Automated Methodology for Context Based Semantic Anomaly Identification in Big Data

Hema.R[1], Vidhya.V[2] , Ishwarya.K.R[3] and Jeyaram.G[4]
1 Assistant Professor /CSE, Annai Vailankanni College of Engineering, hema25me@gmail.com
2 Assistant Professor /CSE, Annai Vailankanni College of Engineering, vidhyya.v@gmail.com
3 Assistant Professor /CSE, Annai Vailankanni College of Engineering, yazhini1722011@gmail.com
4 Assistant Professor/ CSE, Annai Vailankanni College of Engineering, jeyaramgj@gmail.com

ABSTRACT

Performing anomaly detection in Big Data is a difficult task. There are several existing solutions for anomaly detection. Since the existing systems are not scalable, they are not suitable for large scale systems. In this paper, we propose an automated methodology for anomaly detection which concerns not only the content of the data but also the context of the data in streaming sensor network. Additionally, we present a method of inferring invariants about the normal behaviour of dynamic data feeds. These invariants are used as proxies for specification to perform on-going semantic anomalies detection in the data feed. Finally, we show the feasibility of our approach and its usefulness for the context based semantic anomaly identification in Big Data.

Key Words: BigData, Anomaly identification, Sensor network, Semantic detection

1. INTRODUCTION

Anomaly detection is the identification of abnormal events or patterns that do not conform to expected events or patterns [1]. Anomaly detection, also called as outlier detection refers to identifying patterns in a given data set that deviates from usual manner. The patterns identified are called anomalies or intrusions. There are several existing solutions for anomaly detection. However, it is difficult to successfully find anomalies from huge amount strident, high-dimensional data, because of the limitations in existing anomaly detection techniques. Since the existing systems are not scalable, they are not suitable for large scale systems. Anomaly detection, also called as outlier detection refers to identifying patterns in a given data set that deviates from usual manner. The patterns identified are called anomalies or intrusions. There are several existing solutions for Anomaly detection. However, it is difficult to successfully find anomalies from huge amount strident, high-dimensional data, because of the limitations in existing anomaly detection techniques. Since the existing systems are not scalable, they are not suitable for large scale systems. One problem with standard anomaly detection approaches is that there is little concern for the context of the data content. One interesting, and growing, field where anomaly detection is prevalent is in Big sensor Data. Sensor data that is streamed from sources such as electrical outlets, water pipes, telecommunications, Web logs, and many other areas, generally follows the template of large amounts of data that is input very frequently. In all these areas it is important to determine whether faults or defects occur. In electrical and water sensors, this is important to determine faulty sensors, or deliver alerts that an abnormal amount of water is being consumed, as an example. In Web logs, anomaly detection can be used to identify abnormal behaviour, such as identify fraud. In any of these cases, one difficulty is coping with the velocity and volume of the data while still providing real-time support for detection of anomalies.

As Big Data requirements shift to the general public, it is important to ensure that the algorithms which worked well on small

Systems can scale out over distributed architectures, such as those found in cloud hosting providers. Where an algorithm may have excelled in its serial elision, it is now necessary to view the algorithm in parallel; using concepts such as divide and conquer, or MapReduce [3].

This paper describes a technique to detect contextually anomalous values in streaming sensor systems. This research is based on the notion that anomalies have dimensional and contextual locality. To cope with the volume and velocity of Big Data, the technique will leverage a parallel computing model, MapReduce. The primary goal of this technique is to provide a scalable way to detect, classify, and interpret anomalies in sensor-based systems.

We are interested in making the use of dynamic data feeds from online data sources more dependable. Examples of data sources include stock quotes, weather forecasts, airline ticket prices and news reports. This paper also deals with the semantic problems with the data feeds by using inferring invariants.

The following sections of the paper are organized as follows: Section II will describe related works in the field of anomaly detection in streaming sensor systems. Section III describes the approach taken by the proposed research. Finally, Section IV will describe the semantic anomaly detection concluding thoughts and ideas for future work in this area.

II. RELATED WORK

Anomaly detection algorithms can be broadly categorized as point anomaly detection algorithms and context-aware anomaly detection algorithms [1]. Hill and Minsker [2] propose a data-driven modelling approach to identify point anomalies in such a way. In their work they propose several one-step ahead predictors; i.e. based on a sliding window of previous data, predict the new output and compare it to the actual output. Hill and Minsker [2] note that their work does not easily integrate several sensor streams to help detect anomalies. This is in contrast to the work outlined in this paper where the proposed technique includes a contextual detection step that includes historical information for several streams of data, and their context. In an earlier work, Hill et al. Miller et al. [6] discuss anomaly detection in the domain of attributed graphs. Their work allows for contextual data to be included within a graph structure. [4] Proposed an approach to use several streams of data by employing a real-time Bayesian anomaly detector. The Bayesian detector algorithm can be used for single sensor streams, or multiple sensor streams. However, their approach relies strictly on the sequential sensor data without including context. Focusing an algorithm purely on detection point anomalies in the sensor domain has some drawbacks. First, it is likely to miss important relationships between similar sensors within the network as point anomaly detectors work on the global view of the data. Second, it is likely to generate a false positive anomaly when context such as the time of day, time of year, or type of location is missing.

For example, hydro sensor readings in the winter may fluctuate outside the acceptable anomaly identification range, but this could be due to varying external temperatures influencing how a building manages their heating and ventilation. Little work has been performed in providing contextaware anomaly detection algorithms. Srivastava and Srivastava [5], proposed an approach to bias anomaly detectors using functional and contextual constraints. Their work provides meaningful anomalies in the same way as a post-processing algorithm would however, their approach requires an expensive dimensionality reduction step to flatten the semantically relevant data with the content data.

One interesting result is that considering additional metadata forced the algorithm to explore parts of the graph that were previously less emphasized. A drawback of Miller et al.'s [6] work is that their full algorithm is difficult for use in real-time analytics. To compensate, they provide an estimation of their algorithm for use in real-time analytics, however the estimation is not explored in detail and so it is difficult to determine its usefulness in the real-time detection domain. [7] propose a SVM approach to multiclass classification and anomaly detection in wireless sensor networks. Their work requires data to have known classes to be classified into, and then those data points which cannot be classified are considered anomalous. One issue that the authors present is the difficulty in setting one of the algorithm's parameters. [8] have proposed work to detect anomalies by leveraging Hadoop. Hadoop is an opensource software framework that supports applications to run on distributed machines. Their work is preliminary in nature and mostly addresses concerns and discussion related to anomaly detection in Big Data. Another online anomaly detection algorithm has been proposed by Xie et al [9]. The authors address this as future work, indicating that inclusion of contextual and semantics of the data would improve the generality and detection performance of their algorithm.

III. CONTEXTUAL ANOMALY DETECTION

The proposed technique is composed of two distinct components: the content anomaly detector and the contextual anomaly detector. The content anomaly detector will be discussed in Section III-A, and the contextual anomaly detector will be discussed in Section III-B. The primary reason for creating a separation of concerns between content and context is in the interest of scalability for large amounts of data. The content-based detector will be capable of processing every new piece of data being sent to a central repository as it will use an algorithm with a fast testing time. In contrary to this, the context-based detector will be used in two situations: to help determine if the anomaly detected by the content detector is a false positive, and to randomly ensure that the sensor is not producing wholly anomalous results. The latter reason being that a sensor may be acting non-anomalous within its own history of values, but not when viewed with sensors with similar context. Section III-B will also outline the MapReduce technique to train the sensor profiles, thus determining which sensors are contextually similar. We define here a sensor profile as a contextually aware representation of the sensor as a subset of its attributes. Broadly, comparing an incoming sensor value with the corresponding sensor profile consists of comparing the incoming value with an average of all the sensor values composing the corresponding sensor profile. A pictorial representation of the sensor profile concept is illustrated in Figure 1.

Algorithm 1 illustrates the process of the technique from a component-level. This algorithm corresponds with the diagram shown in Figure 2. The UnivariateGaussianPredictor function evaluates the sensor against historical values taken from the same sensor. The function will calculate a prediction based on the previous values and compare that prediction against the actual result. An anomaly will be flagged based on the distance between the actual value and the discovered value. This will be discussed in more detail in Section III-A. The GetSensorProfile function will request the other sensors that are contextually similar to the sensor being evaluated. MultivariateGaussianPredictor then compares the sensor value with a mean value from the sensors found in the sensor profile. Again, based on the result of this evaluation, the anomaly can be rejected as being anomalous or confirmed as being both a content and context-based anomaly. Another important note is the IsRandomContextCheck function which is part of the if-statement. This will determine whether a random, perhaps non-anomalous, sensor value be sent to the context-based detector. The reason for this is primarily to check whether the sensor is behaving anomalous with respect to the sensor profile.

A. Content Anomaly Detection

Content anomaly detection, or point anomaly detection, has been well explored in literature. In particular, the proposed content anomaly detection technique will use a univariate Gaussian predictor to determine

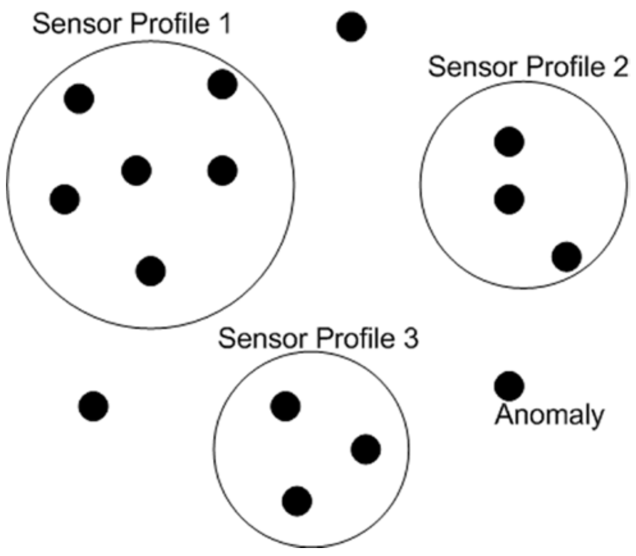

Fig1: Sensor Profile

anomalies. Univariate Gaussian predictors build a historical model of the data, and then predict and compare new values based on the model. The predictor will be univariate in that it will only consider the historical sensor readings to adjust the parameters of the model. This ensures that the predictor can classify new values quickly while sacrificing some accuracy. Speed is the most important characteristic for the point detector as it needs to evaluate a high velocity and volume of data in real-time. The accuracy short coming will be handled by the contextual anomaly detector.

```
input : $SensorValue$, $SlidingWindow$,
        $SensorHistory$
output: $Anomaly$
content ← UnivariateGaussianPredictor ($SensorValue$,
$SlidingWindow$, $SensorHistory$)
if IsAnomalous ($content$) ||
IsRandomContextCheck () then
    profile ← GetSensorProfile ($SensorValue$);
    context ←
    MultivariateGaussianPredictor
    ($SensorValue$, $profile$);
    if IsAnomalous ($context$) then
        return $Anomaly$=true;
    end
    else
        return $Anomaly$=false;
    end
end
else
    return $Anomaly$=false;
end
```

Algorithm 1: Contextual Anomaly Detection

B. Contextual Anomaly Detection

The contextual anomaly detector is based on two concepts: defining the sensor profiles and assigning each sensor to one of the sensor profiles, and evaluating the current sensor value (declared anomalous by the content anomaly detector) against the sensor profile's average expected value. The sensor profiles are defined using a multivariate clustering algorithm; the algorithm is multivariate to include the sensors multidimensional contextual metadata which may include location, building, ownership company, time of year, time of day, and weather phenomena. The clustering algorithm will place each sensor within a sensor profile and then assign that profile group to the sensor. When a sensor has been declared anomalous by the content anomaly detector, the context anomaly detector will determine the average expected value of the sensor group. Then, in a similar way as in Equation 3, the context anomaly detector will determine whether the sensor value falls within the acceptable prediction interval. The contextual anomaly detection algorithm will be learned offline, using a MapReduce [4] parallelism model to more efficiently perform on Big Data.

MapReduce programs consist of two procedures Map() and Reduce(). The Map() procedure filters or sorts the data into manageable pieces, while the Reduce() procedure summarizes the results of the Map() procedures. In the contextual anomaly detection algorithm, the Map() procedure generates small clusters for partial sets of the sensor data. The Reduce() procedure takes the small clusters and aggregates them together to produce the final set of clusters. Concretely, both the Map() and Reduce() steps in Figure 3a use a clustering algorithm to determine cluster centroids. The Reduce() step, however, only uses the centroids determined by the Map() procedure, thus reducing the computational complexity exponentially. The Map() and Reduce() steps in Figure 3 do not use a clustering algorithm. Instead, this MapReduce step is essentially re-mapping the cluster groups from the output of the first MapReduce step. For example, each initial Map() step will create k*number of Map() calls labels. The initial Reduce() step determines a set of k labels. It is the job of the second MapReduce procedure to map those k*number of Map() calls to the k labels. The first MapReduce process will use the kmeans clustering algorithm [14].

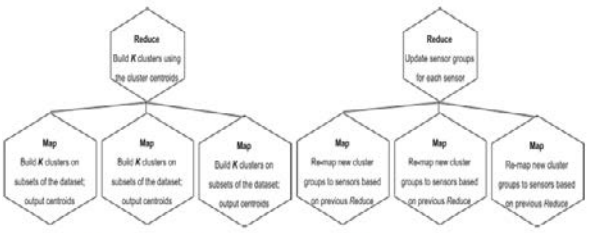
Reduce
Build K clusters using the cluster centroids
Map
Build K clusters on subsets of the dataset; output centroids
Map
Build K clusters on subsets of the dataset; output centroids
Map
Build K clusters on subsets of the dataset; output centroids
Reduce
Update sensor groups for each sensor
Map
Re-map new cluster groups to sensors based on previous Reduce
Map
Re-map new cluster groups to sensors based on previous Reduce
Map
Re-map new cluster groups to sensors based on previous Reduce
(a) Build Initial and Overall Cluster Groups (b) Re-Map Overall Groups to Initial Groups

Fig. 3: MapReduce for Determining the Cluster Groups

IV. SEMANTIC ANOMALY DETECTION

Our approach observes the behaviour of dynamic data feeds and infers invariants about their usual behaviour. These invariants can augment the existing specifications to express expectations for the behaviour of the data feed. We show that it is possible to infer useful invariants about the semantics of dynamic data feed from the behaviour and that this can be done, to a large extent, automatically. Such invariants can be effectively used to discover semantic anomalies in these data feeds.

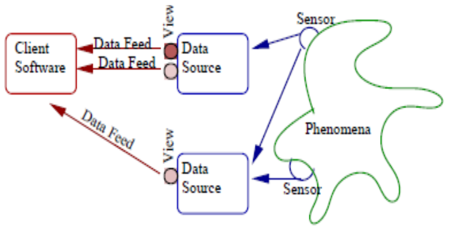

Figure 1: A client software uses a dynamic data feed that results from a view (query) on the content provided by a data source. A data source samples phenomena via sensors.

Online data sources are in particular need of improved dependability. It is hard to detect anomalies in such data sources because of their incomplete specifications. We deal with online data sources rather than programs. We use off-the-shelf techniques for inferring invariants. If complete and correct explicit specifications exist, there is no need to infer invariants. Often, the explicit specifications do not match the actual behaviour of the data

way of validating, enhancing or evolving the explicit specifications. Our approach to the difficulties of obtaining invariants is to infer a useful subset of invariants. The available observations determine relevant semantic

This approach consists of three major steps:1) initial inference of an expectation 2)Applying the expectation over unseen data to detect anomalies and 3) Updating the expectation which used to indicate an anomaly if they are broken.

$\{OBS_{i,j}(t)\}$	Criterion
(1) single data feed; stateless; comparable	internal consistency
(2) single data feed; stateful unordered	reasonable range
(3) redundant data feeds	timeliness
	accuracy
(4) correlated data feeds	inter-consistency

Table 1: Examples of determining semantic quality criteria by observation properties; numeric attributes.

.V. CONCLUSION

The work in this paper presents a novel approach to anomaly detection in Big Sensor Data. In particular, a contextual anomaly detection algorithm is proposed to enhance a point anomaly detection algorithm. To cope with the velocity and volume of Big Data, the anomaly detection algorithm relies on to find anomalies in real-time from sensor streams. This approach allows the algorithm to scale to Big Data requirements as the computationally more expensive algorithm is only needed on a very small set of the data, i.e. the already determined anomalies. Also this paper proposes a method of inferring invariants about the normal behaviour of dynamic data feeds which is effective in discovering semantic anomalies in a data feed.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009

[2] D. J. Hill and B. S. Minsker, “Anomaly detection in streaming environmental sensor data: A data-driven modeling approach,” Environ. Model. Softw., vol. 25, no. 9, pp. 1014–1022, Sep. 2010.

[3] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,”

feed. Inferred invariants could be used as a

quality criteria such as consistency, accuracy, timeliness, accuracy, reasonable range and completeness. Table1 gives examples of determining relevant criteria for data feeds of numeric attributes.

Commun. ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[4] D. J. Hill, B. S. Minsker, and E. Amir, “Real-time bayesian anomaly detection in streaming environmental data,” Water Resources Research, vol. 45, no. 4, 2009.

[5] N. Srivastava and J. Srivastava, “A hybrid-logic approach towards fault detection in complex cyber-physical systems,” in Prognostics and Health Management Society, 2010 Annual Conference of the, 2010, pp. 13–24.

[6] A. Mahapatra, N. Srivastava, and J. Srivastava, “Contextual anomaly detection in text data,” Algorithms, vol. 5, no. 4, pp. 469–489, 2012.

[7] B. Miller, N. Arcolano, and N. Bliss, “Efficient anomaly detection in dynamic, attributed graphs: Emerging phenomena and big data,” in Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on, 2013, pp. 179–184.

[8] A. Shilton, S. Rajasegarar, and M. Palaniswami, “Combined multiclass classification and anomaly detection for large-scale wireless sensor networks,” in Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on, 2013, pp. 491–496.

[9] J. R. Lee, S.-K. Ye, and H.-D. J. Jeong, “Detecting anomaly teletraffic using stochastic self-similarity based on Hadoop,” in Network-Based Information Systems (NBiS), 2013 16th International Conference on, 2013, pp. 282–287.

[10] M. Xie, J. Hu, and B. Tian, “Histogram-based online anomaly detection in hierarchical wireless sensor networks,” in Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, 2012, pp. 751–759.