Soft Sensing in Smart Cities: Handling 3Vs Using Recommender Systems, Machine Intelligence, and Data Analytics

Hadi Habibzadeh, Andrew Boggio-Dandry, Zhou Qin, Tolga Soyata, Burak Kantarci, and Hussein T. Mouftah

Today's existing smart city research involves many overtly futuristic applications that require a data acquisition structure which gathers its data from a countless number of sensors. At the core of this big data infrastructure lie the 5Vs: veracity, volume, velocity, variety, and value. Despite its seemingly insurmountable size, the acquired data is highly redundant, and systematic use of machine intelligence and data analytics can facilitate processing by extracting only the relevant information.

Abstract

Today's existing smart city research involves many overtly futuristic applications such as smart transportation, in which smart roads warn drivers of bad traffic conditions ahead, smart parking, which communicates the location of unoccupied parking spaces to drivers, and smart environment, which enables fully automated homes and workplaces to adjust their temperature to conserve energy. The realization of these applications hinges on a data acquisition structure that gathers its data from a countless number of sensors, either deployed for predefined tasks (hard sensing) or built into the mobile devices of smart city residents (soft sensing). At the core of this big data infrastructure lie the 5Vs: veracity, volume, velocity, variety, and value. The soft sensing component of a smart city sensing network is particularly affected by 3Vs: veracity, volume, and velocity. To address the unique challenges of big data, recommender systems, statistical reputation systems, and context analysis are used to ensure the veracity of acquired data, machine learning algorithms are applied to handle the data volume, and data analytics algorithms are implemented to manage data velocity. Despite its seemingly insurmountable size, the acquired data is highly redundant, and systematic use of machine intelligence and data analytics can facilitate processing by extracting only the relevant information; in this article, we study the role of these algorithms through the lens of the 3Vs in facilitating soft sensing within the framework of smart city applications.

Introduction

Smart city applications are built on an infrastructure, as depicted in Fig. 1. The *sensing* component enables data acquisition from numerous sensors deployed throughout the city (e.g., traffic cameras and speed sensors) or built-in sensors in residents' smartphones (e.g., GPS and accelerometers in a crowdsensing setting), both of which transmit the acquired data to the cloud for further processing. The *processing* component utilizes machine intelligence algorithms to make real-time intelligent decisions (e.g., detecting traffic accidents), while data analytics algorithms allow the city to extract valuable statistical correlations (e.g., identifying the most likely cause of traffic accidents) in the long term. Finally, the *publishing and control* component facilitates cooperation between human decisions (e.g., city operators) and machine algorithms. This article investigates some specific cases of smart city applications of machine intelligence and data analytics.

Central to the operation of every smart city application is a data collection network of Internet of Things (IoT) sensors [1], either built on traditional fixed-location dedicated sensors (e.g., cameras) or sensors built in to the mobile devices of city residents who volunteer in crowdsensing platforms (e.g., the mobile phone GPS and accelerometer use in pothole detection [2]). These two sensing platforms are termed *hard sensing* and *soft sensing*, respectively. Regardless of the sensing platform, the amount of data generated far exceeds the current processing capabilities of its eventual destination — the cloud; this "big data problem" is characterized by the 5Vs: volume, variety, velocity, veracity, and value. Of these, this article investigates the 3Vs that have the highest impact on soft sensing: veracity, volume, and velocity.

To address veracity, the trustworthiness of the acquired data is analyzed. Outlier detection techniques are implemented to improve the reliability/validity of sensed data. Decentralized recommender systems are utilized to filter outliers before any statistical analysis or data fusion is performed. Statistical reputation systems are employed in order to establish the trustworthiness of individual users in participatory sensing, and to detect fraud and misinformation. Context analysis is used to detect and prevent the injection of sensor data resulting from spoofed identity.

To handle the processing of high-volume data, the use of machine intelligence algorithms is investigated. As the data collection network transmits information to the cloud, machine intelligence algorithms make decisions and predictions based on the data, eliminating data redundancy and significantly reducing the amount of information to be processed. These algorithms can operate in real time, with a large portion of the preprocessing being run locally on the sensors themselves.

Digital Object Identifier:
10.1109/MCOM.2018.1700304

Hadi Habibzadeh, Andrew Boggio-Dandry, and Tolga Soyata are with SUNY Albany; Zhou Qin is with the National University of Defense Technology; Burak Kantarci and Hussein T. Mouftah are with the University of Ottawa.

0163-6804/18/$25.00 © 2018 IEEE

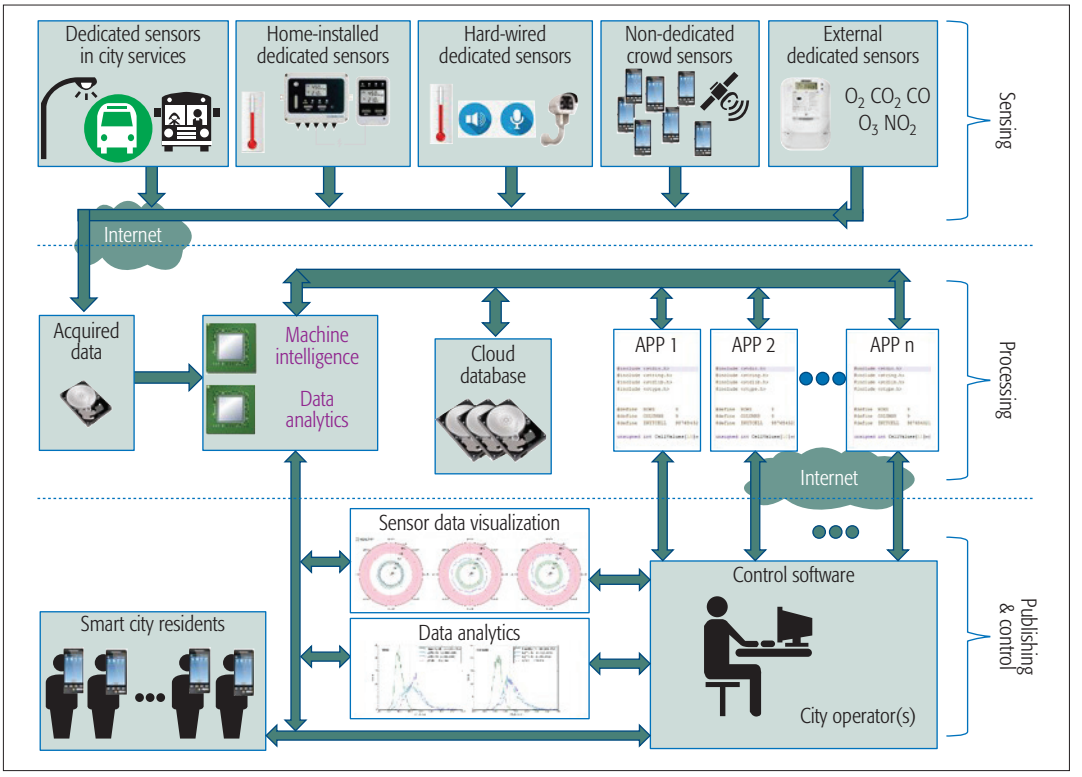

Figure 1. Components of a smart city. The *sensing* component is composed of a wired or wireless distributed sensor network, which acquires data through dedicated and non-dedicated sensors and transmits it to the cloud. The *processing* component houses a set of smart city applications, stores raw or processed data, and monitors the conflicts among different applications using machine intelligence. The *publishing and control* component is responsible for providing a visualization of the acquired sensor data in addition to analytics about that data. This component is controlled by one or more city operators that can adjust the operational parameters of different software components.

To address the issue of data arriving at high velocity, the application of data analytics algorithms is studied. Once information is stored in the cloud, data analytics algorithms can be used to extract valuable statistical information (e.g., causes of traffic congestion based on vehicle trajectories from nearby roads, optimized using *k*-nearest neighbor, kNN [3]). This information can then be used to optimize smart city operations (e.g., adjusting traffic light sequences to help alleviate congestion).

Smart City Soft Sensing

An IoT-big data ecosystem in a smart city setting acquires data via distributed sensors that can be uniquely identified, localized, and communicated with. The IoT denotes the interconnection of sensors, RFID tags, smartphones, and other objects in a scalable manner [1]. In such an ecosystem, *hard sensing* (or, alternatively, *dedicated sensing*) is the primary sensing paradigm in many smart city applications, as it can be tailored to precisely meet the application requirements. Typical hard sensing systems consist of sensing stations that measure certain pre-defined parameters. The distinguishing characteristic of dedicated sensing is that architectural components (sensing, processing, publishing, and control, as shown in Fig. 1) are either owned or leased by the system administrator. On the other hand, *soft sensing* (or, alternatively, *non-dedicated sensing*) is an essential component of a smart city, which calls for distributed approaches to ensure veracity and efficiency.

Soft sensing includes various non-dedicated sensing paradigms such as opportunistic sensing, participatory sensing (i.e., crowdsensing), and social sensing where citizens serve as sensing nodes. The number of connected mobile devices equipped with built-in sensors is predicted to exceed half of the world's population by the end of 2018 [4]. Soft sensing calls for novel methods for storage, management, and processing of the sensed data by using predictive analytics, data mining, text analytics, and statistical analysis. GPS, camera, accelerometer, gyroscope, and microphone are among the most common built-in sensors; widespread use of these devices signals their future potential for being an integral part of IoT-based sensing in smart cities.

Soft Sensing Veracity: Data Usefulness and Trustworthiness

Heterogeneity of sensors and sensing platforms introduces the problem of usefulness and trustworthiness of sensed data. We start with the veracity aspect because data trustworthiness needs to be handled at the data acquisition stage prior to mining or analysis of the data. Data trustworthiness can be addressed via three methodologies:

- Applying reputation systems to detect fraud and misinformation
- Applying recommender systems to filter malfunctioning sensors
- Applying anomaly detection schemes to cope with misuse of devices

Soft-sensing calls for novel methods for storage, management, and processing of the sensed data by using predictive analytics, data mining, text analytics, and statistical analysis. GPS, camera, accelerometer, gyroscope, and microphone are among the most common built-in sensors; widespread use of these devices signals their future potential for being an integral part of the IoT-based sensing in smart cities.

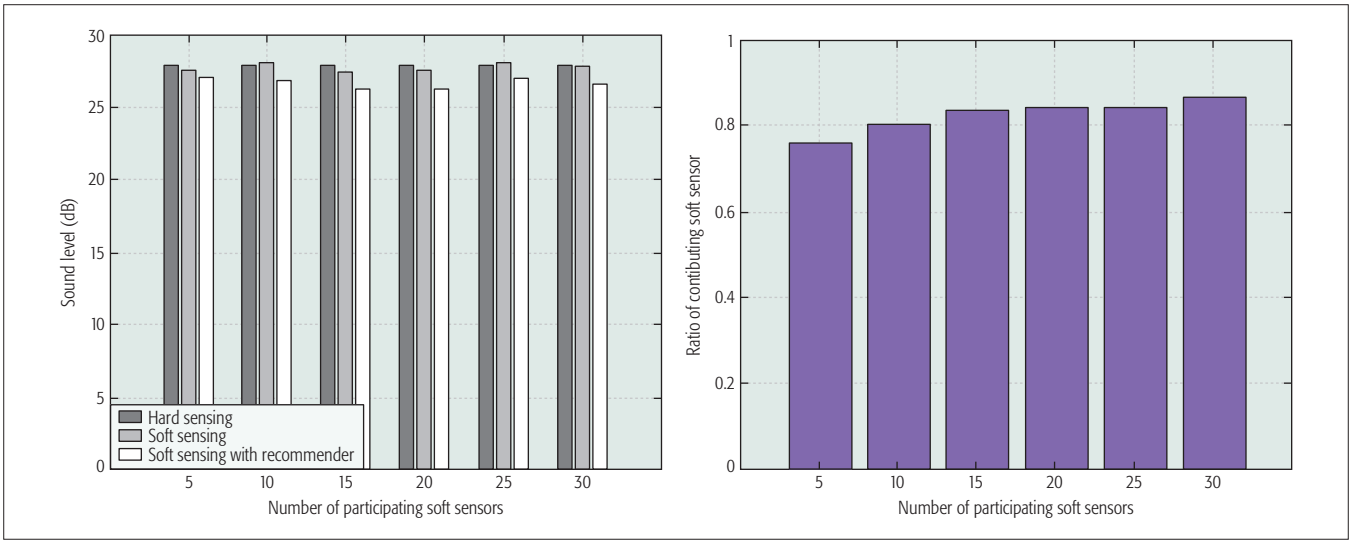

Figure 2. A simple feasibility study of recommender systems in soft sensing: (left) sound level reported by a hard sensor, sound level sensed by soft sensors, and sound level reported by soft sensors with a recommender system; (right) number of participating sensors vs. number of contributing sensors when a recommender system is used. Soft sensing with and without a recommender system leads to an error between 2 and 4 percent. A simple recommender system can achieve the same level of veracity with 14–24 percent fewer soft sensors.

Fraud and Misinformation in Soft Sensing

Outlier detection techniques are also used to improve the reliability/validity of sensed data. In soft sensing, each sensing request — submitted to non-dedicated sensors — has a value for the requesting platform. The entirety of the sensed data by all participants of this platform corresponds to the "value" of the received data (or, more generally, information). This value is quantified in [4] on a discrete scale for all sensing tasks by considering the average reputation of the participants.

Veracity of Soft Sensing in Participatory Contexts

Soft sensor veracity may be impacted by low-quality data, sometimes caused by malicious behavior. Low veracity of soft sensor data is not always caused by malicious behavior; inaccurate readings resulting from sensor malfunction or other environmental factors, such as interference, can also affect veracity. While statistical analysis to obtain the reputation of soft sensors can reveal malicious behavior, decentralized recommender systems can also help filter outliers before they undergo any statistical analysis or data fusion [4].

Since individual soft sensors may not always be available (or, in the case of participatory sensing, users of devices with built-in sensors need to be compensated/awarded for participation), a recommender system can eliminate some participants while ensuring the same level of accuracy as a soft sensing network where all participants contribute. In a simple recommender system the fundamental assumption is that each sensor hears the values read by other sensors, and then casts a vote for every other soft sensor in the vicinity. If the similarity score of sensor x is within a Δ interval, a +1 vote is cast for the sensor; otherwise, a −1 vote is cast. The sum of the received votes for each sensor is normalized by the total number of participating soft sensors, and a hypothetical veracity value is obtained for each soft sensor. If the hypothetical veracity of a sensor is below a predetermined threshold τ, the reading of the sensor is excluded from the data fusion process.

We have run a simple experiment and simulation by recording the sound level in a room through the sound sensor module of a Google Nexus 9 tablet for five minutes, where the average sound level at the end of the fifth minute was used as the dedicated sensing value. To assess the performance of non-dedicated sensing in a similar setting, we have simulated an identical scenario where N sensors were distributed in the same terrain with Gaussian distribution, and additive white Gaussian noise was introduced to the reading of each non-dedicated sensor. We set the similarity threshold (Δ) at 0.65, whereas the predetermined threshold to exclude the soft sensor from fusion (τ) was set at 0.8. The results appear in Fig. 2, where each bar represents the average of 100 runs. The left chart predicts sound levels reported by a hard sensor, soft sensors, and soft sensors with a recommender system, while the right chart depicts the number of participating sensors vs. the total number of contributing sensors when a recommender system was used. The maximum complexity of this system will be $O(n^2)$, where n is the number of nodes in the system. For the sake of scalability, the number of recommenders can be limited so that the system is able to effectively scale itself if the number of nodes increases.

Device Misuse in Soft Sensing

Since soft sensing relies on non-dedicated resources, security and privacy are crucial concerns from the users' perspective. Besides the traditional solutions to protect sensed data, secure access and user privacy preservation are important considerations. Continuous identification and authentication via behaviometrics is an emerging concept that promises to provide a cost-effective alternative without compromising security. Despite ensuring security through continuous recognition of behavioral patterns, non-dedicated sensing

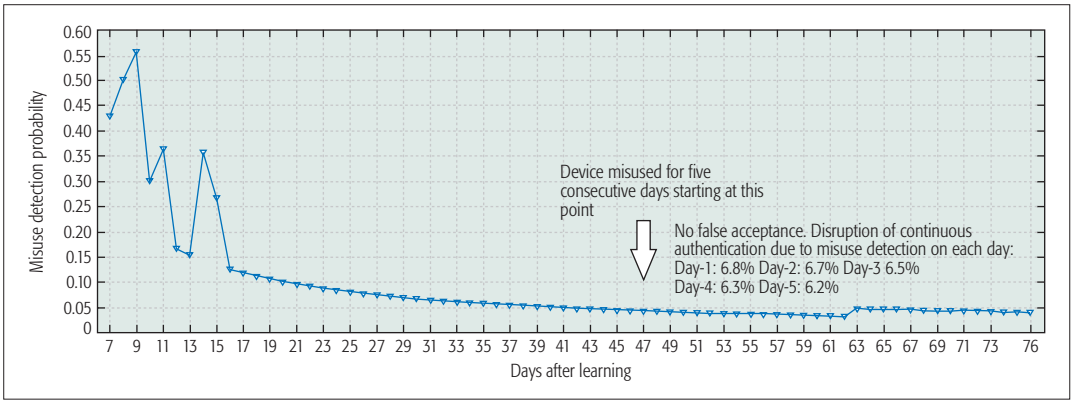

Figure 2. Continuous authentication of a representative mobile user under normal conditions and after misuse has been introduced for five consecutive days. Authenticity is ensured by a 90 percent true rejection rate when misuse is present in social sensing settings.

Using multiple layers in deep learning algorithms makes the application more computationally intense; while this does not present a computational challenge for cloud platforms, it can become infeasible for sensing nodes with limited computational and energy resources.

through built-in sensors reveals users' behavioral patterns, raising further privacy concerns.

A grand challenge in soft sensing is the misuse of devices, which can be prevented by applying machine intelligence techniques. In [5], the authors adopt the concept of an Internet of Biometric Things to reduce false acceptance to below the typical 10^{-4} to 10^{-6} range that most biometric systems claim; they utilize the rich contextual information and biometric identification methods built into most of today's mobile devices to provide authentication via knowledge-based spatio-temporal abstraction.

Figure 3 illustrates a case study for a user selected from the same dataset in [5] who provided context data (the user's social activity on various mobile applications on a smartphone) for two months. After monitoring the user data, a classifier was trained in an unsupervised manner via the DBSCAN algorithm, identifying the user with high accuracy. When artificially induced anomalous behavior patterns are injected into a sensor, the detection of a spoofed identity becomes possible by incorporating machine intelligence into contextual pattern recognition.

Soft Sensing Data in High Volume: Machine Intelligence in Smart City Sensing

Although not an exhaustive listing, Table 1 provides an overview of how machine learning algorithms can be used in smart city applications. This section elaborates on some of these algorithms.

Machine Intelligence to Aid Data Preprocessing

As a result of the increase in the number of connected devices with various built-in sensors, the amount of global data traffic will soon lead to a "data tsunami," creating the possibility of a networking bottleneck between sensors and the cloud. As discussed in [7], the sensing network can process data rather than strictly acquiring it; this creates a decentralized distributed machine intelligence system that benefits from the same accuracy as cloud-based algorithms and drastically reduces network traffic. The ever-increasing processing power in mobile and wearable devices provides the capability to apply machine intelligence techniques locally rather than in the cloud. Their proposed technique is based on *GreedyTL*, a hypothesis transfer learning (HTL) algorithm; focusing on binary classification, it allows the application of already learned models (source models) to smaller datasets (target model) in order to improve the accuracy of model extraction, assuming the datasets are independent. Their hybrid approach produced results as accurate as traditional centralized cloud processing with a decrease up to 77 percent in network overhead.

A principal difficulty in designing a classification system is extracting and defining the proper set of features; deep learning techniques are preferred due to their inherent ability to not rely on feature extraction. However, using multiple layers in deep learning algorithms makes the application more computationally intense; while this does not present a computational challenge for cloud platforms, it can become infeasible for sensing nodes with limited computational and energy resources. For these reasons, deep learning cannot extract all of the features needed for an application. As a remedy, deep learning is complemented by predefined, or *shallow*, features in a hybrid approach to boost accuracy and reduce computational intensity in [8]. Deep and shallow features are trained in a unified machine intelligence network, in which spectral representations of the data are processed using 1D convolutional kernel (similar to convolutional neural networks) with stochastic gradient descent (SGD) used to minimize the loss function. Their experimental results show that in many databases the accuracy of the algorithm is higher than deep-learning-only and shallow-feature-only approaches; classification times on Nexus 5, Galaxy S5, and Intel Edison devices were 53.8, 125.2, and 198.8 ms, respectively, proving the real-time feasibility of this approach.

Machine intelligence algorithms are also well-suited for mobile big data (MBD) as they can extract features from the vast quantity of unlabeled data gathered by mobile devices, and tend to provide highly accurate results. The biggest challenge facing machine learning in MBD is slow processing time due to the necessity for a large amount of computational power, the inherent volatility of MBD, and multidimensionality. To overcome this hurdle, an Apache Spark-based deep learning framework for MBD analytics is proposed in [10]. The decision making process is

The Street Bump application developed by the City of Boston identifies streets in need of repair by using the GPS and accelerometer sensors in citizens' mobile devices to detect bumps while driving, essentially creating an infrastructure-free soft sensing network.

Algorithm	Application	Description
Support Vector Machine	Smart transportation	Support vector machines can be used to classify data based on their position relative to a determined hyperplane, such as in the Smart Bump application which uses SVM to identify streets in need of repair [2].
Support Vector Regression	Smart environment	Support vector regression is generally used for geometrical interpretations of kernels in a feature space in the absence of local minima. HazeEst uses SVR to estimate air pollution in a given area [6].
Hypothesis Transfer Learning	Smart sensors	Hypothesis transfer learning is used in situations where direct access to sensed data is not available. A hypothesis is deduced from the data, such as in GreedyTL for binary classifcation [7].
Neural Network	Smart sensors	CNN is used in a hybrid approach for combining deep and shallow features in machine learning [8].
Linear Classifier	Smart grid	Linear classifiers can be used to characterize data received from smart grid sensors based on a linear combination of its characteristics.
Quadratic Classifier	Smart parking	A broader implementation of linear classifier, quadratic classifiers can be used in smart parking by using a quadratic surface to assist in the separation of multiple object classes obtained from soft sensors.
Binary Classification	Smart health	Binary classification uses classification rules to make a decision whether sensed data contains some specific characteristic, such as in passive RFID tags used in medical settings [9].
Decision Tree	Smart lighting	Decision trees use a tree-like model of possible decisions and their resulting consequences to determine the best course of action, such as turning lights on or off or dimming lights during certain time frames.

Table 1. A sampling of machine intelligence algorithms and their use in smart city applications.

sped up by breaking the data into partitions contained in resilient distributed datasets (RDDs) for processing in the cloud by the Spark engine. Their deep model results found recognition errors of only 14.4 percent. Data labeling has always presented a challenge in MBD, most often requiring human intervention; they call for further research in labeling approaches, such as *paid crowd-labeling*, in which people are paid by annotating data based on quantity and accuracy, and *embedded crowd-labeling*, in which people annotate data without realizing it (e.g., through CAPTCHA). The incentive(s) offered should be proportional to the value and accuracy of the data being provided.

Machine Intelligence in Smart Transportation

The *Street Bump* application developed by the City of Boston identifies streets in need of repair by using the GPS and accelerometer sensors in citizens' mobile devices to detect bumps while driving, essentially creating an infrastructure-free soft sensing network. Machine intelligence algorithms, including support vector machines (SVM) and AdaBoost, are used to establish an anomaly detection and decision support framework to identify streets in need of repair by using acquired data (i.e., coordinates of the bump, speed of the vehicle, course, and x-, y-, and z-axis readings from the accelerometer). "Bump" events are divided into two categories:

- Actionable events, such as potholes created by accidents or weather conditions, which must be detected and transmitted for repair
- Non-actionable events, such as train tracks, drains, or manhole covers, which are predictable and do not require repair [2]

The mobile device *senses* the event, while machine intelligence *processes* the data, and, in the case of actionable events, the gathered information is transmitted to notify appropriate city services (*control* personnel), as seen in Fig. 1. Street Bump was shown to accurately detect almost 50 percent of actionable events, with a false alarm rate of only 20 percent.

Machine Intelligence in Environmental Monitoring

To combat the sparsity of dedicated environmental monitoring sensors, a crowdsensing solution is proposed in [6]. The nodes, which measure levels of various atmospheric gases, use Bluetooth 4.0 (with a range of 250 ft) to transmit their data to volunteers' mobile devices, which act as relays between the sensors and the cloud. Their system, *HazeEst*, combines existing historical air quality data gathered from fixed sensors with crowdsensed node data to form a learning model used to estimate air pollution over a given area. Data was collected with their network and multiple regression models were used, but the best results were obtained from support vector regression (SVR), decision tree regression (DTR), and random forest regression (RFR). Results obtained using SVR had the lowest estimation mean absolute error (MAE) of 1.95 and the lowest root mean square error (RMSE) of 3.17, enabling HazeEst to clearly identify areas of higher pollution.

Soft Sensing Data in High Velocity: Data Analytics in Smart Cities

Once the data acquired from smart cities' sensing networks has been processed and stored, the application of data analytics algorithms can

reveal invaluable insights into many aspects of smart city applications. The value chain of "big data" is formed by the generation, acquisition, storage, and analysis of data gathered from smart city applications where data acquisition is broken down into data collection, data processing, and data control (Fig. 1). Data collection technologies include log files, sensing, and acquiring network data (e.g., libpcap-based packet capture technology, zero-copy packet capture technology, and mobile equipment). Data transmission can be achieved by inter- and intra-data-center network transmissions, whereas data preprocessing consists of integration, cleaning, and redundancy stages. Big data storage uses advanced technologies such as key-value databases, column-oriented databases, and document databases.

In several smart city applications historical data and currently acquired data need to be decoupled, as the dataset is dynamic and exposes uncertainties. In Fig. 4, we present a generic framework for IoT-driven analytics solutions for smart cities. As seen in the figure, the analytics engine basically runs parallel classifiers on a MapReduce system. This minimalist illustration of an IoT-analytics framework for smart cities can be applied to all analytics-backed smart city applications, such as those visited in this article. As such, we present this generic model for any smart city application.

Generalized Data Analytics

Some traditional data analysis methods are cluster analysis, factor analysis, correlation analysis, regression analysis, A/B testing, statistical analysis, and data mining algorithms. With the advent of the big data phenomenon, techniques such as Bloom filter, hashing, Triel (i.e., trie tree), and parallel computing are being utilized. The Bloom filter introduces the benefit of high space efficiency and query speed. Hashing can read, write, and query rapidly, but defining the proper hash function is difficult. The index method is effective in reducing disk read/write costs, and in improving insertion, deletion, modification, and query performance, but these benefits come at the expense of the additional storage space necessary for the index files. Triel uses common character string prefixes to greatly reduce the need for comparison of character strings. Parallel computing partitions a task into several independent subtasks, which can then be performed simultaneously by multiple processing devices. Message Passing Interface (MPI), MapReduce, and Dryad are some traditional parallel computing models.

A cloud-based analytics-as-a-service solution is presented in [11], which emphasizes both the design and portability of tools and methodologies. The system architecture can be divided into three layers:

- The *data acquisition, analysis, and filtering layer*, responsible for distributed and heterogeneous repositories where the data are retrieved, analyzed, and filtered
- The *resource data mapping and linking layer*, which establishes mappings among the resources where links are connected to make data semantically relevant
- The *interactive explorer layer*, which provides a scalable browsing platform to distributed datasets

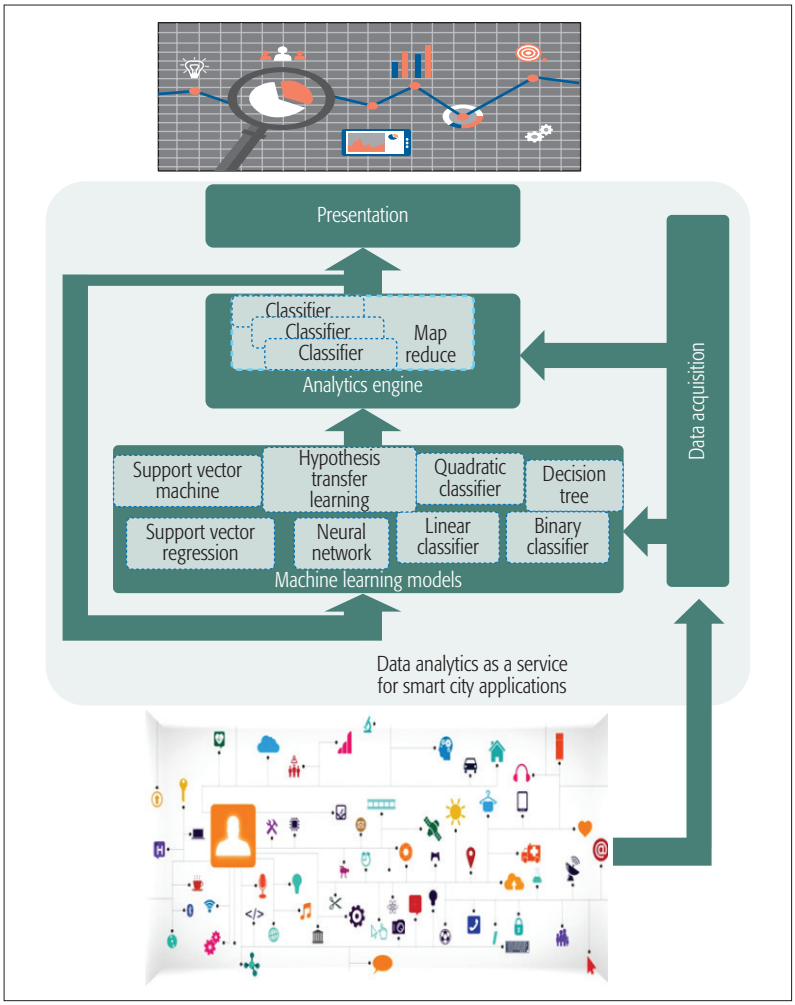

Figure 4. Building blocks of an IoT-data analytics framework for smart cities: the *data acquisition layer* acquires data through IoT devices; the *machine learning layer* builds a training model based on current and historical data; the *analytics engine layer* uses current data as well as the training model that is built offline by a machine learning algorithm; the *presentation layer* is responsible for visualization of the analytics-backed data.

Users can fix queries after obtaining results, establishing querying as an interactive process. Existing big data management tools include OpenStack and Apache Cassandra, while RapidMiner is a popular analysis tool.

Smart Grid Analytics

The processing of big data collected from sensors is crucial in the smart city application of smart grid, which can gain insights into system behavior and automatize control. A large amount of data must be collected, processed, and correlated with consumers' historical behavioral profiles. Data mining plays a crucial role in grid stability detection, required to monitor anomalous situations. Predictive models are also needed to forecast future power demand and supply in order to take proper action. In [12], the authors build a cloud-based software platform for smart grid sensor data analytics.

The platform supports *dynamic demand response* (D^2R) optimization on data from sensors and dynamic data sources, *secure repository* for easily sharing data, *scalable machine learning models* for demand prediction, and a *web portal*

The numerous smart city applications are not static; in order to properly function as a system, smart city applications need to be integrated and complementary. New data acquisition techniques are needed to ensure high-quality data, thereby increasing the veracity of soft sensor data. Valuation of data and proportional incentivization also need to be investigated.

and *mobile application* for visualization of analytics outputs. Current grids only have static demand response (DR) strategies (e.g., time-of-use price), while a smart grid offers instantaneous communication capability between customers and utilities and autonomous control at buildings, enabling D^2R to approach real-time detection, response, and notification. *Demand forecasting* and *curtailment strategy selection* are two key successful D^2R operations. The forecasting models are trained with historical energy usage patterns via regression tree machine learning, and auto-regressive integrated moving average (ARIMA) time-series are used to offer accurate predictions for D^2R. Data-driven models are beneficial as they do not require extensive technical knowledge of the system. Feature combinations can also be used to determine the most influential energy demand factors.

Smart Parking and Environment Analytics

The authors in [13] introduce a combined system for smart city development and urban planning by using big data analytics via a four-tier system: the *bottom tier* is responsible for data generation and collection; the *intermediate tier-I* is responsible for communication between sensors, devices, and the Internet backbone; the *intermediate tier-II* is responsible for data management and processing, using a Hadoop framework; and the *top tier* is responsible for the application of data analysis. It is implemented by using Hadoop with Spark, voltDB, Storm, or S4 for real-time data processing. Historical datasets are analyzed by Hadoop with MapReduce programming. The system analyzes different types of data and presents an analysis of vehicular traffic, parking lots, smart home water usage, flood patterns, and pollution. Vehicular traffic analysis can predict the travel time between two points and provide alternative routes to the destination, as well as information about real-time traffic. Parking lot analysis can identify places with available and less congested parking.

Water usage analysis can help design water usage systems for a house and control systems allowing authorities to control water resources; similar management systems can be used in electricity and gas usage. Flood analysis can predict the predefined thresholds of rain, providing advanced warning in the likelihood of a flood situation. Daily pollution analysis can be used to advise smart city residents of the intensity of pollution and if any activities need to be restricted due to pollution. Longer periods of time analysis can also be useful in urban planning and traffic management.

Smart Transportation Analytics

Smart transportation is an inseparable component of smart cities. The uncertainty in the traffic context is a result of the nonlinear interactions between vehicles, drivers, and other mobile users; the training of a machine-learning-based predictor therefore becomes rather complicated. Furthermore, due to the overfitting problem in machine intelligence and the large number of classes, offline training needs to be coupled with a real-time prediction method. A promising solution to cope with this challenge is introduced in [3]. An online parallel kNN optimization classifies traffic flow based on correlations between the current flow, while the offline distributed training module (running on a MapReduce framework) provides inputs to the online parallel kNN module based on historical data. The data used in flow prediction consists of the location, speed, acceleration, and trajectories of vehicles.

Smart Healthcare Analytics

The creation of vast amounts of healthcare data results in many challenges, including the large scale and rapid generation of data, various types of data structures, and deep value of data. The authors in [14] present a cyber-physical system for healthcare applications and services, *Health-CPS*, which uses a data collection layer with a unified standard, a data management layer for distributed storage and computing, and a data-oriented application service layer. The data collection layer gathers data from researchers, medical billing, and clinical events, and can also include physiological and emotional contributions. The data management layer consists of a *distributed file storage* (DFS) module, which includes data description, data entity, and security tag, and a *distributed parallel computing* (DPC) module, which processes data from DFS, enabling offline computation of massive unstructured data.

Open Issues and Challenges

While smart cities have already evolved into practical and productive implementation, there are still many areas where more research is necessary. It should be noted that the numerous smart city applications are not static; in order to properly function as a system, smart city applications need to be integrated and complementary. New data acquisition techniques are needed to ensure high-quality data, thereby increasing the veracity of soft sensor data. Valuation of data and proportional incentivization (to convince users to offer their non-dedicated resources for use) also need to be investigated.

Storage Challenges

In an IoT-data analytics architecture, the data's source and timestamp can be identified through spatio-temporal queries, since the IoT data is not spatially or temporally static. As it is imperative to keep track of sensed big data for future use, addressing storage challenges along with scalable data retrieval appears to be an important direction to ensure constant mobility of data. The overhead on IoT nodes is introduced by computing and communication, which results in increased energy consumption. Traditional methods such as duty-cycling or employing physical models/verifications on sensed data can be coupled with optimization models; sensed data transmissions toward the back-end can be handled by local bulking of multiple data samples in the same packet. Coordination of IoT nodes that are co-located can also avoid irrelevant or redundant readings.

Data Structuring/Labeling Challenges

Quantification of the quality of data is still an open issue in IoT-driven big data analytics. The profiling of sensed data in time, space, and other domains can assist in efficient computation of the quality of sensed data. Indeed, the trade-off between big

Smart city application	Machine intelligence	Data analytics
Smart rransportation	Support vector machines can be used with IoT sensors to optimize and control many aspects of transportation, such as traffic lights, streetlights, and warning/information signs. Roads in need of repair can be identified, such as with the City of Boston's Street Bump application [2].	Location, speed, accelerometer, and trajectories can be used for online flow prediction. The data is decomposed into historical and current data streams. The latter undergoes a distributed MapReduce framework for training whereas the latter undergoes an online parallel kNN classifier [3].
Smart environment	Support vector regression can be applied to established crowd-sensing networks to monitor air quality, atmospheric greenhouse gas levels, and major pollution sources as in the HazeEst application [6].	Multi-tier analytics framework is used. Hadoop with Spark, voltDB, Storm or S4 is used for real time data processing. Hadoop with MapReduce programming is used for historical dataset analysis [13].
Smart health	Binary classification can be used with vital sign statistics, medication information, and preventative care through observation to streamline medical processes, such as passive RFID tagging of medical devices and personnel seen in [9].	Distributed parallel computing is used for integrating diverse data via machine intelligence and data mining algorithms [14].
Smart parking	Quadratic classifiers are useful in the identification of available parking spaces in smart cities and can reduce the amount of time spent driving in search of parking, thereby reducing emissions and saving fuel and time.	The analytics system defined for smart environment is also proposed for [13].
Smart lighting	Decision trees can be used to activate lights and control power levels. Motion, ambient light, and human presence detection can automate the activation of lighting systems.	Support regression vectors are used to train the system according to daylight level and occupancy states, and this relational model is used to estimate energy consumption of the system before making an upgrade in the lighting system [15]. The proposed analytics-backed framework can be used to estimate energy consumption in smart homes and buildings where sensors are deployed to acquire relevant data.
Smart grid	Linear classifiers can assist in the extrapolation of customer demand from sensed grid data to predict and control the supply of power. Discontinuities in the grid can be detected, power can be rerouted to bypass problem areas, and optimization used to prevent blackout situations.	Regression tree-based machine learning is applied to historical data for training. Auto-regressive integrated moving average (ARIMA) time-series are used for accurate online predictions [12].
Smart utilities	Binary classification assists with real-time meter readings, and statistics can be used to optimize service and delivery of utilities. Malfunctions and areas in need of repair can be detected and forwarded to appropriate repair personnel.	Regression tree-based machine learning as well as Hadoop with MapReduce can be used for training historical data. ARIMA time series can be used with support of Hadoop [12, 13].

Table 2. Comparison of machine intelligence, data analytics, and real-time algorithms in smart city applications in terms of performance indicators. While machine intelligence is traditionally considered to be part of the data analytics process, this table is used to illustrate how efficient machine intelligence techniques can be applied in real-time decision making, and how data analytics can then be used for deeper, long-term analysis.

data storage and data quality quantification introduces an additional challenge. SensorWeb and IrisNet are some known solutions.

Volume, variety, and velocity are the most critical aspects of big data in smart cities, due to high connectivity and D2D communication. Analytics software to work on long-term and real-time data is needed. Software architectures such as Lambda are available; however, real-time systems call for optimization software. Moreover, due to unstructured and untagged data, by 2020 more than half of the cyber world will be hosting non-useful information. Hence, data acquisition and proper tagging and structuring can improve IoT data quality.

Privacy Challenges

The security of sensed big data, as well as device-level privacy in a smart city IoT architecture, is a grand challenge. "Cloudification" of storage, processing, and networking will encompass 40 percent of the cyber world, which will undoubtedly lead to the *security as a service* concept in cloud analytics. Last but not least, continuous sensing and reporting by IoT sensors need to be processed in order to provide output. Thus, scalable and analytics-backed visualization methodologies for long-term data are also necessary to prevent data overloads for IoT big data systems.

Summary and Concluding Remarks

This article provides a discussion of the use of machine intelligence and data analytics algorithms on data acquired from the sensing networks integral to smart city applications. The emergence of crowdsensing in smart cities is revolutionizing the way data is obtained, but serves to increase the already massive volume of data. The processing bottleneck caused by massive quantities of acquired data can be broken with the application of various machine intelligence algorithms. Once the data is processed and stored, valuable statistical correlations and predictions can be extracted through data analytics. The use of machine intelligence and data analytics in various smart city applications are highlighted in Table 2, showing the short-term real-time processing benefits of machine intelligence and the long-term benefits of data analytics. Finally, open issues and challenges facing machine intelligence and data analytics for software-based sensing in smart city applications are discussed.

The emergence of crowd-sensing in smart cities is revolutionizing the way data is obtained, but serves to increase the already massive volume of data. The processing bottleneck caused by massive quantities of acquired data can be broken with the application of various machine intelligence algorithms.

Acknowledgment

This work was supported in part by U.S. National Science Foundation grants CNS-1239423 and CNS-1464273 and Natural Sciences and Engineering Research Council of Canada (NSERC) grant RGPIN/2017-04032.

References

[1] A. Al-Fuqaha *et al.*, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 4, 4th qtr. 2015, pp. 2347–76.

[2] T. S. Brisimi *et al.*, "Sensing and Classifying Roadway Obstacles in Smart Cities: The Street Bump System," *IEEE Access*, vol. 4, 2016, pp. 1301–12.

[3] D. Xia *et al.*, "A MapReduce-Based Nearest Neighbor Approach for Big-Data-Driven Traffic Flow Prediction," *IEEE Access*, vol. 4, 2016, pp. 2920–34.

[4] M. Pouryazdan *et al.*, "Quantifying User Reputation Scores, Data Trustworthiness, and User Incentives in Mobile Crowd-Sensing," *IEEE Access*, 2017.

[5] F. Anjomshoa *et al.*, "Detection of Spoofed Identities on Smartphones via Sociability Metrics," *IEEE ICC*, May 2017.

[6] K. Hu *et al.*, "Hazeest: Machine Learning Based Metropolitan Air Pollution Estimation from Fixed and Mobile Sensors," *IEEE Sensors J.*, vol. 17, no. 11, 2017, pp. 3517–25.

[7] L. Valerio, A. Passarella, and M. Conti, "Hypothesis Transfer Learning for Efficient Data Computing in Smart Cities Environments," *Proc. 2016 IEEE Int'l. Conf. Smart Computing*, May 2016, pp. 1–8.

[8] D. Rav *et al.*, "A Deep Learning Approach to On-Node Sensor Data Analytics for Mobile or Wearable Devices," *IEEE J. Biomedical Health Informatics*, vol. 21, no. 1, Jan 2017, pp. 56–64.

[9] S. Parlak *et al.*, "Passive RFID for Object and Use Detection during Trauma Resuscitation," *IEEE Trans. Mobile Computing*, vol. 15, no. 4, Apr. 2016, pp. 924–37.

[10] M. A. Alsheikh *et al.*, "Mobile Big Data Analytics Using Deep Learning and Apache Spark," *IEEE Network*, vol. 30, no. 3, May/June 2016, pp. 22–29.

[11] Z. Khan, A. Anjum, and S. L. Kiani, "Cloud Based Big Data Analytics for Smart Future Cities," *Proc. 2013 IEEE/ACM 6th Int'l. Conf. Utility Cloud Computing*, 2013, pp. 381–86.

[12] Y. Simmhan *et al.*, "Cloud-Based Software Platform for Big Data Analytics in Smart Grids," *Computing in Science & Engineering*, vol. 15, no. 4, 2013, pp. 38–47.

[13] M. M. Rathore *et al.*, "Urban Planning and Building Smart Cities Based on the Internet of Things Using Big Data Analytics," *Computer Networks*, vol. 101, 2016, pp. 63–80.

[14] Y. Zhang *et al.*, "Health-CPS: Healthcare Cyber-Physical System Assisted by Cloud and Big Data," I*EEE Systems J.*, 2015.

[15] D. Caicedo and A. Pandharipande, "Sensor Data-Driven Lighting Energy Performance Prediction," *IEEE Sensors J.*, vol. 16, no. 16, Aug. 2016, pp. 6397–6405.

Biographies

Hadi Habibzadeh [S'17] received his B.S. in computer engineering from Isfahan University of Technology, Iran, in 2015 and his M.S. degree in technical entrepreneurship and management from the University of Rochester, New York, in 2016. He is currently pursing a Ph.D. degree in the Electrical and Computer Engineering Department of the University at Albany, State University of New York (SUNY Albany) under the supervision of Dr. Tolga Soyata. His current research interests include cyber physical systems and embedded systems with applications in the Internet of Things and smart cities.

Zhou Qin [S'17] received his B.S. in EIE from Harbin Institute of Technology, P.R. China, in 2014. He is currently pursuing his Ph.D degree in the Department of Computer Science of Rutgers University. During his Ph.D. studies, he also pursued his M.S. from the College of Electronic Science and Engineering of the National University of Defense Technology, P.R. China. His current research interests include global navigation satellite systems, integrated navigation systems, and their application in multisensory navigation.

Andrew Boggio-Dandry [S'17] is currently pursuing his B.S. degree in computer engineering in the College of Engineering and Applied Sciences, SUNY Albany.

Tolga Soyata [M'08, SM'16] received his B.S. degree in electrical and communications engineering from Istanbul Technical University in 1988, his M.S. degree in electrical and computer engineering from Johns Hopkins University in 1992, and his Ph.D. in electrical and computer engineering from the University of Rochester in 2000. He joined the University of Rochester ECE Department in 2008. He was an assistant professor — research at University of Rochester ECE when he left to join SUNY Albany, Department of ECE as an associate professor in 2016. His research interests include cyber physical systems, digital health, and GPU-based high-performance computing. He is a Senior Member of ACM.

Burak Kantarci [S'05, M'09, SM'12] is an assistant professor with the School of Electrical Engineering and Computer Science at the University of Ottawa. From 2014 to 2016, he was an assistant professor at the ECE Department at Clarkson University, where he currently holds a courtesy appointment. He received his M.Sc. and Ph.D. degrees in computer engineering from Istanbul Technical University in 2005 and 2009, respectively. During his Ph.D. study, he studied as a visiting scholar with the University of Ottawa, where he completed the major content of his thesis. He is an Editor of *IEEE Communications Surveys & Tutorials* and *IEEE Access*.

Hussein Mouftah [S'74, M'76, SM'80, F'90 LF'12] has been with the School of Electrical Engineering and Computer Science, University of Ottawa since 2002 as a Senior Canada Research Chair and Distinguished University Professor. He was with the Electrical and Computer Engineering Department of Queen's University from 1979 to 2002. He has three years of industrial experience, mainly at BNR of Ottawa (Nortel Networks) from 1977 to 1979. He is the author or coauthor of eight books, 63 book chapters, more than 1300 technical papers, and 12 patents in this area. He is a Fellow of the Canadian Academy of Engineering, the Engineering Institute of Canada, and the Royal Society of Canada RSC: The Academy of Science.