Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2020 SCPE. Volume 21, Issues 4, pp. 611–623, DOI 10.12694:/scpe.v21i4.1759

REAL-TIME BIG DATA ANALYTICS FRAMEWORK WITH DATA BLENDING APPROACH FOR MULTIPLE DATA SOURCES IN SMART CITY APPLICATIONS

MANJUNATHA S[*]AND ANNAPPA B[†]

Abstract. Advancement in Information Communication Technology (ICT) and the Internet of Things (IoT) has to lead to the continuous generation of a large amount of data. Smart city projects are being implemented in various parts of the world where analysis of public data helps in providing a better quality of life. Data analytics plays a vital role in many such data-driven applications. Real-time analytics for finding valuable insights at the right time using smart city data is crucial in making appropriate decisions for city administration. It is essential to use multiple data sources as input for the analysis to achieve better and more accurate data-driven solutions. It helps in finding more accurate solutions and making appropriate decisions. Public safety is one of the major concerns in any smart city project in which real-time analytics is much useful in the early detection of valuable data patterns. It is crucial to find early predictions of crime-related incidents and generating emergency alerts for making appropriate decisions to provide security to the people and safety of the city infrastructure. This paper discusses the proposed real-time big data analytics framework with data blending approach using multiple data sources for smart city applications. Analytics using multiple data sources for a specific data-driven solution helps in finding more data patterns, which in turn increases the accuracy of analytics results. The data preprocessing phase is a challenging task in data analytics when data being ingested continuously in real-time into the analytics system. The proposed system helps in the preprocessing of real-time data with data blending of multiple data sources used in the analytics. The proposed framework is beneficial when data from multiple sources are ingested in real-time as input data and is also flexible to use any additional data source of interest. The experimental work carried out with the proposed framework using multiple data sources to find the crime-related insights in real-time helps the public safety solutions in the smart city. The experimental outcome shows that there is a significant increase in the number of identified useful data patterns as the number of data sources increases. A real-time based emergency alert system to help the public safety solution is implemented using a machine learning-based classification algorithm with the proposed framework. The experiment is carried out with different classification algorithms, and the results show that Naive Bayes classification performs better in generating emergency alerts.

Key words: Big Data, Data blending, Preprocessing, Real time analytics, Public safety, Smart city

AMS subject classifications. 68T05

1. Introduction. Technological advancement in data analytics is changing the business process by enabling faster and better decisions based on real-time analytics. When data analysts can harness useful insights from data faster, it has a significant advantage in reducing costs, increasing efficiency, and profit. Extracting valuable insights from raw data in real-time is critical for many real-time applications. The demand for real-time analytics is high in recent days in various fields where data-driven solutions are being used. In most of the data-driven solutions, real-time processing of data for making timely decisions can enhance the quality of service, improve the accuracy of predictions, and help in making early decisions. It is challenging for the data analysts to process data from multiple sources in real-time for a specific analytical solution. The outcomes of the analytics are more effective and accurate when more data from appropriate data sources get processed for a specific analytical solution.

In smart cities, the data gets generated continuously in real-time from different applications, devices, and social media on a large scale. The data generated from various smart applications and smart devices are of different types and formats. The valuable insights derived from the data generated within the city helps effective management and administration of the city services. The data-driven solutions are widely used in some of the smart city applications such as smart traffic management, smart parking systems, smart environment,

[*]Department of Computer Science and Engineering, National Institute of Technology Karnataka Surathkal, India-575025 (manjunatha.msh@ieee.org).

[†]Department of Computer Science and Engineering, National Institute of Technology Karnataka Surathkal, India-575025 (annappa@ieee.org).

smart policing, smart healthcare, etc. A vast amount of user-generated content within the city are analyzed for finding useful insights to enhance the services and performance of smart city applications. In turn, finding valuable data patterns in real-time greatly help in improving the performance of smart city applications and quality of service. The advancement in digitization in recent days opens up possible creation of user-generated content from various sources — analytics on all available data as input to discover valuable data patterns results in finding accurate data-driven solutions. For example, smart policing applications for public safety; collect the user-generated contents from different social networking applications and any specific smart application designed for the same purpose. Real-time analysis of the data collected from these data sources helps in early predictions and monitoring of crime-related incidents within the city. It is a challenging task for analysts to use multiple data sources with different properties in a specific data-driven solution.

The proposed work is aiming to design a real-time big data analytics framework with a data blending approach for data preprocessing when multiple data sources as input. The motivation for this work is, when target data spread across multiple sources, analysts must use all possible data sources to find hidden patterns and discover valuable insights for more effective solutions. In smart policing applications for public safety in smart cities, data analysts are collecting the data within the city from different sources for finding crime-related data patterns that further used for crime detection and administrative decisions for crime prevention. In this scenario, many popular social media platforms used by the public and any specific applications offered by the police department are the major data sources of information. All these data generated within the city are analyzed for making better decisions or more accurate predictions for crime detection and prevention. The rapid growth of digitization in various fields created ample space for more and more new data sources, which are added regularly in the form of social media and smart applications. It is a challenging task for the data analysts to use additional data sources in their existing data-driven solutions with minimal cost and time. The proposed framework is an attempt to address the issue of blending data from multiple data sources for real-time data analytics in smart city applications for public safety.

In this work, a smart policing system for public safety in a smart city is considered where different data sources from social media and smart application data are collected and analyzed for generating emergency alerts using the proposed framework. The data blending approach proposed with the framework helps the analysts to add up any of the related data-sources of interest in the existing analytics framework. In the proposed mechanism, the analysts can use all identified data sources in real-time for making better analytics solutions without any additional delay. Section 2 of this article describes the importance of real-time analytics and data preprocessing in it, and related work carried out. Section 3 describes the design and implementation of the proposed framework for real-time analytics with a data blending mechanism for a smart policing system use case. Section 4 of this article is a discussion of experimental work along with results, and finally, Section 5 concludes and summarizes the proposed framework along with future work for further enhancement.

2. Backgound.

2.1. Real-time Data Analytics.

Real-time analytics and streaming analytics have become more prevalent in big data applications, where timely decisions are more crucial and beneficial. It is a need in many big data applications to generate results in real-time for better performance. In Real-time analytics, data processed at the very moment it arrives into the system rather than processing at a later stage from data storage wherein it gets stored. Some applications generate data continuously in real-time, which affects the outcome of the analytical results. For example, the applications for environmental monitoring need to collect real-time data such as temperature, humidity readings continuously. Real-time analytics helps the analysts to glean essential insights quickly and find the data-driven solution instantly. The critical part in real-time big data analytics is extracting valuable information from the incoming data as and when it enters an existing big data infrastructure. The predictions or decision making in these applications are affected by both historical data stored and real-time data gets generated continuously. Real-time analytics technologies and processes must be capable of manage and analyze the data as and when it enters the database.

Big data analytics research resulted in many real-time and streaming analytics tools such as Apache Storm, Apache Spark, and Apache Flink. In Spark, data stream represented as a sequence of Resilient Distributed Datasets (RDDs) and in-memory computing feature enables it to compute data batches quicker than Hadoop. Apache Storm is another distributed computing framework for streaming data processing, but there are limited streaming machine learning libraries are available. Apache Flink is brought as an alternative for Spark with its defining traits as real-time processing and low data latency. Spark processes chunks of data known as RDDs, whereas Flink can process rows after rows of data in real-time.

2.2. Smart city and Public safety. Urban development is a crucial issue for any government as the urban population is increasing around the world in recent days [1]. Smart cities are one of the frontline projects in most of the countries for urban development. While 'smart city' means different things to different people, one common thing everyone agrees on is that digital technologies are used in the smart city to improve the quality of the services within the city. The technological growth in digital and communication media incorporated in the smart city for better services within the city. Internet of Things (IoT) and Information Technology (IT) help to accomplish many smart applications in smart cities. It leads to generating a massive amount of data in distinctive formats. The advancement in big data technologies exploits to analyze the data generated within the city for enhanced services in the smart city. The smart applications used in smart cities such as smart traffic, smart environment, smart governance, smart agriculture, smart health-care generates a large amount of data, which can be used to extract useful insights to enhance the quality of the service. The different types of sensors, video surveillance cameras, and smart mobile applications used by smart city applications are the major source for generating data. The smart applications created for smart city services and many popular social media applications are cause for generating large amounts of user-generated content within the city, which helps in enhancing the quality of service within the city.

Smart policing solutions are widely used for public safety due to the technological adoption of the Internet of Things and Cloud [2]. Transport and traffic security, infrastructure security, emergency services for fire and medical, crisis management, and law enforcement are the most common solutions in smart city public safety services. Real-time information is crucial for better implementation of such applications to provide timely services. Real-time crime centers are established in some cities to keep the cities safe by monitoring the activities in real-time within the city. Intelligent analytics on real-time data generated within the city is the solution for smarter crime responses, monitoring, and prevention. Law enforcement agencies are switching towards predictive policing for their routine and investigation procedures. It involves advanced analytics techniques to predict what and where an incident likely to happen. Predictive policing in real-time can help in early monitoring of the crime and preventing it before it happens.

2.3. Data Preprocessing in Big Data Analytics. Data preprocessing is a crucial and significant phase within the data analytics process [3]. The raw data used as input into the analytics system is likely to be noisy, inconsistent, and imperfect. The data preprocessing phase is the set of techniques used for making raw data as analytics-ready in the data analytics process [4]. The preprocessing phase in real-time data analytics becomes challenging, where the raw data enters into the data collection system continuously. The critical part in data preprocessing includes mainly two concepts, such as data cleaning and feature engineering. Data preprocessing is essential for achieving better accuracy and performance in the analytical model. Most of the effort made in the preprocessing of big data is mainly focus on developing feature selection methods [5]. Noise reduction, instance reduction, and missing values imputations are the important preprocessing methods focused by data analysts. When the data is collected from various sources, combining this to form a consistent data is an important process in making the data ready for analysis. Data blending is a technique in preprocessing for combining data from multiple sources to create a common data set for decision-making [6]. It is one of the quick methods to extract common information from multiple data sources.

2.4. Related Work. Big data has been a progressive aspect of the industries due to data explosion, which inhabited all business categories from the past few years. The academic and industry research produced many applications using real-time big data analytics in the area of healthcare, fraud detection, smart grid, social media analytics, sensor data analytics, and many more. The majority of the work is on social media data analytics for knowledge discovery. Congosto et al. [7] proposed a cost-effective framework to perform micro-blogs analytics in twitter stream data. An event detection system is incorporated to detect important events in real-time from twitter data streams is proposed by Hasan et al. [8]. Similar work has done for city event detection for London city using twitter data streams by Zhou et al. [9]. An adaptive filtering algorithm is proposed by Fan et al. [10], to filter interesting tweets from the twitter stream concerning user interest profiles. A medical emergency system is proposed by Rathore et al. [11], to find the intelligent decision by analyzing medical data collected from sensors attached to the human body. Charlie Catlett et al. [12] proposed a spatio-temporal crime forecasting model to detect high-risk crime regions using an auto-regressive model. Pina-Garcia et al. [13] proposed that data generated from different social media platforms can be integrated to enhance big data-driven models for crime prediction. Harnessing multi-source data about public sentiments and activities for informed design is proposed by Linlin You et al. [14] that addresses the process from data collection to data visualization. Zheng Xu et al. [15] proposed a framework for collecting and analyzing data from social media and surveillance cameras to describe public safety events.

Similarly, many works have been attempted for the safety of the city using real-time data. The real-time event detection system is designed to detect and classify the events for high way traffic data by Khazaei et al. [16]. An event detection system designed for real-time data analytics of IoT enabled communication system by Ali et al. [17]. A real-time monitoring system using social big data is proposed for disaster management by Choi and Bae [18]. A real-time road traffic monitoring system is proposed by Wang et al. [19], estimating online vacancies on the road using a traffic sensor data stream. Real-time data analytics for predictions are used in many data-driven applications. A prediction system designed using real-time news data sources to predict future terrorist incidents is proposed by Toure and Gangopadhyay [20]. A predictive model is proposed by Zhang and Yuan [21] for air quality monitoring by analysis of real-time meteorology data from Beijing city. Some of the work attempted for detecting, and monitoring of the crimes are forecasting crime trends in urban areas by Cesario et al. [22], spectral analysis of crimes in the city by Venturini and Baralis [23], Parvez et al. [24] and an intelligent solution for the smart city using real-time crime analysis by Ghosh et al. [25].

From the past few years, the research articles on streaming data analytics have been highlighted the need for the preprocessing mechanism of the data collected in the streaming manner for the analytics [26]. The different technological frameworks have been used for streaming data analytics. Fernando Puentes et al. [27] analyzed characteristics of different open source frameworks available for streaming data analytics. Finding proper preprocessing mechanisms for data in motion in real-time is essential in achieving better performance. Whenever analytics is performed immediately after data is collected, the preprocessing mechanism to be done as soon as the data enters the system. Data preprocessing is becoming a critical methodology for knowledge discovery in streaming data [5]. The authors identify the role of data preprocessing methodologies in the streaming analytics system for better performance. The critical preprocessing methodologies include data reduction, incremental learning, concept drift detection, and adaptation, and stream discretization algorithms. The data preprocessing with manual intervention is of no use for any better analytics system [28]. The authors use an adaptive preprocessing mechanism for prediction on real-time sensor data.

3. Design and Implementation.

3.1. Real-time Analytics Framework. Real-time Big data analytics is an iterative process. Any real-time analytics design is broadly based on one of the two important data processing architectures proposed by Lambda [29] and Kappa architecture [30]. Figure 3.1 shows the overview of Lambda architecture for real-time analytics. Lambda architecture is a data processing technique in a big data environment consisting of three layers, namely batch layer, serving layer, and speed layer. In this architecture, the data enters into the system is passed through both batch layer and speed layer. The batch layer is responsible for managing the master dataset and pre-compute the batch views, while the speed layer is responsible for calculating real-time views with real-time data. The serving layer task is to index and expose the pre-computed batch views for queries to be executed. A query to be executed can be answered through combined results of both batch and real-time views. Kappa architecture is a data processing architecture that is an alternate and simplification of lambda architecture. This architecture targets only on data as a stream, so batch layer as in lambda architecture is removed. It comprises of real-time layer and serving layer. Real-time input data is streamed through the real-time layer and results of which passed into the serving layer for queries to be executed.

In the proposed work, the real-time data from multiple sources are analyzed to discover useful insights for making real-time decisions and predictions. The data processed at the moment is stored for further use in predictive models in later stages. The framework is designed based on Lambda architecture. The data is ingested into the analytical system immediately after it gets generated at the particular source and preprocessed it to make it ready for further analytics. The data from identified sources streamed through the real-time layer

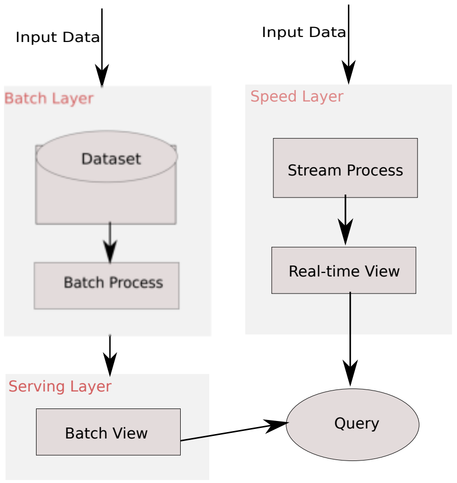


FIG. 3.1. *Real-time analytics architecture*

where it is processed and passed into the serving layer. The real-time queries to be executed using the real-time views of the serving layer. The data stored for further use is executed using the batch views along with the real-time views in the data-driven models.

3.2. Proposed Design. The proposed design for real-time analytics using multiple data sources is as shown in Figure 3.2. The real-time data from identified data sources are collected and processed for a specific data-driven solution. When the input data required for the analysis are identified at different sources, it is essential to use all available data in the process to increase the accuracy or performance of the data-driven solution. Here raw data from multiple sources in real-time are used as input data for the specific data-driven solution. The data is ingested from a particular source as soon as it is generated at the source. The data ingestion phase consists of different data ingestion processors for each input data source used. Each data ingestion processor is comprised of a real-time data ingestion mechanism for the specific data source. The processor also includes an initial stage of preprocessing mechanism for filtering of data of interest for the desired analytical solution. The data ingested and filtered at each source is passed through a data blending

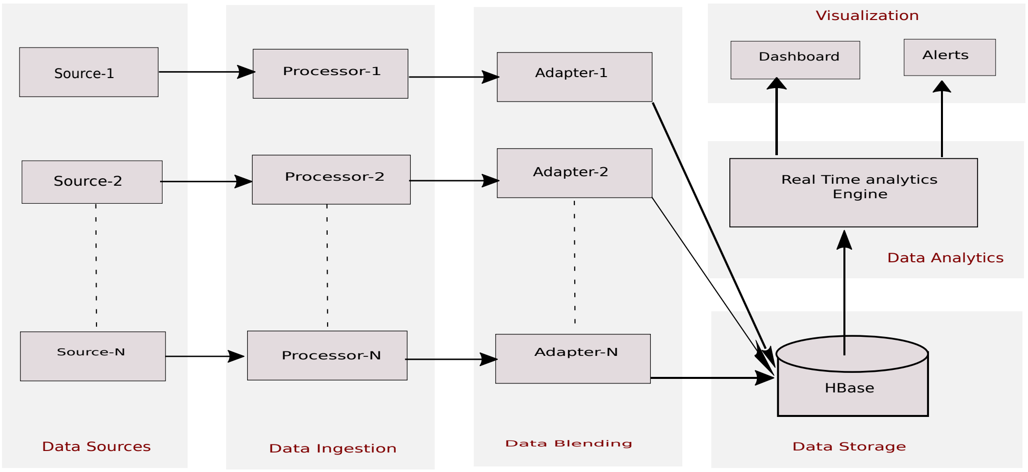


FIG. 3.2. *Proposed framework for Real time big data analytics*

mechanism. The purpose of the data blending mechanism is to integrate the data from different sources to a single common dataset for further analysis. The data blending phase consists of separate adapters for each source, which reads the input from respective data ingestion processors. Each adapter is a real-time task that can read the data immediately when it filtered out from the respective processor. Data blending is performed to extract the common data of interest from each source and append it to a single dataset. The blended data is used in the next stage for analysis to find meaningful patterns in real-time. The outcome of this helps in making data-driven decisions such as emergency alerts of crime incidents, identifying crime hotspots, and prediction of possible occurrences of crimes in the city.

4. Experimental Evaluation. Figure 4.1 illustrates the detailed framework and the flow of the real-time analytics process. Three different data sources identified as input data for experimental work, where data collected in real-time. The data from each source is ingested and filtered by respective ingestion processors and then passed into corresponding adapters for data blending mechanism. Each adapter is comprised of a mechanism to read the data from the respective processor whenever new data arrives at the processor. When the processor passes the new data to the adapter, a real-time job is executed to preprocess the data with a data blending mechanism and store it on the HBase table on top of Hadoop. Here HBase supports real-time read/write access to the data. The preprocessed data stored on the HBase table is a blended data from multiple sources that can be used in the further process for a real-time analytical solution to make desired data-driven decisions. A real-time emergency alert mechanism also introduced during the data blending mechanism to generate alerts on any emergency incidents. The contents of the ingested data from the processors are verified for any topics related to the emergency events. The input data stream is processed by the event processor to generate any emergency alerts.

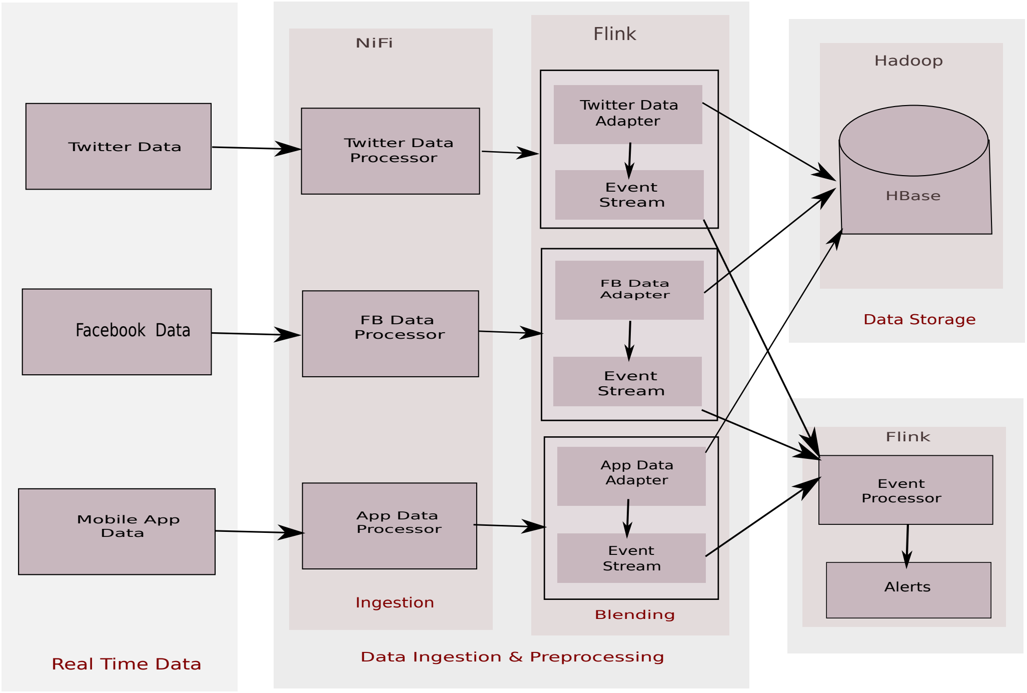


Fig. 4.1. *Data blending of real time data from multiple sources*

The critical approach in the proposed work is the data blending mechanism for preprocessing the data. Here data from multiple sources prepared ready for further analytics process. Data preprocessing is a critical step in the analytics process as it takes the maximum time of the entire process. The quality of the analytical result purely depends on the quality of the data used. Preprocessing the input data with appropriate preprocessing mechanisms is necessary. In the proposed work, analytics to be performed in real-time where it is a challenging task to preprocess the data as the data arrives continuously at data collection end. Preprocessing is to be done whenever new data ingested into the system. In the proposed mechanism, the data from multiple sources are used as input, whereas kinds of literature referred to are targeting the single source of data. When data from various data sources used in the analytics, each source may consist of data in different formats, structures. The proposed design mainly consists of three components like processors for data ingestion, adapters for data blending mechanism, and event processor to generate emergency alerts. Data ingestion processors are responsible for data collection in real-time and also the necessary filtering of expected data in real-time from the data sources. Adapters for data blending mechanism are to preprocess the data for making it ready for analytics and append into blended data. The purpose of an event processor is to analyze the incoming data streams sent from the adapters to detect any emergency incident in the city.

Data Ingestion Processors. For the experimental work, real-time data from Twitter, Facebook posts, and citizen complaint data from the mobile application are used as input data sources. For real-time data collection, separate data ingestion processors are written for each of the data sources, where each processor is performing the task of real-time data ingestion along with the initial stage preprocessing of data. The incoming data is filtered in the initial stage of preprocessing to extract only the data related to crime. For example, in the case of Twitter data, only the tweets related to the crime are considered as required data for our analytics process, and other tweets are discarded. The workflow of each processor is as shown in Figure 4.2. Each processor is configured for the respective data source with the data ingestion mechanism of real-time data. The processors are responsible for real-time data ingestion into the analytical system as and when new data generated at the source. In data ingestion processors, data filtering is done to refine the data to select only the crime-related data of the specific city and discard any other unrelated data streams. If the values of location in the incoming data match with the city location values, then the data is considered for the further process; otherwise, data discarded directly. Further, the actual content of the accepted data is verified for having any information related to crime.

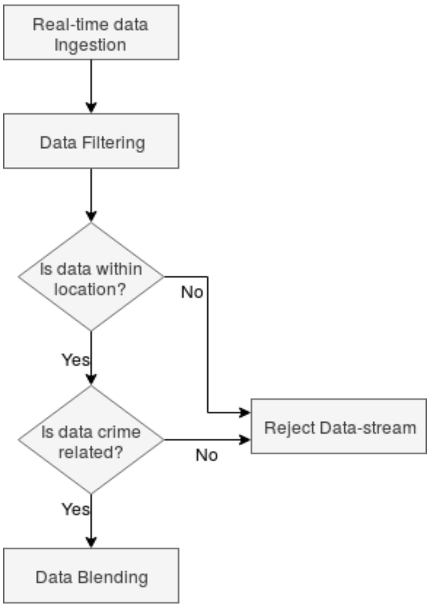


Fig. 4.2. *Processor for Crime data filtering*

The incoming data streams filtered based on city location values and further verified for whether contents of the incoming data related to crime or not. A knowledge-base is created, which consists of crime-related words and phrases to compare with the incoming data to find out any crime-related information is present in the incoming data. For experimental work, 565 words and phrases which are related to different categories of crimes are used in the knowledge base with the help of Cambridge and Macmillan dictionaries. The contents of the knowledge base are used to verify the crime-related information in the contents of the incoming data stream. If any matching information present in the incoming data, then the data stream is passed to the next stage of preprocessing. The outputs of the processors are passed through respective adapters for data blending mechanism.

Data blending mechanism. Real-time data ingested from each source by respective data ingestion processors are passed to respective adapters. Each adapter process the incoming data from the respective processor with the data blending mechanism. These adapters are the real-time jobs written using Apache Flink as the real-time processing tool. With its streaming architecture, Apache Flink helps to process the events in real-time with consistently high speed with low latency. In this experiment, all three data sources used for the analytics streamed from the data ingestion mechanism are in javascript object notation (JSON) format, but the structure of the data is different in each source. Data blending mechanism is the process of combining the data from the multiple sources into a single dataset. Data blending is a different mechanism than the data integration process. Data blending is about working with multiple data sources by preparing them and joining them together for a specific use case, whereas data integration typically stores as a single source in the data warehouse for a user to access.

The proposed data blending mechanism is implemented with adapters to process data streams from the respective data ingestion processors. The adapters are written as Flink jobs that can read new data from the respective processor as and when it arrives. A blending mechanism is a process of combining the data received from the different processors and store it on a particular data storage for further use. Here, we use HBase to store the blended data received from the adapters. We also added an emergency event process mechanism within the adapters. During the data blending mechanism, incoming data stream contents are observed for data patterns related to any emergency events. Such data streams are passed to an emergency event processor to generate emergency alerts.

The working of the Twitter data adapter is as shown in Algorithm 1. Here, an adapter can read the twitter data ingested and filtered at the respective processor immediately once it is available. The adapter for the twitter data source is written as a Flink job, which reads each new input JSON file from the output of the twitter data ingestion processor. This JSON file is parsed to filter the target fields, which are the useful information to be stored on blended data for further analytics. In the JSON file from the twitter data source, the values from specific fields such as created-at, name, location, and text are considered for the analytics at the next stage. This information from each of the incoming data streams used to store on the HBase table. Another information source-id is stored as ‘1’ for all the new appended rows from the twitter adapter. The source-id is to be used in the further process to find the identity of the data source. During the blending mechanism in the adapter, contents of the text in the tweet are observed to identify the target events for emergency alerts. Each incoming data is passed through an event processor to generate any emergency alerts.

Algorithm 1 Twitter data Adapter

```
1. Read datastream from Twitter data processor
2. Parse the datastream to select target fields (created-at, name, location, text)
3. Send the selected fields of datastream to event processor
4. Write the values of selected fields to new row in HBase table BLENDED_TABLE as
            valueof(source-id) <- 1
            valueof(created-at) <- Time
            valueof(name) <- User
            valueof(location) <- Location
            valueof(text) <- Contents
5. Repeat Step-1
```

The working of the adapters for the other data sources used in the experiment is also similar to the twitter data adapter. The structure of incoming data is different with different attribute names in each data source. The working of the facebook data adapter shown in Algorithm 2 is similar to the adapter for Twitter data, but the target fields selected are created-time, id, location, and message. The values of these attributes in the

Algorithm 2 Facebook data Adapter

```
1. Read datastream from Facebook data processor
2. Parse the datastream to select target fields (created-time, id, location, message)
3. Send the selected fields of datastream to event processor
4. Write the values of selected fields to new row in HBase table BLENDED_TABLE as
        valueof(source-id) <- 2
        valueof(created-time) <- Time
        valueof(id) <- User
        valueof(location) <- Location
        valueof(message) <- Contents
5. Repeat Step-1
```

input data are stored on the blended table on HBase. In this case, the source-id is stored as ’2’ for all new rows appended on blended data. During this process, the data stream with the selected attributes is passed through an event processor to detect any emergency events. Similarly, for the third data source used, an application data adapter is added where the attributes such as created-time, complaint-id, incident-location, and description are selected for further process. For this data source, source-id as ’3’ is assigned for all new rows to append on the blended table. Here the contents of ‘message’ in the input data are used for finding the patterns related to emergency events, as explained in the twitter data adapter and facebook data adapter. Similarly, one can add any other data source available for the analysis.

Event Processor. The purpose of the event processor is to process the event streams passed from the adapter to find any emergency incidents. The event processor consists of a machine learning-based classification model to generate emergency event alerts from the incoming data stream. A training model is developed by using information about different categories of crime incidents such as fire incidents, vehicle accidents, robbery, rape, murder, and gang-war. The initial training model is created using the data related to these six categories of crime incidents data. This training data set is updated regularly as the model is tested with new incoming data streams. The contents of the newly arrived data stream from any of the three data adapters are verified for any emergency incidents. When such incidents are detected during the process, an emergency alert gets generated to help in taking appropriate action by the law enforcement authorities.

As and when the new data stream is passed to the event processor, the content of the data is processed for selecting the topic feature by adopting Latent Dirichlet Allocation (LDA) [31]. Initially, non-English contents are filtered out by using a language detection library, and then stop words are filtered out from the contents. Latent Dirichlet Allocation (LDA) is used to train a topic model that can output the distribution of topics. Then, the classification model developed in the event processor is used to find any emergency incident. This model has experimented with the most popular classification algorithms to choose the better one for the most appropriate results.

In this work, the most commonly used classification algorithms in streaming data analytics such as the Naive Bayes (NB) classifier, Support Vector Machines (SVM) classifier, Logistic Regression (LR), and Random Forest (RF) algorithms are used. NB classifier is a probabilistic classification algorithm based on the application of Bayes’s theorem [32]. The model assumes that the presence of a specific feature is unrelated to the presence of any other feature. SVM classifier [33] is based on separating hyperplane according to which new samples are classified. Logistic Regression (LR) is a linear classifier that measures the relationship between the dependent variable and independent variables by determining the probabilities using a logistic function [34]. Random Forest (RF) is based on the forest construction procedure where features as at nodes grow like branches of a tree, finally combining all trees form a Random Forest model [35]. To evaluate the performance of the model, frequently used three statistical metrics like accuracy, precision, and recall are used. Out of the four different classifiers used, the NB classifier gives the most accurate results. Hence this classifier is used to generate emergency alerts in the proposed system. A detailed comparison of the classifier is provided in the next section.

5. Result and Discussion.

The performance of the four different classifiers used in the experiment for emergency events classification is as shown in Figure 5.1. The experiment targeted for emergency incidents by considering the six different categories of crimes. The performance metrics are computed for each category of crime incidents. Then, the overall measure is calculated as the average of the per class measure. Here NB classifier achieves a higher accuracy of 73%, which is a 3% improvement over RF, 5% improvement over SVM, and 8% improvement over LR classifier.

The proposed data blending mechanism helps the analysts to collect more input data in real-time. Figure 5.2 shows the observations after the blending mechanism. Figure 5.2.(a) represents the comparison of the data appended on the blended table from each data source. The values for each data source used in the experiment are calculated using source-id in the blended table. For each source-id, the total number of data updated at an hourly basis is observed. The consolidated data appended at an hourly basis is considered as data from multiple sources. The x-axis represents each hour of execution of the experiment, and the y-axis represents the number of data rows appended on the blended table related to the respective source. Similarly, Figure 5.2.(b) shows the number of crime data of different categories from different sources in the observation at a particular period. When data used from multiple sources in the experiments, analysts can benefit from processing more data to achieve better analytical results.

The proposed mechanism is beneficial whenever any new data source is available for the analytics. If any real-time data source to be considered as an additional input in the existing experiment, then one can easily

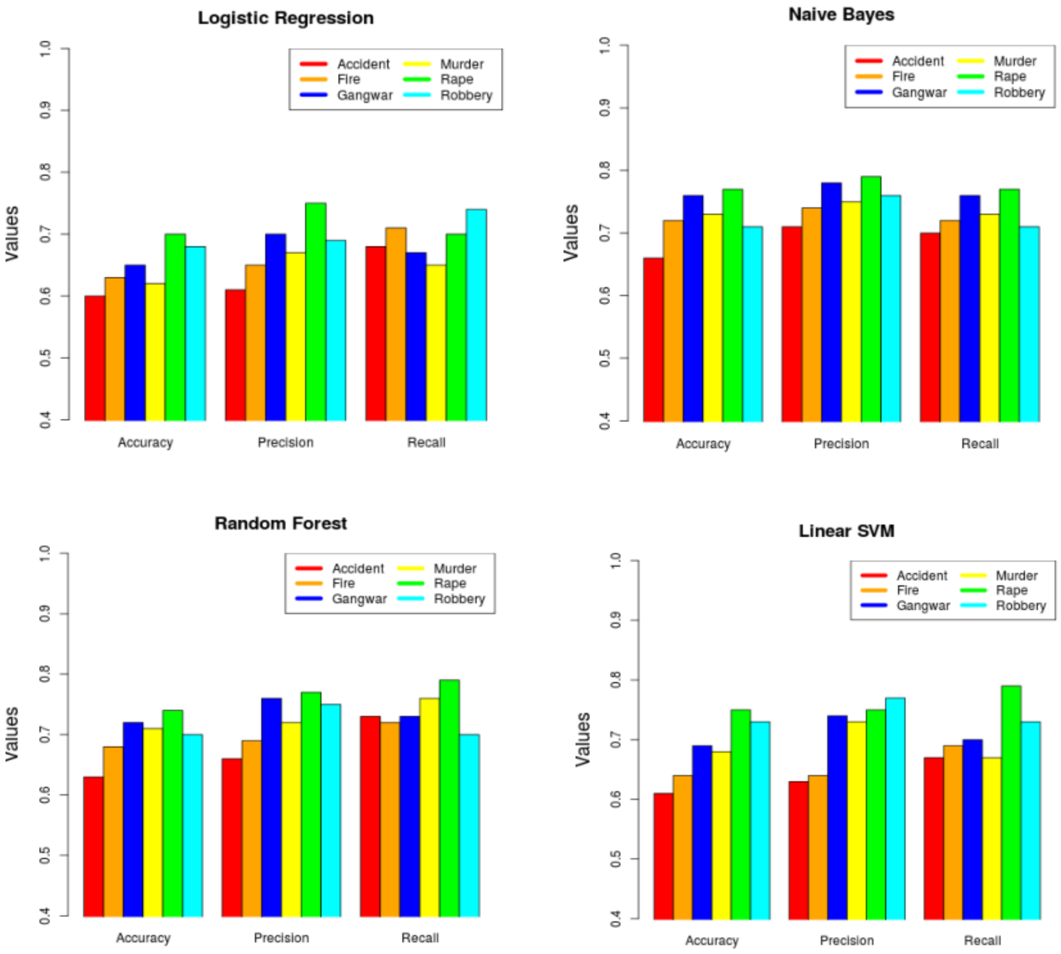
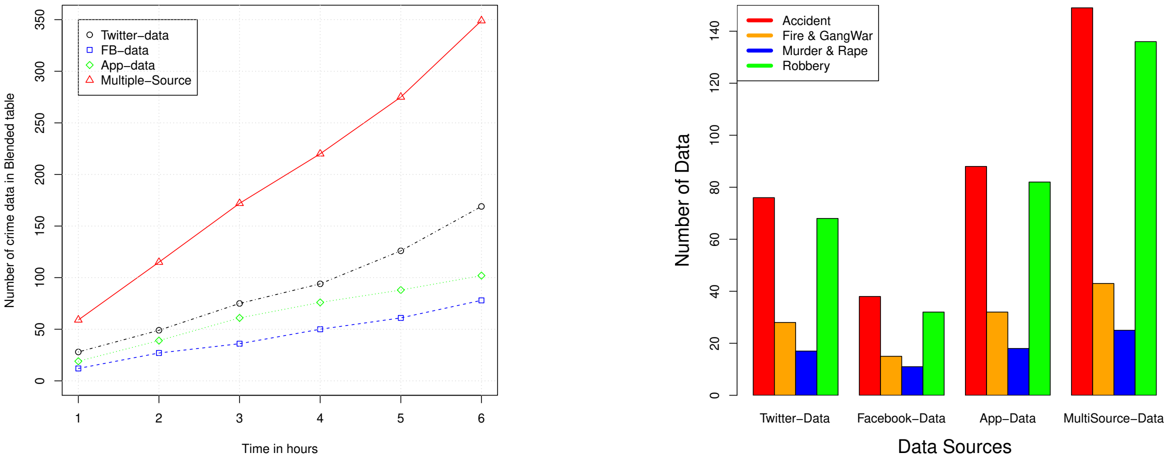


Fig. 5.1. *Performance Comparison of Classification Algorithms for Emergency Alerts*



(a) Hourly-data

(b) Source-wise data

FIG. 5.2. *Data blending of real-time data from multiple sources*

consider it for analytics by adding a new processor and an adapter for the particular data source. The new processor to be added must consist of a mechanism for ingesting the identified data source with a preprocessing mechanism, as explained in the earlier sections. Also, an adapter to be added with a data blending mechanism, as discussed in the earlier section. Each of the new data sources considered gives additional input to the event processor for emergency alerts. More and more input data collection in real-time helps the analysts to increase the performance of the analytics outcome.

6. Conclusions and future work. Real-time data analytics is invaluable in many data-driven applications for quick response and actions. Public safety is one of the key services in smart city applications, where timely decisions and predictions are much beneficial for detecting and preventing crimes. In real-time data analytics, it is crucial to perform the entire analytics process as quickly as possible. In the data analytics process, the majority of the time spent for preprocessing the data to make it prepare as analytics-ready. In real-time analytics, whenever new data arrives in the data collection phase, it must be preprocessed and analyzed for a desired analytical solution without much delay in the entire process. Collecting the maximum data for the analysis helps in achieving better outcomes. Hence, it is essential to use multiple data sources for input data in analytics to find much better and accurate outcomes. The real-time analytics framework with the data blending approach proposed in this work is appropriate to preprocess the data from multiple sources in real-time. A real-time event processing mechanism is proposed for emergency alerts to any such incidents within the city. Analytical solutions such as predictions and data-driven decisions are possibly more accurate when all available data are used instead of a single data source. The proposed mechanism is much flexible to add any new data source to be used for the analytics with the existing experimental setup.

The future work includes adopting the proposed framework with more number of input data sources. The input data used from all the data sources in the proposed work are text data ingested in the JSON format, thus future work of real-time analytics targets to use different data sources with different types of data formats and structures. The proposed data blending mechanism can be incorporated with any other data-driven applications where one can use input data from multiple sources. The classification model developed for generating emergency event alerts can be improved further for achieving more accuracy. The classification model is developed by selecting the better performing algorithm by comparing the four popularly used classification algorithms in streaming data. The future work is to use a few more algorithms for any better performance with the proposed mechanism. Further research focused on using the proposed mechanism for real-time crime hotspots predictions for public safety in the smart city. It also intended to extend this experimental setup in the crime predictions when real-time data used along with the historical data.

Acknowledgement. The authors would like to thank Ministry of Electronics and Information Technology (MeitY), Government of India, for their support in a part of the research.

REFERENCES

[1] S. Dirks, C. Gurdgiev, M. Keeling, *Smarter cities for smarter growth: How cities can optimize their systems for the talent-based economy*, in: IBM Institute for Business Value, May 2010.

[2] Market-Research-Report, *Public safety solution for smart city- global forecast to 2023*,Tech. rep., Information and Communication Technology Press Release, July 2018.

[3] D. Pyle, *Data Preparation for Data Mining*, 1st Edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[4] S. Garcia, S., J. Luengo, F. Herrera, *Data Preprocessing in Data Mining*,Springer Publishing Company, Incorporated, 2014.

[5] S. Garcia, S. Ramirez-Gallego, J. Luengo, J. M. Benitez, F. Herrera, *Big data preprocessing: methods and prospects*, Big Data Analytics 1 (1)(2016) 9. doi:10.1186/s41044-016-0014-0..

[6] Alteryx, *The definitive guide to data blending*, White Paper.

[7] M. Congosto, P. Basanta-Val, L. Sanchez-Fernandez, *T-hoarder: A framework to process twitter data streams*, Journal of Network and Computer Applications 83 (2017) 28 – 39. doi:https://doi.org/10.1016/j.jnca.2017.01.029.

[8] M. Hasan, M. A. Orgun, R. Schwitter, *Real-time event detection from the twitter data stream using the twitternews+ framework*, Information Processing and Management 56 (3) (2019) 1146 – 1165. doi:https://doi.org/10.1016/j.ipm.2018.03.001.

[9] Y. Zhou, S. De, K. Moessner, *Real world city event extraction from twitter data streams*, Procedia Computer Science 98 (2016) 443 – 448, the 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016)/The 6th International Conference on Current and Future Trends of Information and Communication Technologies in Health-care (ICTH-2016)/Affiliated Workshops. doi:https://doi.org/10.1016/j.procs.2016.09.069..

[10] F. Fan, Y. Feng, L. Yao, D. Zhao, *Adaptive evolutionary filtering in real-time twitter stream*, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, ACM, New York, NY, USA, 2016, pp. 1079–1088. doi:10.1145/2983323.2983760.

[11] M. M. Rathore, A. Ahmad, A. Paul, J. Wan, D. Zhang, *Real-time medical emergency response system: Exploiting iot and big data for public health*, Jornal of Medical System. 40 (12) (2016) 1–10. doi:10.1007/s10916-016-0647-6.

[12] Charlie Catlett, Eugenio Cesario, Domenico Talia, Andrea Vinci, *Spatio-temporal crime predictions in smart cities: A data-driven approach and experiments*, Pervasive and Mobile Computing, 53, 62-74, (2019), https://doi.org/10.1016/j.pmcj.2019.01.003.

[13] C.A. Pina-Garcia, L. Ramirez-Ramirez, *Exploring crime patterns in Mexico City*, Journal of Big Data, 6, 65, (2019), https://doi.org/10.1186/s40537-019-0228-x.

[14] L. You, B. Tunçer, H. Xing, *Harnessing Multi-Source Data about Public Sentiments and Activities for Informed Design*, IEEE Transactions on Knowledge and Data Engineering, 31, 2, 343-356, (2019),doi: 10.1109/TKDE.2018.2828431.

[15] Zheng Xu, Lin Mei, Zhihan Lv, Chuanping Hu, Xiangfeng Luo, Hui Zhang, Yunhuai Liu, *Multi-Modal Description of Public Safety Events Using Surveillance and Social Media*, IEEE Transactions on Big Data, 5, 4, 529-539, (2019),doi: 10.1109/TBDATA.2017.2656918.

[16] H. Khazaei, R. Veleda, M. Litoiu, A. Tizghadam, *Realtime big data analytics for event detection in highways*, in: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), 2016, pp. 472–477. doi:10.1109/WF-IoT.2016.7845461.

[17] M. I. Ali, N. Ono, M. Kaysar, Z. U. Shamszaman, T.-L. Pham, F. Gao, K. Griffin, A. Mileo, *Real-time data analytics and event detection for iot-enabled communication systems*, Journal of Web Semantics 42 (2017) 19–37. doi:https://doi.org/10.1016/j.websem.2016.07.001.

[18] S. Choi, B. Bae, *The real-time monitoring system of social big data for dis- aster management*, in: J. J. J. H. Park, I. Stojmenovic, H. Y. Jeong, G. Yi (Eds.), Computer Science and its Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 809–815.

[19] F. Wang, L. Hu, D. Zhou, R. Sun, J. Hu, K. Zhao, *Estimating on-line vacancies in real-time road traffic monitoring with traffic sensor data stream*, Ad Hoc Networks 35 (2015) 3–13, special Issue on Big Data Inspired Data Sensing, Processing and Networking Technologies. doi:https://doi.org/10.1016/j.adhoc.2015.07.003.

[20] I. Toure, A. Gangopadhyay, *Real time big data analytics for predicting terrorist incidents*, in: 2016 IEEE Symposium on Technologies for Homeland Security (HST), 2016, pp. 1–6. doi:10.1109/THS.2016.7568906.

[21] C. Zhang, D. Yuan, *Fast fine-grained air quality index level prediction using random forest algorithm on cluster computing of spark*, in: 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), 2015, pp. 929–934. doi:10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.177..

[22] E. Cesario, C. Catlett, D. Talia, *Forecasting crimes using autoregressive models*, in: 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), 2016, pp. 795–802. doi:10.1109/DASC-PICom-DataCom-CyberSciTec.2016.138.

[23] L. Venturini, E. Baralis, *A spectral analysis of crimes in san francisco*, in: Proceedings of the 2Nd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics, UrbanGIS '16, ACM, New York, NY, USA, 2016, pp. 4:1–4:4.

doi:10.1145/3007540.3007544.

[24] M. R. Parvez, T. Mosharraf, M. E. Ali, *A novel approach to identify spatio-temporal crime pattern in dhaka city*, in: Proceedings of the Eighth International Conference on Information and Communication Technologies and Development, ICTD'16, ACM, New York, NY, USA, 2016, pp. 41:1–41:4. doi:10.1145/2909609.2909624.

[25] D. Ghosh, S. A. Chun, B. Shafiq, N. R. Adam, *Big data-based smart city platform: Real-time crime analysis*, in: Proceedings of the 17th International Digital Government Research Conference on Digital Government Research, dg.o '16, ACM, New York, NY, USA, 2016, pp. 58–66. doi:10.1145/2912160.2912205.

[26] S. Ramirez-Gallego, B. Krawczyk, S. Garcia, M. Wozniak, F. Herrera, *BA survey on data preprocessing for data stream mining: Current status and future directions*, Neurocomputing 239 (2017) 39 –57. doi:https://doi.org/10.1016/j.neucom.2017.01.078. URL http://www.sciencedirect.com/science/article/pii/S0925231217302631.

[27] F. Puentes, M.D. Perez-Godoy, P. Gonzalez, M.J. Del Jesus, *An analysis of technological frameworks for data streams*, Progress in Artificial Intelligence(2020), https://doi.org/10.1007/s13748-020-00210-6.

[28] I. Zliobaite, B. Gabrys, *Adaptive preprocessing for streaming data*, IEEE Transactions on Knowledge and Data Engineering 26 (2) (2014) 309–321. doi:10.1109/TKDE.2012.147.

[29] N. Marz, J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*, 1st Edition, Manning Publications Co., Greenwich, CT, USA, 2015.

[30] J. Kreps, *Questioning the Lamda Architecture*, O'reilly, July 2014.

[31] D. M. Blei, A. Y. Ng, M. I. Jordan, *Latent dirichlet allocation*, Journal of Machine Learning Research.

[32] G. H. John, P. Langley, *Estimating continuous distributions in bayesian classifiers*, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 338–345. URL http://dl.acm.org/citation.cfm?id=2074158.2074196.

[33] C. Cortes, V. Vapnik, *Support-vector networks*, Machine Learning 20 (3) (1995) 273–297. doi:10.1023/A:1022627411411.

[34] I. Witten, E. Frank, M. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd Edition, Elsevier, 2011.

[35] L. Breiman, *Random forests*, Machine Learning, 5–32, (2001), doi:https:600//doi.org/10.1023/A:1010933404324.

Edited by: Rajkumar Rajasekaran
Received: May 8, 2020
Accepted: Dec 7, 2020