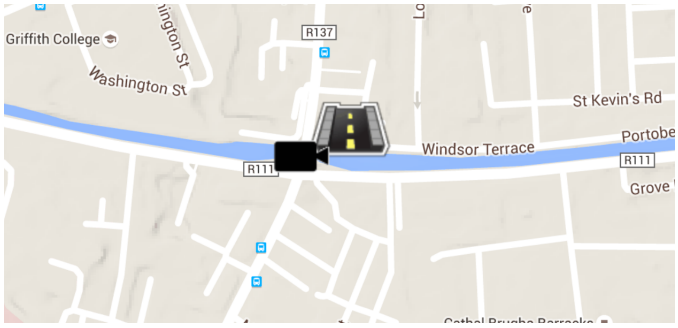


Figure 4: The locations of the camera and the SCATS sensor that we decided to monitor

provided annotation for the CCTV camera 31 we identified the SCATS measurements from sensor 81 with the closest timestamp. Then the set of features x were the values of degree of saturation and traffic flow of the different lanes of the junction while the label y was provided from the annotation. This procedure generated a dataset $\mathcal{D}$ with $2,386$ labeled SCATS data. The unlabelled dataset $\mathcal{U}$ consisted of $50,000$ data points with the same features as $\mathcal{D}$ but without a label y. The two datasets $\mathcal{D}$ and $\mathcal{U}$ are chronologically disjoint.

We performed our experiments measuring the accuracy for a particular budget of available queries $\mathcal{B}$ and different number of initial data points $initDP$, used in order to create our initial model. In order to create the dataset $\mathcal{L}$ and the stream of data points $\mathcal{S}$ we selected to use the first $initDP$ data points of the set $\mathcal{D}$ in order to create the labeled dataset $\mathcal{L}$. The data stream $\mathcal{S}$ is created using the rest $|\mathcal{D}| - initDP$ points. Then in order to evaluate the performance of our framework we performed 5-fold cross validation in the sequentially transmitted data stream $\mathcal{S}$. This set contains the labels for each data point and when the learner requested a label y for a particular instance of $\mathcal{S}$ that can be easily retrieved.

Finally, we selected to perform our binary event detection classification using a SVM classifier, with a polynomial kernel. The SVM implementation that we used provided the class membership probabilities that are required in Equations 1 and 2, in order to calculate the informativeness.

5.2 Dynamic Vs Random Selection

In Figure 7 we illustrate the histogram of informativeness for the unlabeled data. As it can be observed, informativeness takes values between 0 and 0.5. This is due to the fact that we perform binary classification. The probability of belonging to each class is given from Equation 2. The most informative points are those for which the classifier is least confident.

We compared our technique with another approach, named *Random* selection, which selects the instances that will be annotated randomly, without taking into account their informativeness. We measured the accuracy of our technique, noted as *Dynamic* compared to the *Random* with the following values of initial labelled data points [10, 25, 50, 100, 200, 500] and for the following values of available queries' budget [1, 3, 5, 10, 25, 50, 100, 150, 200, 500]. Our results are presented in Figure 8.

From the different results illustrated in Figure 8 we can

(a) Label: *Traffic* (b) Label: *Traffic* (c) Label: *No Traffic* (d) Label: *No Traffic*

Figure 5: Screenshots from different hours of the day and different days from the camera with ID: 31, Figures (a) and (b) represent traffic events at the intersection, while Figures (c) and (d) represent the normal (no traffic) behavior. Our model aims to detect these labels monitoring the SCATS data.

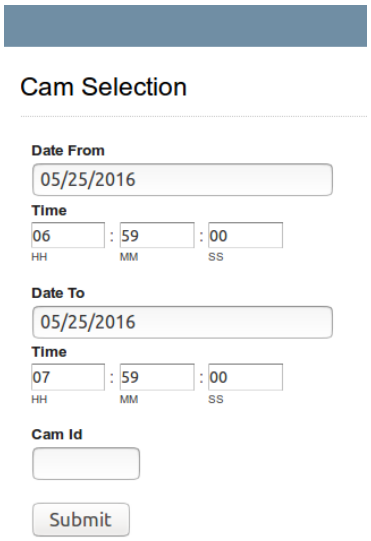


Figure 6: The web interface that helps the human annotators to easily select the appropriate time period for a particular camera ID in order to annotate the respective images

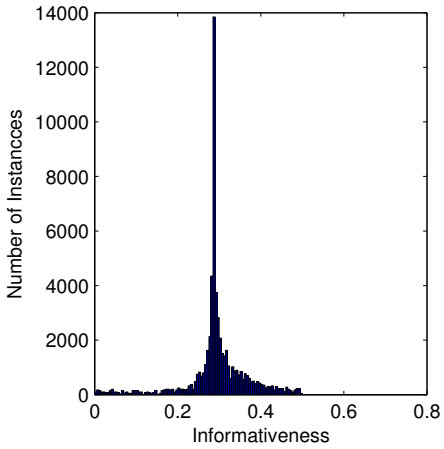


Figure 7: The histogram of informativeness measurement for the set of unlabeled data points

Inititial Data Points	Accuracy
10	0.51
25	0.71
50	0.72
100	0.72
200	0.76
500	0.75

Table 2: The accuracy of the SVM classifier for different number of initial data points and without querying the annotators ($\mathcal{B} = 0$)

observe that our *Dynamic* selection of querying points outperforms the *Random* selection in terms of accuracy. This is explained by the fact that our method selects for labelling the instances with the highest informativeness score, setting appropriately the dynamic informativeness threshold. This threshold is dynamically updated over time in order to ensure that queries are posed to the users during the whole learning period $\mathcal{T}$, without frivolously wasting the budget $\mathcal{B}$.

Also we observe in all the plots that the greater the budget $\mathcal{B}$ the more accurate the classifier becomes, for both approaches. Thus applying Active Learning on urban data analysis scenarios is extremely helpful and creates robust models even with small amounts of labeled data points. Furthermore, we observe in Table 2 that the classifier with 10 initial data points in the labeled dataset $\mathcal{L}$ has much lower accuracy compared to the rest of the experiments, when no budget is available ($\mathcal{B} = 0$). However, having more than 25 initial datapoints barely increases the accuracy.

Another observation is that the classifiers that are initialized with smaller labeled initial datasets $\mathcal{L}$ have greater learning rate compared to the those that start with larger labelled datasets. This could be explained by the fact that the initial labelled data points are randomly selected and a large proportion of these dataset may not be informative for the model $\mathcal{M}$. However, this observation suggests that even if we start with small labeled dataset we can quickly improve it and end to a robust and accurate classification model.

In addition when the classifier starts with 10 labeled data points and the given budget is set to 25 (35 annotated data points in total), Figure 8a, our method's performance is equal to using 200 initial data points and zero budget (ac-

curacy around 0.75 as you can see in Table 2). Also the performance achieved with those 35 annotated data points outperforms that of models trained with larger number of labeled instances (*i.e.*, 50, 100 and 500 initial points and zero budget). This suggests that we can build accurate classifiers with limited number of questions if we can carefully select which instances to query. Also it should be clear that we are able to build robust classification models that their accuracy converges quickly to the performance of batch approaches that use much more training examples and without the option to require labels for new data points.

6. CONCLUSION

In this work we proposed a novel Active Learning framework that is able to efficiently select the most informative instances from a set of sequentially received streaming data in order to be annotated. As presented in the evaluation of our technique our *dynamical* threshold outperformed the random selection approach leading to more accurate prediction models. This framework could be extremely beneficial for the analysis of urban data where the set of posed constraints are similar with those considered in this work. Finally, in our experimental evaluation we focused on a real world scenario, creating models that detect traffic events from SCATS data streams at the city of Dublin. In our future work we plan to examine the efficiency of our approach when human users annotate traffic events in real-time using our CrowdAlert application.

Acknowledgments. This research has been financed by the European Union through the FP7 ERC IDEAS 308019 NGHCS project, the Horizon2020 688380 VaVeL project and a Yahoo Faculty award.

7. REFERENCES

[1] Crowdalert. http://crowdalert.aueb.gr/.

[2] I. M. Alabdulmohsin, X. Gao, and X. Zhang. Efficient active learning of halfspaces via query synthesis. In *AAAI*, pages 2483–2489, 2015.

[3] G. Andrienko, N. Andrienko, M. Mladenov, M. Mock, and C. Poelitz. Extracting events from spatial time series. In *2010 14th International Conference Information Visualisation*, pages 48–53. IEEE, 2010.

[4] A. Artikis, M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, V. Kalogeraki, J. Marecek, et al. Heterogeneous stream processing and crowdsourcing for urban traffic management. In *EDBT*, pages 712–723, 2014.

[5] A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. Koutsopoulos, and C. Moran. Ibm infosphere streams for scalable, real-time, intelligent transportation services. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1093–1104. ACM, 2010.

[6] I. Boutsis and V. Kalogeraki. Crowdsourcing under real-time constraints. In *IPDPS*, pages 753–764, Boston, MA, May 2013.

[7] I. Boutsis and V. Kalogeraki. On task assignment for real-time reliable crowdsourcing. In *ICDCS*, pages 1–10, Madrid, Spain, June 2014.

[8] F. Chirigati, H. Doraiswamy, T. Damoulas, and J. Freire. Data polygamy : the many-many relationships among urban spatio-temporal data sets. In *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data (SIGMOD '16)*, 2016.

[9] H. Doraiswamy, N. Ferreira, T. Damoulas, J. Freire, and C. T. Silva. Using topological analysis to support event-guided exploration in urban data. *IEEE transactions on visualization and computer graphics*, 20(12):2634–2643, 2014.

[10] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich. Traveling time prediction in scheduled transportation with journey segments. *Information Systems*, 2015.

[11] S. C. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, pages 633–642. ACM, 2006.

[12] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher. Smartroad: a crowd-sourced traffic regulator detection and identification system. In *Information Processing in Sensor Networks (IPSN), 2013 ACM/IEEE International Conference on*, pages 331–332. IEEE, 2013.

[13] T. Hunter, R. Herring, P. Abbeel, and A. Bayen. Path and travel time inference from gps probe vehicle data. *NIPS Analyzing Networks and Learning with Graphs*, 12(1), 2009.

[14] D. Ienco, A. Bifet, I. Žliobaitė, and B. Pfahringer. Clustering based active learning for evolving data streams. In *Discovery Science*, pages 79–93. Springer, 2013.

[15] L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5):311–331, 2007.

[16] J. Mareček, R. Shorten, and J. Y. Yu. Signaling and obfuscation for congestion control. *Int. J. Control*, 88(10):2086–2096, 2015.

[17] T. Monahan and J. T. Mokos. Crowdsourcing urban surveillance: The development of homeland security markets for environmental sensor networks. *Geoforum*, 49:279–288, 2013.

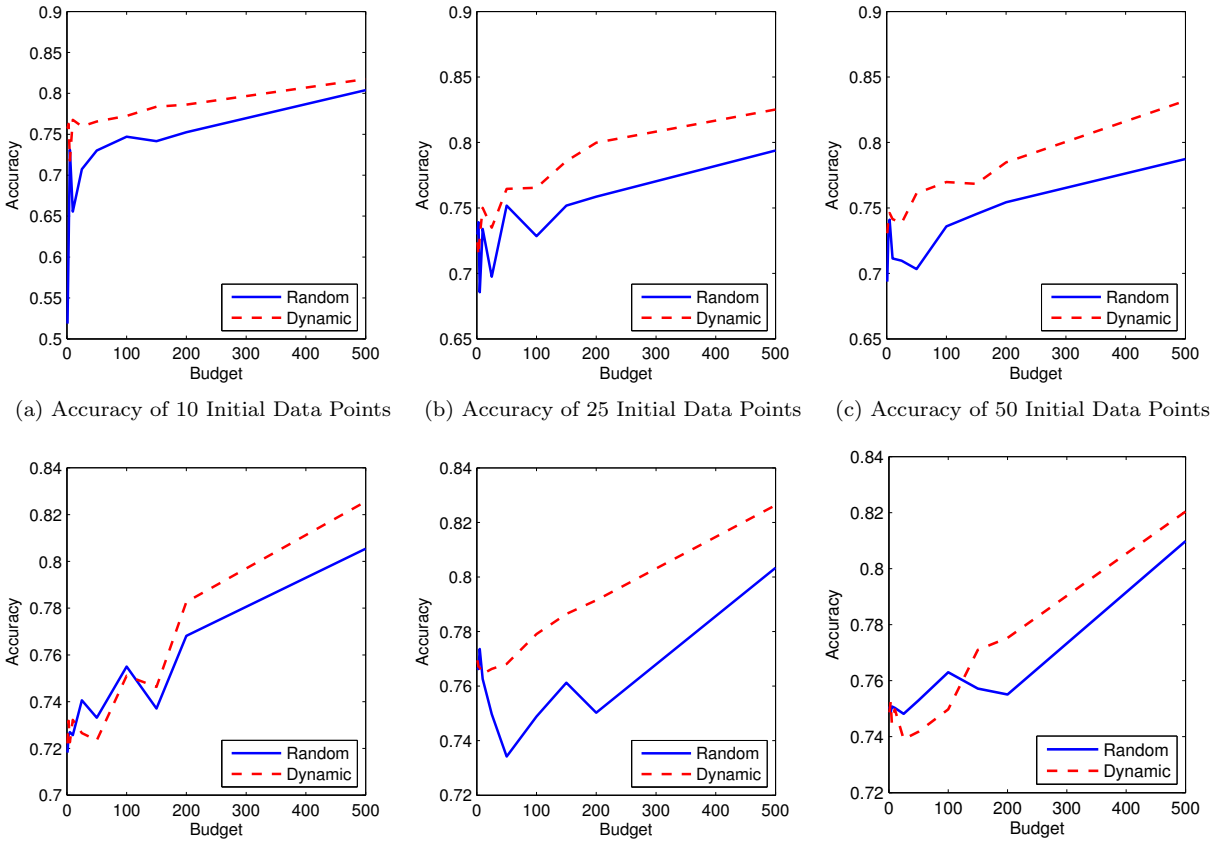
[18] H. Nguyen, W. Liu, and F. Chen. Discovering congestion propagation patterns in spatio-temporal traffic data.

[19] A. Overeem, J. R Robinson, H. Leijnse, G.-J. Steeneveld, B. P Horn, and R. Uijlenhoet. Crowdsourcing urban air temperatures from smartphone battery temperatures. *Geophysical Research Letters*, 40(15):4081–4085, 2013.

[20] K. Patroumpas and T. Sellis. Event processing and real-time monitoring over streaming traffic data. In *Web and Wireless Geographical Information Systems*, pages 116–133. Springer, 2012.

[21] F. Rodrigues, F. Pereira, and B. Ribeiro. Gaussian process classification and active learning with multiple annotators. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 433–441, 2014.

[22] F. Schnitzler, T. Liebig, S. Mannor, and K. Morik. Combining a gauss-markov model and gaussian process for traffic prediction in dublin city center. In *EDBT/ICDT Workshops*, pages 373–374, 2014.



(a) Accuracy of 10 Initial Data Points (b) Accuracy of 25 Initial Data Points (c) Accuracy of 50 Initial Data Points

(d) Accuracy of 100 Initial Data Points (e) Accuracy of 200 Initial Data Points (f) Accuracy of 500 Initial Data Points

Figure 8: Accuracy for different number of initial data points and for different values of available budget $\mathcal{B}$

[23] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[24] S. Sivaraman and M. M. Trivedi. A general active-learning framework for on-road vehicle recognition and tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):267–276, 2010.

[25] M. Sugiyama and S. Nakajima. Pool-based active learning in approximate linear regression. *Machine Learning*, 75(3):249–274, 2009.

[26] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun. An interactive-voting based map matching algorithm. In *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management*, pages 43–52. IEEE Computer Society, 2010.

[27] N. Zacheilas, V. Kalogeraki, N. Zygouras, N. Panagiotou, and D. Gunopulos. Elastic complex event processing exploiting prediction. In *Big Data (Big Data), 2015 IEEE International Conference*, pages 213–222. IEEE, 2015.

[28] F. Zambonelli. Pervasive urban crowdsourcing: Visions and challenges. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 578–583. IEEE, 2011.

[29] N. Zhang, H. Chen, and X. Chen. Transfer learning for urban computing: A case study for optimal retail store placement.

[30] Y. Zheng, L. Liu, L. Wang, and X. Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *Proceedings of the 17th international conference on World Wide Web*, pages 247–256. ACM, 2008.

[31] X. Zhu, P. Zhang, X. Lin, and Y. Shi. Active learning from stream data using optimal weight classifier ensemble. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(6):1607–1621, 2010.

[32] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(1):27–39, 2014.

[33] N. Zygouras, N. Panagiotou, N. Zacheilas, I. Boutsis, V. Kalogeraki, I. Katakis, and D. Gunopulos. Towards detection of faulty traffic sensors in real-time. In *Proceedings of the 2nd International Workshop on Mining Urban Data co-located with 32nd International Conference on Machine Learning (ICML 2015), Lille, France, July 11th, 2015.*, pages 53–62, 2015.

[34] N. Zygouras, N. Zacheilas, V. Kalogeraki, D. Kinane, and D. Gunopulos. Insights on a scalable and dynamic traffic management system. In *EDBT, Brussels, Belgium*, 2015.