Ma *et al. J Big Data* (2020) 7:103
https://doi.org/10.1186/s40537-020-00378-7

Journal of Big Data

METHODOLOGY Open Access

Big data actionable intelligence architecture

Tian J. Ma[1,2*], Rudy J. Garcia[1,2], Forest Danford[1,2], Laura Patrizi[1,2], Jennifer Galasso[1,2] and Jason Loyd[1,2]

*Correspondence:
tma@sandia.gov
[1] Sandia National Laboratories, Albuquerque, NM 87185, USA
Full list of author information is available at the end of the article

Abstract

The amount of data produced by sensors, social and digital media, and Internet of Things (IoTs) are rapidly increasing each day. Decision makers often need to sift through a sea of Big Data to utilize information from a variety of sources in order to determine a course of action. This can be a very difficult and time-consuming task. For each data source encountered, the information can be redundant, conflicting, and/or incomplete. For near-real-time application, there is insufficient time for a human to interpret all the information from different sources. In this project, we have developed a near-real-time, data-agnostic, software architecture that is capable of using several disparate sources to autonomously generate Actionable Intelligence with a human in the loop. We demonstrated our solution through a traffic prediction exemplar problem.

Keyword: Big data, Actionable intelligence, Data architecture, Data fusion, Traffic prediction

Introduction

The amount of data produced by sensors, Internet of Things (IoTs), social and digital media, are rapidly increasing each day [1]. The International Data Corporation expects that there will be 175 zettabytes of data worldwide by 2025 [2]. There is significantly more information as compared to the number of people analyzing it. This becomes a potential problem, where lots of data could get overlooked. Data storage, retrieval, and maintenance can become extremely costly due to the explosion of data. At some point, it might not be financially feasible to store all the data that is received. Hence, if data is not analyzed as it is received, the information collected could be lost forever. Decision support in a dynamic real-time environment using large volumes of structured, unstructured, and semi-structured data can be a research challenge [1]. Many Big Data analytic techniques such as regression analysis [3] and machine learnings [4] have been available for many years. However, data mining and data analytics [5] are post-event processes [6, 7, 8], which are inadequate to support real-time decision making. Actionable intelligence is the next level of data analysis where data are analyzed in near-real-time to create insights that support decision making [1]. In this paper, we will discuss a Big Data Actionable Intelligence (BDAI) framework that can quickly turn real-time streaming data from a variety of sources into actionable insights. Our framework architecture has demonstrated the ability to integrate disparate data sources from a variety of interfaces in near-real-time. Our platform addresses the National Spatial Data Infrastructure

Executive Order 12,906 concepts by providing "the technology, policies, standards, and human resources necessary to acquire, process, store, distribute, and improve utilization of geospatial data." [9]. This paper is organized as follow. "Exemplar problem" section provides a discussion on the data sources and exemplar we used to demonstrate our architecture. "Related works" section goes over any related work in current open literature. "Methods" section discusses our approach to the BDAI problem. "Results and discussion" section provides a discussion of the results in our project. "Conclusion" section goes over the conclusion of our research.

Exemplar problem

Exemplar description

To demonstrate our capability of transforming Big Geospatial Data to Actionable Intelligence in near-real-time, we focused on an exemplar problem of generating Actionable Intelligence in regard to the traffic congestion in the city of Chicago. The traffic prediction problem is extremely complex, which makes it hard to accurately predict traffic condition based on off-line data (patterns, trends, road networks, etc.) or crowdsourcing applications such as Waze [10] due to the dynamic changes of real-time environment (i.e. accidents, sport events, weather changes, etc.). This exemplar highlights the importance of Actionable Intelligence. For example, first responders need to safely and expeditiously transport a victim to the hospital. Rapidly identifying the fastest route to a medical facility increases the survivability of the victim. Actionable Intelligence provides timely information such as heavy traffic, which allows the first responders to make important time saving transportation decisions.

Data sources

Table 1 provides the data sources used to test the BDAI framework. Figure 1 provides a high-level pictorial illustration of each data types. The data sources were extremely diverse, in terms of data types and data frequency. Most of the data interfaces provided ways to geospatially constraint the results within the Chicago city limits. One of the data sources included a 3-h ground truth dash camera video experiment to validate actionable intelligence created from our framework.

System requirement

A summary of requirements and metrics that we used to evaluate our system is depicted in Table 2.

Assumption about data

We made the following general assumptions in regards to data:

1. Data can be referenced by time and geospatial extent.
2. Each data type may not follow a standardized format. Hence, architecture needs to accommodate needed flexibility to onboard new format.
3. Input data can come from variety of form (structured, semi-structured, or unstructured).
4. Data might not be immediately available for retrieval due to site restriction.

Table 1 Heterogenous data sources

Data Sources	Source Type	Frequency	Description
Twitter [11]	Live text	Live. Query every 5 min	Decahose – Geo-tagged tweets within Chicago city limits
Travel Mid-west [12]	Various	Traffic camera images every 15 min Vehicle Detection System (VDS) every 10 min Dynamic Message Sign (DMS) every 10 min Thousands of camera locations	Traffic Cameras VDS—Vehicle Speeds, Vehicle Occupancy DMS – Traffic times, Lane Closures, Accidents
City of Chicago [13]	Various	Traffic Segments every 10–15 min Traffic Region every 10–20 min Construction Moratorium—Infrequent	Traffic Segments – Vehicle Speeds, Vehicle Occupancy Traffic Region—Vehicle Speeds, Vehicle Occupancy Construction Moratorium – Road closures
GDELT [14]	Various	Every 15 min	Global Knowledge Graph – provides context and feeling between people, organizations, and locations Event Mentions, Events
MapQuest [15]	Various	Every 5 min	Reported Incidents
Digital Globe [16]	Satellite Imagery	1–3 images a day	Satellite Imagery (limited number of images)
Dash Camera	3-h Video	Field experiment	Dash Camera Video (Live Experiment and Validation)

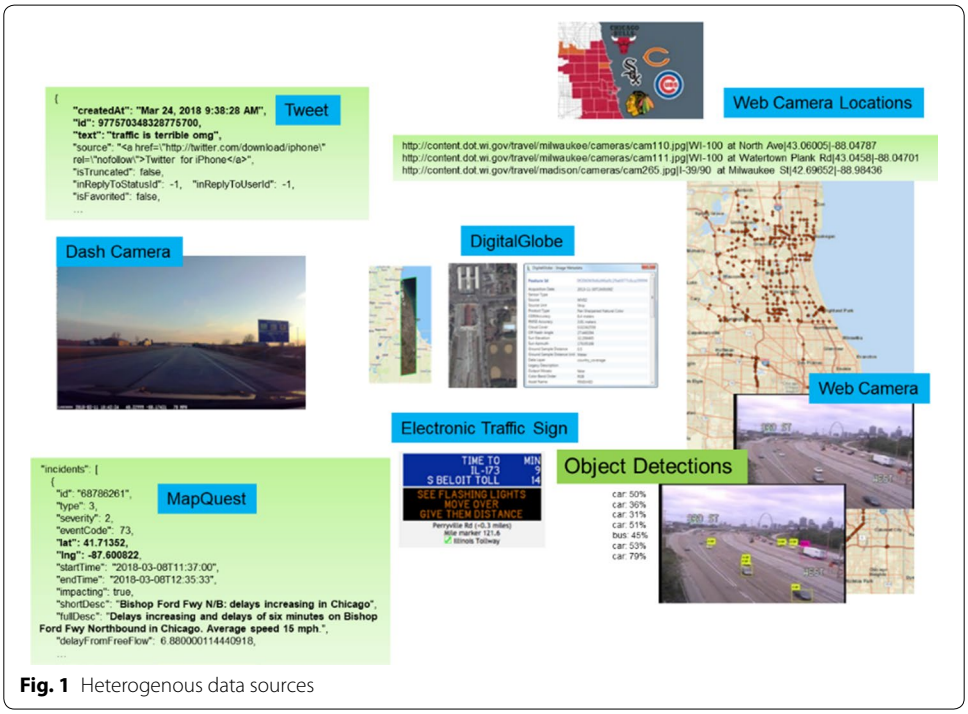


Fig. 1 Heterogenous data sources

5 Data might not always be updated on a regular interval.

Table 2 System requirement

Requirements	Descriptions	Goal	Threshold
Scalability	Number of streaming location supported	150 streaming location	100 streaming location
Data Variety	Structured, Unstructured, Semi-structured	Structured, Unstructured, Semi-structured	Structured, Semi-structured
Average Throughput Per Location	Average data transfer rate per location	1 Mbps per source Location	0.50 Mbps per source location
Average Data Latency	Time measured from data creation to the time the data has arrived and indexed into our system	Less than or equal to the polling frequency	max (polling frequency, data update frequency) + 2 min
Data Management Guarantees	Level of guarantee on which message to be processed	Fully process each message	Drop message on failure
Traffic Classification Accuracy	Traffic Classification Accuracy	95% accuracy on trained location	90% accuracy on trained location

Related works

Traffic prediction analysis is typically done in a crowd sourcing way, where location information from GPS apps are shared among users to help predict the fastest route [17]. Recently, improvement in traffic prediction accuracy using social media data has been demonstrated [18]. Despite many researches on traffic prediction [19], many existing research focuses on using few data sources for traffic prediction. Based on our research, we were not aware of any existing work utilizing a combination of data sources such as Twitter, web camera imageries, satellite imagery, dash camera video, Mapquest, and GDELT to support near-real-time traffic prediction. Our work uses seven disparate data sources as described in Table 1. Each data sources can be streamed from multiple locations. The web camera data in particular, involves the live streaming of over hundreds of camera locations around the City of Chicago. The traffic reports are received from hundreds of stations. Existing software architecture [20] typically focuses on acquisition, storage, and the retrieval of Big Data. However, our architecture focuses on Actionable Intelligence generations. Several data architecture has been proposed for network traffic monitoring applications [21-23], but our data architecture supports multiple disparate data sources. A general five-layer Big Data Processing and Analytics (BDPA) involves a collection layer, a storage layer, a processing layer, an analytic layer, and an application layer [24]. However, this architecture does not address actionable intelligence generation in their framework. In 2019, Zhu et al. states: "Currently, there are no widely accepted BDPA solution, especially a general-purpose solution fit for both traditional and internet industries [24]." Liu et al. [25] proposed a general multi-source framework [25] to map disparate data sources to a common unified data format for Big Data fusion. Their paper suggested the benefits of combining heterogenous sources to provide a better solution, but it did not provide a solution on how this framework can be integrated with Big Data streaming sources. Hence, the motivation for our work focuses on using Big Geospatial Data to answer key customer geospatial and temporal questions. Big Geospatial Data is Big Data with geospatially tagged features and error estimates. As stated by the NIST Big Data Public Working Group (NBD-PWG), "Big Data consists of extensive datasets, primarily in the characteristics of volume, variety, velocity, and/or variability—that require a scalable architecture for efficient storage, manipulation,

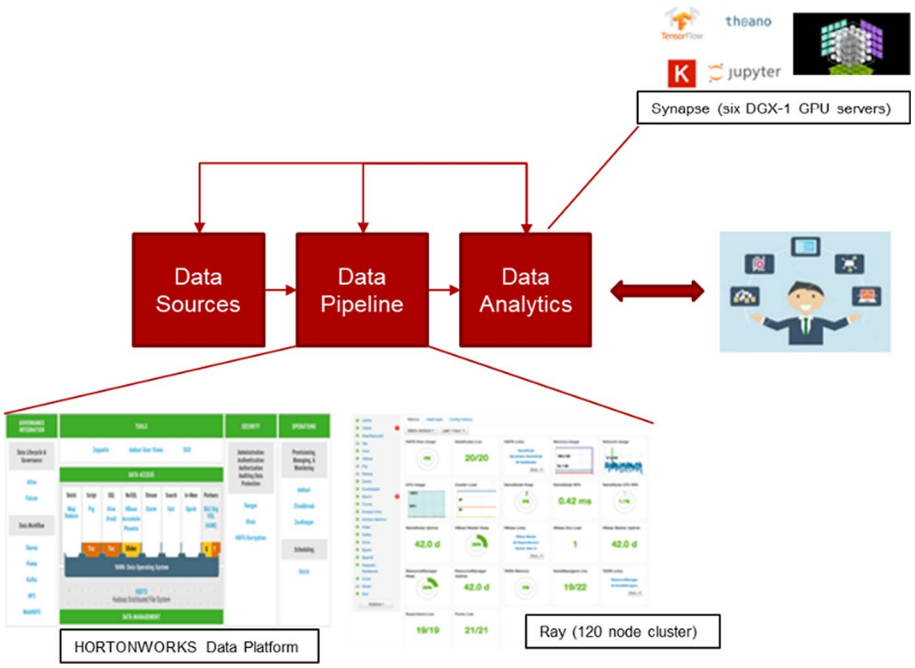


Fig. 2 BDAI SNL architecture

and analysis." [26]. While most Big Data information fusion solution focuses on social media data sources [27], our architecture accommodates a variety of geospatially tagged data sources at various velocities and veracities. Our traffic prediction exemplar allows us to test and validate key BDAI capabilities: handling heterogenous data sources, hosting data pipelines on distributed processing platforms, and running machine learning algorithms in near-real-time. The exemplar is not meant to compete with crowd sourcing GPS apps, but rather serve as a generic exemplar that can be extended to other Big Data Actionable Intelligence problems.

Methods

System setup

Our BDAI software was initially deployed to a bare metal system named "Ray". We deployed, configured, and tested the HORTONWORKS Data Platform (HDP) Apache Hadoop Distro [28] to the Ray cluster, composing of 120 computing nodes and 400 TB of Hadoop Distributed File System (HDFS) [29] storage. Since initial deployment, we have migrated our BDAI software to run on a cloud infrastructure (Azure Stack [30]). Most of our custom data processing code is implemented in Java, [31] with some processing implemented in Python [32].

BDAI architecture contributions

A high level of our BDAI architecture is depicted in Fig. 2. While a similar architecture has been proposed in open literature [20, 24], these architectures focus on acquisition, storage and retrieval of Big Data, and on the use of specific datatypes [22, 23]. The key question we want to answer in this paper is: Can we create a near-real-time data agnostic software

Fig. 3 Big data technology stack

architecture that can process many disparate sources while autonomously generate Actionable Intelligence? In order to combine and fuse disparate streaming data sources to produce actionable intelligence, we believe Big Data should be curated as it arrives to the system. Our main contributions to the Big Data Architecture field is listed as such: 1. Provide a general framework to map data from disparate data sources into a common frame of reference indexed by time and geo-spatial extent. This enables our architecture to stay data agnostic, which provides the possibility to quickly onboard new data sources that allows for agile responses to complete new and orthogonal scenarios. This method also provides the ability to ask questions generically over many disparate data sources, which minimizes the learning curve to perform meaningful fusion and analysis. 2. Provide a high-level description of our implementation in which our architecture uses a modern Big Data technology stack (depicted in Fig. 3). This software stack is natively distributed and built for high-throughput streaming that allows us to tackle problems of mission-level magnitude. 3. Demonstrate and prove that our architecture and technology stack are capable of supporting the streaming of disparate data sources to produce actionable intelligence.

BDAI Architecture–algorithm workflow

Our architecture contains four levels of processing: Data Source, Data Pipeline, Data Analytic, and Data Reporting. First, we set up a streaming interface connection for each data source. We utilized Apache Storm's topology [33] and Apache Kafka's [34] interprocess communication mechanism to implement our Data Pipelines because they are known to achieve a high level of scalability, low latency, fault-tolerant, and the data is guaranteed [35, 36]. A general workflow of our data pipeline is depicted in Fig. 4.

We created a separate processing Storm Topology [33] for each data type. Each topology follows a similar workflow of acquiring, normalizing, processing, and publishing the data (see Fig. 3). Apache Kafka is used as a central messaging broker, connecting each step of the processing. For example, when incoming data arrives, it will first be placed in Kafka, and the "Getter" will be informed to obtain the data. The "Getter" is responsible for acquiring the data from an individual data source. The "Normalizer" is responsible

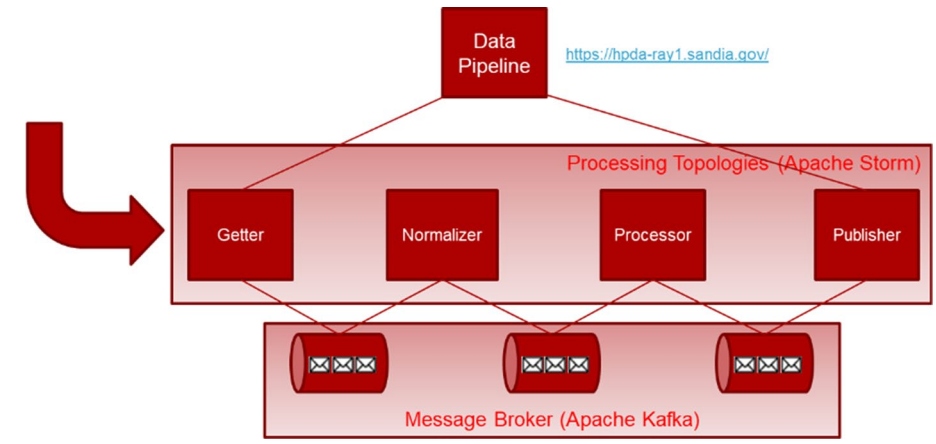


Fig. 4 Data pipeline

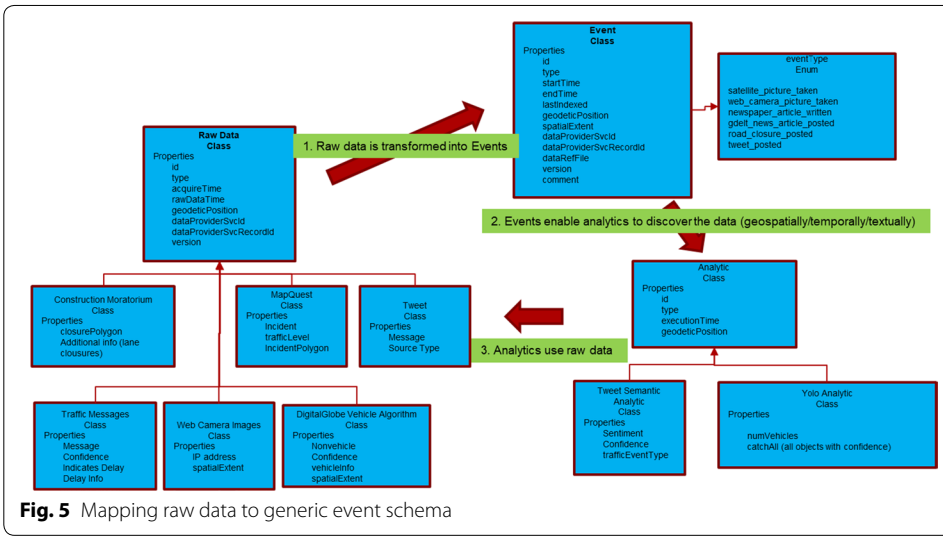


Fig. 5 Mapping raw data to generic event schema

for transforming the data by mapping out both raw data and metadata into a common event schema. A description of the event schema is depicted in Fig. 5. The ontology mapping of each individual data source into a common event description is depicted Fig. 6. The mapping of each individual data source into a common data schema is necessary to establish a common frame of reference for events that occurs at a given in time and space. This design makes searching for the events in a specific time or space to be easily accessible. All the data sources are "normalized" with the same common event schema, in which they are all "linked" by the time and its location. By tagging the data in this manner, it ensures that the data can be discoverable by geospatial analytic processing in later steps.

The "Processor" is responsible for extracting events from raw sensor data and then populating its results in the event schema. The "Publisher" is responsible for "indexing" the data to enable search and discovery at the "Data Analytic" level. Apache Solr [37],

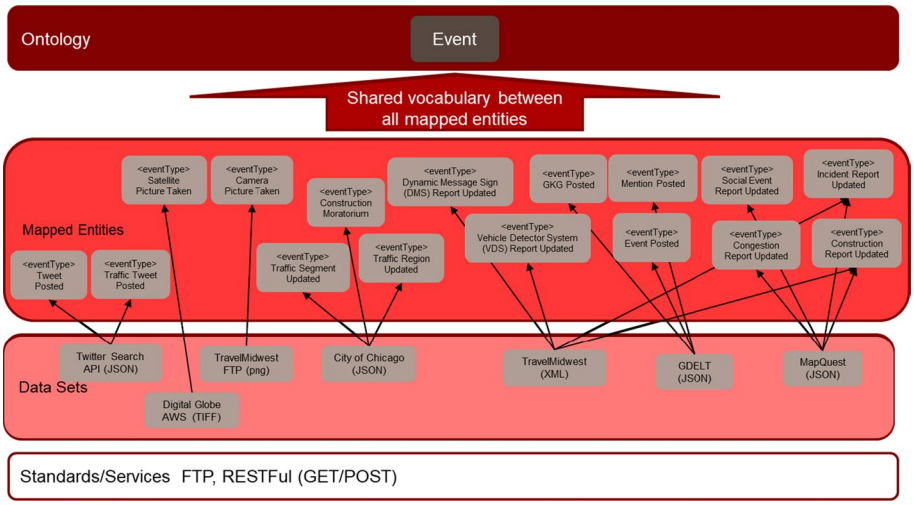


Fig. 6 General event schema

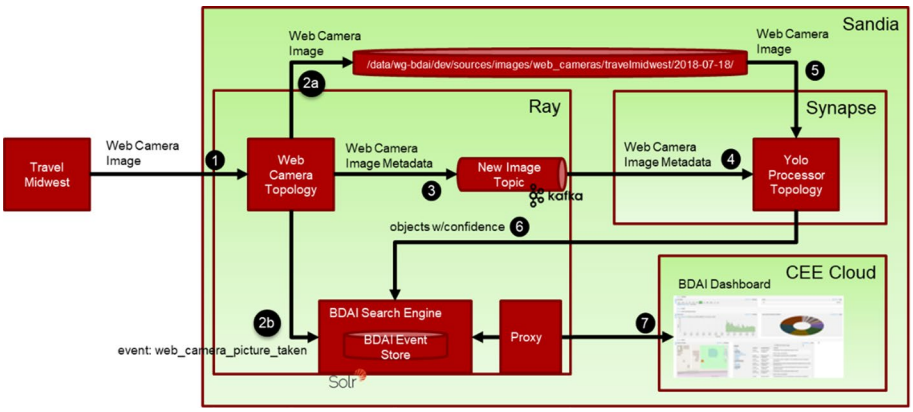


Fig. 7 Camera topology example

an enterprise search engine, is used for both indexing and querying the geospatial and temporal data.

In our design, we developed a custom topology for each data type. The custom design provides flexibility to support different data types. An illustration of a web camera topology insertion is depicted in Fig. 7. In this example, the "Processor" was built based on an object detection algorithm called You Only Look Once (YOLO) [38]. As depicted in Fig. 8, the pre-trained YOLO processor did not yield good results. Hence, we labeled and re-trained YOLO using the web camera images from Travel Mid-West. Results of the re-trained YOLO processing are also depicted in Fig. 8 as a comparison. The output of YOLO is used to determine the number of cars in each camera image. The event (i.e. number of cars at a location) generated from the YOLO topology is indexed by image time (when the image is captured) and image location (i.e. latitude and longitude of where the event occurred).

For the Tweeter Topology, we implemented a separate machine-learning "Processor" to process live tweets to generate traffic sentiment. Similarly, we indexed tweeted events by

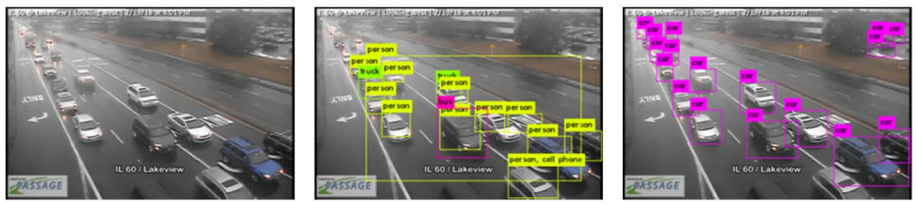


Fig. 8 YOLO results

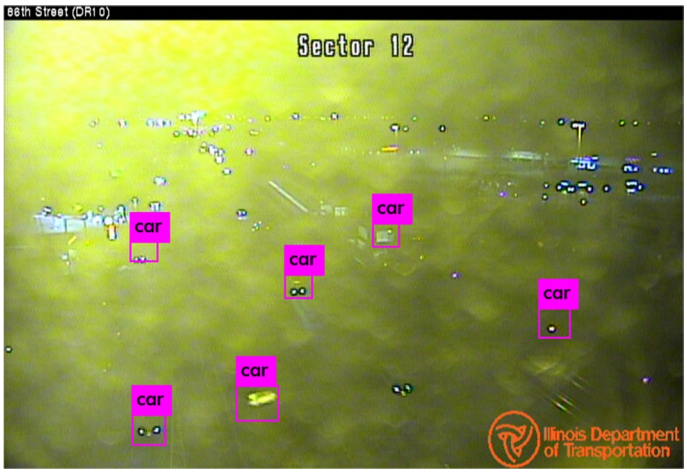


Fig. 9 Vehicle detections reported by YOLO processor between 87th St and 71st

their time and location on where/when the events were tweeted. Following a similar workflow, we created separate topologies for each of the other data types as listed in Table 1.

Muti-source data fusion

Information Fusion (IF) is a process of combining data or information to develop improved estimates or predictions of entity states [39]. Information obtained from a single source can be unreliable or insufficient to make an accurate determination. For example, in one traffic scenario on the Dan Ryan Expressway Inbound between 87th St and 71st St on March 22, 2019, our YOLO topology had reported light traffic conditions because there were very few cars detected (see Fig. 9). However, information received from our Tweet Processor indicated that the road was closed due to police activity (see Fig. 10). Since the Tweet information had already been indexed by time and location, we could easily perform a geospatial query to obtain the Tweet's Information to match the closest image time and location. Hence, the use of multiple data sources is necessary in order to improve the reliability and quality of the information provided to decision makers.

```
tweet_traffic_posted:
      [
       {
        "comment": "Closed due to police activity in #DanRyan on Dan Ryan
Inbound between 87th St and 71st St #traffic #Chicago
https://t.co/GCesNZoWap",
        "dataProviderSvcId": "TTWN Chicago",
        "loc": "DanRyanat86thSt.",
        "endTime": "2019-03-22T08:44:23Z",
        "eventType": "tweet_traffic_posted",
        "id": "11090129394266685 45",
        "spatialExtent": "POINT (-87.62453 41.73621)",
        "dataRefFile": "1109012939426668545",
        "startTime": "2019-03-22T08:44:23Z",
```


Fig. 10 Tweets reported by Tweet processor between 87t St and 71st

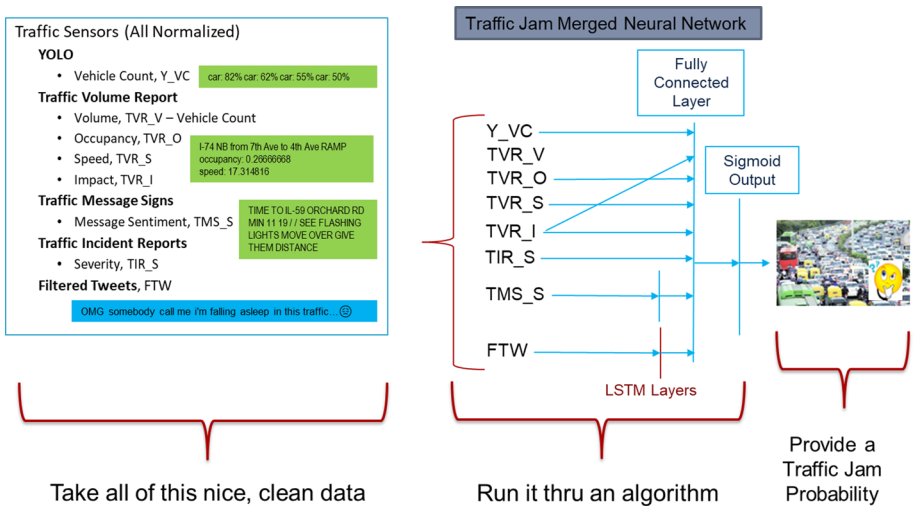


Fig. 11 BDAI merge neural network

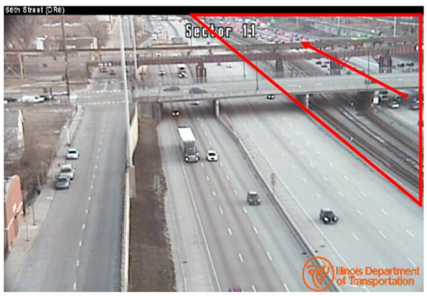
BDAI architecture analytical fusion algorithm

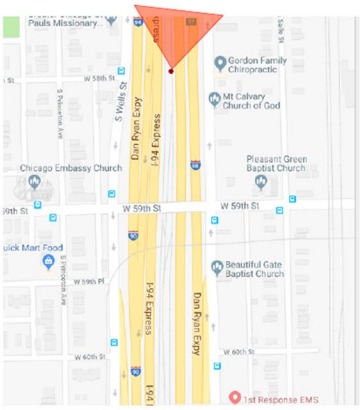
Our BDAI analytic seeks to combine event data from disparate sources to predict traffic congestion by improving the outcome beyond what could be done with a single source of information. At the data analytic level, we first query the normalized and curated data from all data sources by time and location. Then, we performed a data analytic on events occurring at similar times and locations. To demonstrate how machine-learning algorithm can be integrated into our architecture, we designed a Merged Neural Network (as depicted in Fig. 11) to perform the traffic congestion classification. The algorithm takes input from all the normalized event data (related by time and location) to produce a traffic congestion probability. The output is a real-valued number between 0 and 1, as related to the level of traffic, where 0 is negligible traffic and 1 is a severe, complete standstill traffic jam.

Results and discussion

Chicago traffic analytic–multi-source analytical fusion demonstration

A web camera image which captured the traffic condition on the Dan Ryan Expressway is depicted in Fig. 12. At the corresponding time frame, our BDAI system was






```
"id": "417dff01-921c-492b-a2b5-c2b45a4a2bfe",
"eventType": "camera_picture_taken",
"startTime": "2019-03-20T01:12:16Z",
"endTime": "2019-03-20T01:12:16Z",
"geodeticPosition": "POINT (-87.630990 41.788890)",
"spatialExtent": "POINT (-87.630990 41.788890)",
"version": "1.5",
"dataProviderSvcId": "travelmidwest",
"dataProviderSvcRecordId": "2019-03-20T01:12:16Z_DR6.jpg",
"dataRefFile": "/data/wg-bdai/dev/sources/images/web_cameras/travelmidwest/2019-03-20/2019-03-20T01:12:16Z_DR6.jpg",
"comment": "Dan Ryan at 58th St.",
"_version_": 1628485129289597000,
"lastIndexed": "2019-03-20T01:14:02.452Z"
```


Fig. 12 Camera image indicating traffic congestion on Dan Ryan


```
"eventType": "tweet_traffic_posted",
"startTime": "2019-03-20T00:40:20Z",
"geodeticPosition": "POINT (-87.63108 41.7871)",
"dataProviderSvcId": "TTWN Chicago",
```

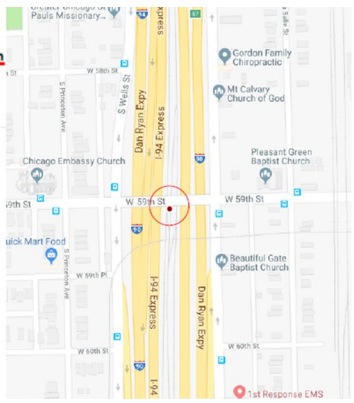
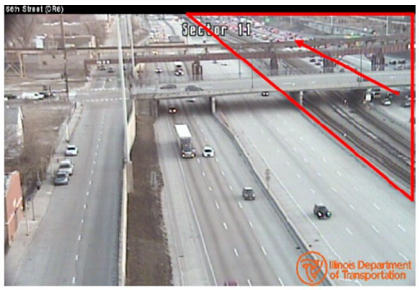



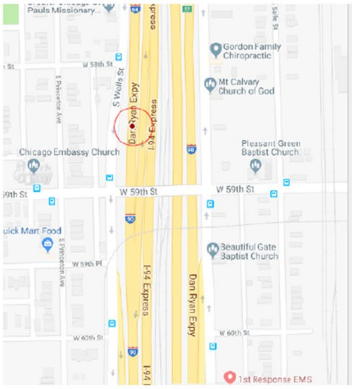
Fig. 13 Tweets indicated Dan Ryan Outbound at 59th St was closed due to an accident

able to locate a tweet from the Total Traffic Chicago data source indicating that the road was closed due to an accident in the area (see Fig. 13). At a similar time frame, the BDAI system had confirmed slow traffic through a traffic report from Mapquest (see Fig. 14). However, Mapquest had reported that the West Dan Ryan Expressway had light traffic (see Fig. 15). This information was also confirmed by the small number of cars detected (Fig. 16) by our web camera topology. Taking into account all of the sources, BDAI was able to distinguish the traffic congestion level on both sides of the West Dan Ryan Expressway.

Traffic classifier performance

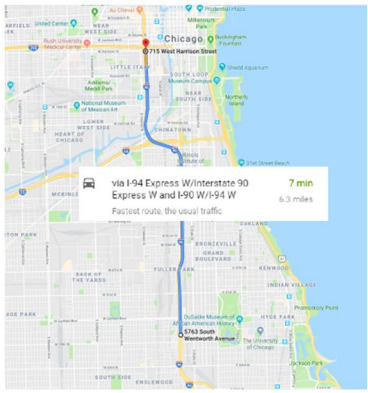
Overall, the BDAI Merge Neural Network classifier performed extremely well on intersections where the network was trained. We also tested the BDAI Merge Neural






```
"startTime": "2019-03-20T00:31:33Z",
"eventType": "vds_report_updated",
"geodeticPosition": "POINT (-87.631590 41.788020)",
"occupancy": 19.971428,
"speed": 25.032557,
"volume": 960,
"streetName": "I-90",
"crossStreet": "59th St",
"trafficDir": "EAST_BOUND"
```


Fig. 14 Congestion Report from Map Request reported slow speed on EAST BOUND



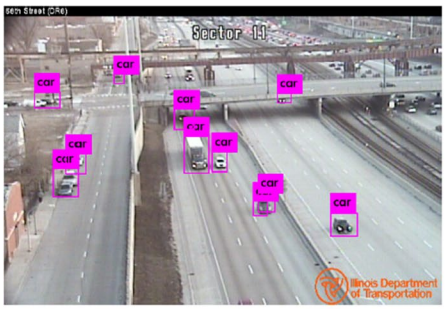

```
"startTime": "2019-03-20T00:40:11Z",
"eventType": "dms_report_updated",
"geodeticPosition": "POINT (-87.630840 41.789370)",
"occupancy": -1,
"speed": -1,
"volume": -1,
"streetName": "Dan Ryan Express Lane",
"crossStreet": "57th St",
"trafficDir": "WEST_BOUND",
"comments": "STOP THE TEXTS STOP THE WRECKS / 7 MINUTES TO JANE BYRNE INTERCHANGE"
```


Fig. 15 Congestion Report from Map Request reported light traffic

Network classifier on intersections where it was not trained. As expected, the performance was not good. A summary of the performance of our classifier is depicted in Fig. 17.

BDAI dashboard

The output of the BDAI system is visualized using a Banana Dashboard [40], as depicted in Fig. 18. The BDAI Dashboard is back ended by an Apache Solr Cluster, which contains all event data. The map in the lower left represents the event records that were ingested in one of our data pipelines. The icons are the actual geospatial locations of the events. The event metadata is the table to right of the map.




```
"analyticType": "yolo object identification",
"analyticOutput": "car: 73% car: 65% car: 65% car: 56% car: 50% car: 47% car: 44% car: 40% car: 39% car: 35% car: 25%",
"processVersion": "yolo-v2"
```


Fig. 16 Small Number of Cars is detected East Bound Traffic

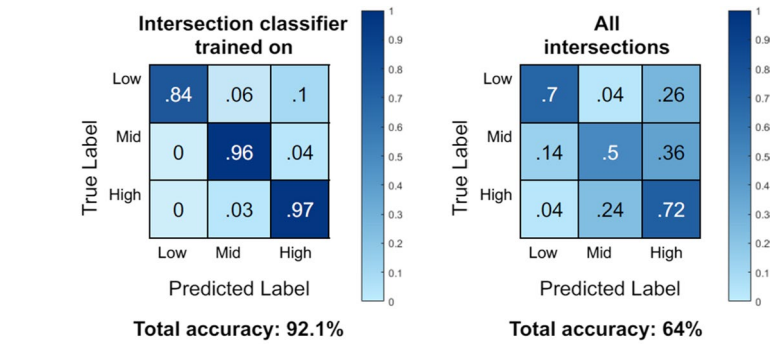


Fig. 17 Merge Neural Network Results

System performance

Our BDAI software was deployed to a bare metal system named "Ray" ("System setup" section). A summary of the system performance from "Ray" for all event types is depicted in Table 3. We are not aware of any similar systems that are published in open literature to draw a direct comparison from our effort. The missing entries in the table are due to insufficient information in that particular event type to derive statistics. Each event type can have multiple sources as there may be multiple camera locations or traffic report stations active at a given time. Our data architecture supports concurrent streaming from each data sources. Each event type restricts how frequent we can "poll" the data. Hence, "polling" is not done instantaneously when the event is available, but rather done at a fixed time interval, as permitted by the external source. This is not a limitation in our architecture, but rather a limitation set forth by an external data source. Latency in Table 3 is measured from the time an event happens, to the time that the event is curated and indexed into Solr. This does not account for any additional latency required

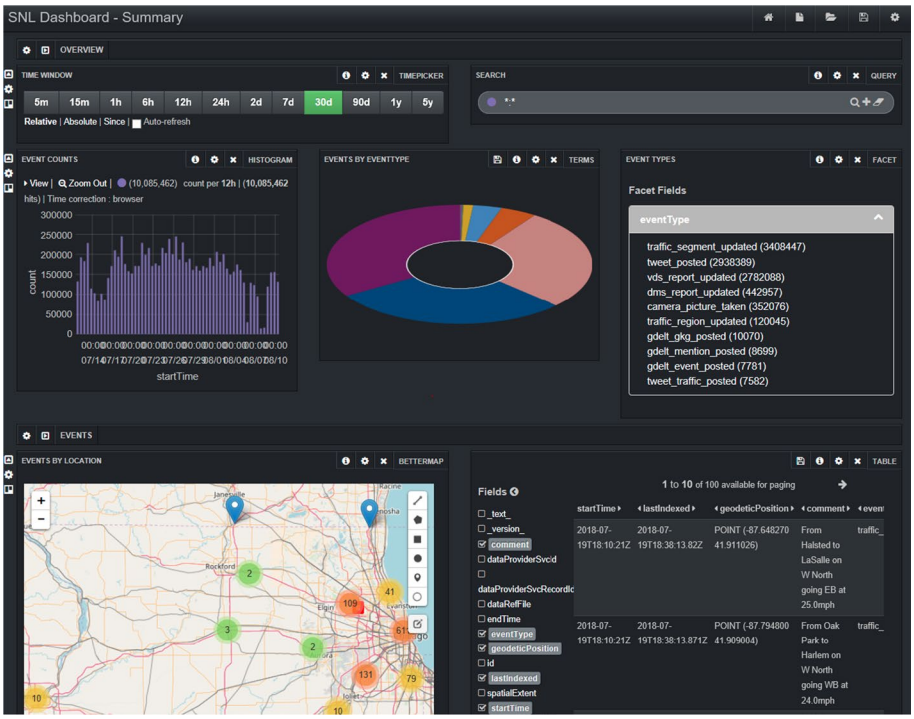


Fig. 18 BDAI dashboard

by downstream analytic processing. Once the data is indexed into Solr, the data is immediately available to perform any sort of analysis. Some events, such as the web camera imagery requires additional processing (i.e. using the YOLO processor). The time for data processing highly depends on the specific type of algorithm implemented. Our Merge Neural Network ("Muti-source data fusion" section) used for actionable intelligence generation performs a poll from "Solr" every 15 min. All information retrieved over the time interval are used to create actionable intelligence. The execution time for the Merge Neural Network is negligible (within a millisecond). The "polling" period is not a limitation in the architecture, but it is an adjustable parameter depending on the arrival time of each individual data sources. The polling rates for each topology is depicted in Table 4. The overall turnaround time for actionable intelligence generation is mainly driven by the availability of data sources and the frequency we poll the data since actual data processing is deemed negligible.

Performance vs requirement discussion

In regards to the original requirement as depicted in Table 2, our system has achieved the scalability and flexibility needed for Big Data processing. We have demonstrated that our system is horizontally scalable to hundreds of locations. For example, the data we ingested include: traffic segments received from 818 stations, vehicle detection system reports received from 818 stations, images received from 150 camera locations, and dynamic message signs reported from 150 stations. We ingested an average of 132,000 tweets a day, 14,000 camera images a day, and 10,000 posts from Gdelt. A comparison

Table 3 System performance

Event type	Avg num of locations	Avg record size (bytes)	Avg daily record total	Total num of records	Avg latency (min)	Avg throughput (bytes/sec)
Tweet_posted	1	3835	132,305	16,875,132	5.5	5873
Traffic_segment_updated	818		117,957	4,362,857	29.9	
Vds_report_updated	818	639	117,250	2,305,651	4.7	
Gdelt_gkg_posted		109,388	4254	1,104,740	3.0	5386
Gdelt_mention_posted		7509	4504	1,034,357	1.0	391
Camera_picture_taken	150	350,000	14,041	516,782	2.2	56,879
Dms_report_updated	150	2200	20,906	508,288	34.9	532
Gdelt_event_posted		1204	1793	217,691	1.7	25
Traffic_region_updated			4156	153,555	37.3	
Tweet_traffic_posted	1	3835	561	68,124	5.8	25
Construction_moratorium			1000	37,000		
Congestion_report_updated			74	28,092	28.6	
Incident_report_updated			238	9346		
Construction_report_updated			99	800		
Social_event_report_updated				225		

Table 4 Topology polling rates

Topology	Polling Rate
ChicagoTrafficTrackerTopology	no more frequently than 10 min (~ between 10 and 12 min)
XmlTopology	no more frequently than 10 min (~ between 10 and 12 min)
MoratoriumTopology	24 h
CamerasTopology	15 min
GDELT	15 min
MapQuestTopology	5 min
TweetTopology	every 5 or 15 min subject to twitter rate limits

breakdown of the statistics for requirement analysis is depicted in Table 5. The majority of the data met our requirement specification. The only exception is the dynamic message sign report topology. The larger latency was associated with an inconsistent update interval provided in the server rather than the actual latency in our system. As depicted in Table 5, the overall latency performance of each data types are largely driven by external site restrictions on how frequent we are allowed to query the data. Despite this restriction, most data sources had an average latency less than the "polling" time. It is possible that the latency can be further reduced if the data can be pushed to the

Table 5 Latency performance vs requirement

Event type	Avg Record Per Day	Num of Source Station	Query Freq (min)	Polling Freq (min)	Average throughput (bytes/sec)	Average latency (measured)	Status
Tweets	132 K	1	5 or 15 (subject to rate limit)	5 or 15 min subject to rate limit)	5873	5.5 min	Met Goal
Tweet Traffic Posts	0.5 K	1	5 or 15 (subject to rate limit)	5 or 15 (subject to rate limit)	25	5.8 min	Met Goal
Camera Images	14 K	150	15	15	56,879	3 min	Met Goal
Gdelt Global Knowledge Graphs	4.2 K	1	15	15	5386	3 min	Met Goal
Gdelt Mention Posts	4.5 K	1	15	15	391	1 min	Met Goal
Gdelt Event Posts	1.7 K	1	15	15	25	1.7 min	Met Goal
Dynamic Message Sign Report	20.9 K	150	15	10–12	532	34.9 min	Failed

consumer at a higher rate. Evaluation of this architecture using a different application exemplar with real-time accessible data would be left for future exploration.

Conclusion

In conclusion, our big data architecture provides a framework for machine-learning algorithms to learn and analyze streaming data (e.g. near real-time analytics) from heterogenous data sources (texts, signal waveforms, images, videos) to turn them into actionable information for decision makers. Our data-agnostic solution is accomplished by mapping different data types into a common frame of reference that requires both temporal and geospatial metadata. We have demonstrated through a traffic prediction exemplar that our architecture can support actionable intelligence generation in near-real-time using disparate data sources. Our traffic prediction exemplar allowed us to test and validate key BDAI capabilities: handling heterogenous data sources, hosting data pipelines on distributed processing platforms, and running machine learning algorithms in near-real-time. Our BDAI platform was designed with flexibility in mind, allowing us to quickly onboard new data sources and apply machine learning algorithms. Our data platform's agility and common frame of reference allows us to rapidly provide Actionable Intelligence to our customer's mission relevant problems. The framework architecture is a generalized architecture that can enable solutions for other BDAI problems with similar data diversity and data volume. The BDAI architecture has been fully implemented into a software system that is currently running and is hosted at Sandia National Laboratories for over a year. Our work has been featured on the local news media [1]. The current BDAI system can produce first order of data analytics (i.e. combining data from multiple source to assess what is happening at current time). In the future, we plan to further develop statistical techniques such as minimum variance to optimize the resultant estimate. In addition, we plan to extend BDAI's capability to include a second order

of analytics by providing the decision maker with a list of suggested actions, based on the assessment of the current situation using multiple data sources.

Authors' information

Tian Ma is a Distinguished R&D Computer Scientist at Sandia National Laboratories. He has over 17 years of experience in data analysis, data processing, and data exploitation of remote sensing systems, especially in the nuclear nonproliferation arena. He is a nationally recognized expert in detection algorithms and a pioneer in the field of tracking systems where he has innovated and delivered state-of-the-art real-time detection and tracking algorithms to operational systems that have solved U.S. government technical challenges and provided new, needed mission enabling capabilities. He has numerous patents and is honored with multiple national awards. He received a B.S. in Computer Engineering and an M.S. in Electrical and Computer Engineering from the University of Illinois at Chicago and an M.B.A. in Management of Technology from the University of New Mexico. Rudy Garcia is a Distinguished R&D Computer Scientist at Sandia National Laboratories. He has over 25 years of experience in software development, engineering, and data architect of software data systems. From his early days at IBM developing Fiber Distributed Data Interface (FDDI) simulators to most recently integrating multiple sensor modality detections into several different cloud environments, Rudy's focus has always been on the data and how to engineer software systems to acquire, transform, store, and retrieve the data. Rudy has been engineering a number of software systems that would collect data, transform the data to standardize formats, load the data into a data store and make available for efficient retrieval. He received a B.S. in Computer Engineering the University of New Mexico and an M.S. in Computer Engineering from Boston University. Forest Danford has been a software engineer for the last 5 years at Sandia National Laboratories. He enjoys big data problems, data visualization, and developing analytics to serve the national interest. He received a B.S. in Computer Science, a B.S. in Biosystems engineering, and an M.S. in Computer Science from the University of Arizona. Laura Patrizi is a Senior Software Engineer at Sandia National Laboratories. Her interests include software systems design and development, big geospatial data processing, and data fusion. She received a B.S. in Computer Science and Mathematics from Colorado State University and an M.S. in Computer Science from the University of New Mexico. Jennifer Galasso has spent her professional career at the Naval Research Laboratory, Washington, DC, and Sandia National Laboratory, Albuquerque, NM, writing software in domain areas including Marine Geosciences, Nuclear Weapons, Machine Learning, and Satellite-based communication. She received a B.S. in Computer Science and a B.S. in Geology from the University of Maryland, College Park, as well as an M.S. in Geology from George Washington University. Jason Loyd worked for the USDA Forest Service before beginning work at Sandia National Laboratories in 2013. He is interested in resilient systems and applying computer science theory and software engineering principles to software development. He received a B.S. and an M.S. degree in computer science from the University of New Mexico.

Abbreviations
SNL: Sandia National Laboratories; BDAI: Big Data Actionable Intelligence; LSTM: Long Short-Term Memory; IoT: Internet of Thing; GPS: Global Positioning System; HDP: HORTONWORKS Data Platform; HDFS: Hadoop Distributed File System; IF: Information Fusion; YOLO: You Only Look Once.

Acknowledgement
We would like to thank the University of Illinois Applied Research Institute for their support of various data pipeline development. We would like to thank Professor Minh Do from University of Illinois for his contribution to the vehicle detection algorithm. Would like to thank Professor Ramavarapu "RS" Sreenivas for this contribution to the robust route planning algorithm of this project. We would like to thank team members from the SNL's science & engineering computing environment department for providing infrastructure support for hosting the Big Data for Actionable Intelligence (BDAI) system. We would like to thank Thushara Gunda (Systems Research Analyst – SNL) for her support of creating the topology for Digital Globe. We would like to thank Timothy Draelos (Deep Learning Expert—SNL) for consulting with us on machine learning techniques for disparate data source. Finally, we would like to thank Vanessa Whittemore (Office Administrative Assistant—SNL) for proofreading and editing of the document.

Authors' contributions
TM and RG were the co-principal investigators of this project. They contributed to the original concept and design of this project as well supported analysis, and interpretation of results. FD, LP, JG, JL were key team members that have made significant contributions to the implementation, integration, creation, and hosting of the BDAI software. All authors read and approved the final manuscript.

Funding
This work of this paper is funded by Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Availability of data and materials
The data sources described in this paper is publicly available in references: [11, 12, 13, 14, 15, 16].

Competing interests
The authors declare that they have no competition interests.

Author details
[1] Sandia National Laboratories, Albuquerque, NM 87185, USA. [2] Livermore, CA 94550, USA.

Received: 1 May 2020 Accepted: 4 November 2020
Published online: 23 November 2020

References

1. Sandia Labs News Service. "Wrangling Big Data", *Albuquerque Journal*, November 4, 2019. https://www.abqjournal.com/1386752/wrangling-big-data-to-locate-actionable-info-a-lot-faster.html
2. Reinsel D, Gantz J, Rydning J. *Data Age 2025 - The Digitization of the World From Edge to Core.* Framingham, MA: International Data Corporation (IDC). 2018. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf
3. Ma P, Sun X. Leveraging for Big Data Regression. Wiley Interdisciplinary Reviews: Computational Statistics. 2015;7:70–6. https://doi.org/10.1002/wics.1324.
4. Qiu J, Wu Q, Ding G, et al. A survey of machine learning for big data processing. EURASIP J Adv Signal Process. 2016;2016:67. https://doi.org/10.1186/s13634-016-0355-x.
5. Majumdar J, Naraseeyappa S, Ankalaki S. Analysis of agriculture data using data mining techniques: application of big data. J Big Data. 2017;4:20. https://doi.org/10.1186/s40537-017-0077-4.
6. B. Chandramouli J, Goldstein, Duan S. Temporal Analytics on Big Data for Web Advertising. In: *2012 IEEE 28th International Conference on Data Engineering*, Washington, DC, 2012, pp. 90–101. https://ieeexplore.ieee.org/document/6228075
7. Rathore MM, Ahmad A, Paul A, Rho S. Urban planning and building smart cities based on the Internet of Things using Big Data analytics. Comput Netw. 2016;101:63–80.
8. Zhou D, et al. Distributed Data Analytics Platform for Wide-Area Synchrophasor Measurement Systems. In: IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. 2397–2405, Sept. 2016. https://ieeexplore.ieee.org/iel7/5165411/5446437/07420696.pdf
9. National Spatial Data Infrastructure (NSDI), "Presidential Documents", Federal Register. Vol. 59, No. 71 Wednesday, April 13, 1993. https://www.archives.gov/files/federal-register/executive-orders/pdf/12906.pdf
10. Waze. https://www.waze.com/
11. Twitter Data Source. https://twitter.com/?lang=en
12. Travel Midwest Data Source. https://www.travelmidwest.com
13. City of Chicago Data Source. https://www.chicago.gov/city/en.html
14. GDELT Data Source. https://www.gdeltproject.org/
15. Mapquest Data Source. https://www.mapquest.com/

16. Digital Globe Data Source. https://www.digitalglobe.com/
17. Necula E. Dynamic Traffic Flow Prediction Based on GPS Data. In: 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, Limassol, 2014, pp. 922–929. https://ieeexplore.ieee.org/document/6984576
18. Lv Y, Chen Y, Zhang X, Duan Y, Li NL. Social media based transportation research: the state of the work and the networking. In: IEEE/CAA Journal of Automatica Sinica, vol. 4, no. 1, pp. 19–26 2017. https://ieeexplore.ieee.org/document/7815548
19. Barros J, Araujo M, Rossetti RJF. Short-term real-time traffic prediction methods: A survey. In: 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Budapest, 2015, pp. 132–139. https://ieeexplore.ieee.org/abstract/document/7223248
20. Hu H, Wen Y, Chua T, Li X. Toward scalable systems for big data analytics: a technology tutorial. IEEE Access. 2014;2:652–87. https://doi.org/10.1109/ACCESS.2014.2332453.
21. Marchal S, Jiang X, State R, Engel T. A Big Data Architecture for Large Scale Security Monitoring IEEE International Congress on Big Data. Anchorage, AK. 2014;2014:56–63. https://doi.org/10.1109/BigData.Congress.2014.18.
22. Chen Z, Guobin X, Mahalingam V, Ge L, Nguyen J, Wei Y, Chao L. A cloud computing based network monitoring and threat detection system for critical infrastructures. Big Data Res. 2016;3:10–23. https://doi.org/10.1016/j.bdr.2015.11.002.
23. Casas P, D'Alconzo A, Zseby T, Mellia M. Big-DAMA: Big Data Analytics for Network Traffic Monitoring and Analysis. In: Proceedings of the 2016 workshop on Fostering Latin-American Research in Data Communication Networks (LANCOMM '16). Association for Computing Machinery, New York, NY, USA, 1–3. 2016. DOI: https://doi.org/10.1145/2940116.2940117
24. Julie Z, Bo T, Victor L. A five-layer architecture for big data processing and analytics. Int J Big Data Intelligence. 2019;6:1.
25. Weiming L, Chen Z, Bin Y, Yitong L. A General Multi-Source Data Fusion Framework. In: Proceedings of the 2019 11th International Conference on Machine Learning and Computing (ICMLC '19). Association for Computing Machinery, New York, NY, USA, 285–289. 2019. https://doi.org/10.1145/3318299.3318394.
26. NIST Big Data Public Working Group (NBD-PWG), "NIST Special Publication 1500–1: NIST Big Data Interoperability Framework: Volume 1, Definitions", National Institute of Standards and Technology, California, September 2015. https://doi.org/10.6028/NIST.SP.1500-1.
27. Bello-Orgaz G, Jung JJ, Camacho D. Social big data: Recent achievements and new challenges. Inform Fusion. 2016;1(28):45–59.
28. https://www.cloudera.com/downloads/hdp.html
29. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
30. Microsoft Azure Stack. https://azure.microsoft.com/en-us/overview/azure-stack/
31. Java Programming Language. https://www.java.com/en/
32. Python Programming Language: https://www.python.org/
33. Apache Storm. https://storm.apache.org/index.html
34. Apache Kafka. https://kafka.apache.org/
35. Nasiri H, Nasehi S, Goudarzi M. Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities. J Big Data. 2019;6(1):52. https://doi.org/10.1186/s40537-019-0215-2.
36. Aung T, Min HY, Maw AH. Performance Evaluation for Real-Time Messaging System in Big Data Pipeline Architecture. 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Zhengzhou, China, 2018, pp. 198–1986, https://doi.org/10.1109/CyberC.2018.00047.
37. Apache Lucene. https://lucene.apache.org/solr/
38. Joseph R, Santosh D, Ross G, Ali F. You Only Look Once: Unified, Real-Time Object Detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. https://ieeexplore.ieee.org/document/7780460
39. Snidaro L et al. Context-Enhanced Information Fusion: Boosting Real-World Performance with Domain Knowledge. 2016. https://doi.org/10.1007/978-3-319-28971-7.pdf
40. Banana Dashboard. https://doc.lucidworks.com/lucidworks-hdpsearch/2.5/Guide-Banana.html

Publisher's Note