Composite Event Detection and Identification for WSNs using General Hebbian Algorithm

K.Ali*, T. Anwar[o], I. H. Naqvi, M. H. Jafry
Department of Electrical Engineering, LUMS School of Science and Engineering, Pakistan
*Department of Computer Science and Engineering, Michigan State University, USA
[o]Department of Electrical Engineering, Rensselaer Polytechnic Institute, New York, USA
Email: alikamr3@msu.edu, anwart@rpi.edu, {ijaznaqvi, hassan.jafry}@lums.edu.pk

***Abstract*—In this paper, we propose an on-line technique for in-network, distributed and composite event detection and identification for streaming sensor data in resource constrained Wireless Sensor Networks (WSNs). We use General Hebbian Algorithm (GHA) to find out principal components of a multi-attribute input data which has a linear complexity as opposed to quadratic complexity with eigen value decomposition (EVD). This allows for on-line computation of percentage contributions of individual attributes towards detected event. Comparison with other event detection techniques shows that our scheme incurs low communication overhead as compared to some state-of-the-art schemes. Moreover, our hyper-ellipsoidal clustering based event detection algorithm is shown to achieve high detection rates (DRs) of over 98.88% and very low false positive rates (FPRs) of below 0.01%. Our simulation results and the hardware implementation also show that the accuracy of proposed identification scheme is in strong agreement with EVD based techniques, proving it to be a successful event identification method for WSNs.**

Keywords*—Outlier detection, event detection, event identification, Hebbian algorithm, clustering, wireless sensor networks.

I. Introduction

A wireless sensor network (WSN) consists of many energy and memory constrained sensor nodes typically equipped with sensors for monitoring multiple attributes of the surrounding environment e.g. temperature, humidity, acceleration, etc. These nodes are capable of wirelessly communicating important information to the *base-station* or *sink*. Each sensor reading can be characterized as either normal or an *outlier* if it significantly deviates from the normal pattern. *Events* however are genuine abnormalities in one or multiple attributes of interest due to an exceptional or unexpected change in environmental conditions or a hazardous condition for example a fire or gas leakage [1][2]. Outlier detection schemes are used in a *distributed* or *centralized* manner in order to detect anomalies in sensor data across WSNs.

An event detection scheme is essentially derived from an outlier detection scheme. We define an event as a sequence of outliers with spatial as well as temporal correlation in a streaming data set [3][4]. Events can be characterized as simple (atomic), composite, local or global [5], [2]. A simple event can be declared based on the observation of one attribute. However a composite event is the combination of different atomic events [2]. Similarly, local events are detected in the vicinity of a single node only whereas a global event is declared only after reception of multiple local event reports. Although event detection schemes have received some treatment in the literature [4][3][2], event identification schemes capable of determining the relative contribution of each of the attribute in a declared *event*) have rarely been investigated.

In this paper, we present a joint event *detection* and *identification* (EDI) scheme. The proposed scheme is on-line, accurate and can be used easily on resource constrained nodes of real world WSNs. Our basic outlier detection algorithm is based on hyper-ellipsoid clustering [6] which precedes the event *detection* and *identification* phase. Our event identification algorithm is based on dimensionality reduction using General Hebbian Algorithm (GHA) which allows for *on-line* computation of percentage contributions of individual attributes towards detected event as compared to Eigen Value Decomposition (EVD) which needs large batches of measurements for accurate calculation of principal components. We further provide a simple mechanism that allows detection of composite and global events in the network while keeping communication overhead very low. Our event *detection* and *identification* techniques have been simulated on real dataset. Our detection scheme is shown to achieve very high detection rate (DRs) and low false positive rate (FPR). Our on-line event identification algorithm achieves accuracy as high as that of standard *off-line* EVD based dimensionality reduction technique. The promising results of the hardware implementation of our on-line detection and identification scheme on Texas Instruments CC430 based WizziMotes [7], highlight the effectiveness and high accuracy of our proposed scheme.

In the rest of the paper we start with some related work on event detection and event identification schemes and background of the Hebbian algorithm in Section II. Section III introduces the system model and event detection and identification algorithms based on our system model. Simulation results and hardware implementation of the proposed scheme have been presented and discussed in Sections IV and V respectively. Section VI concludes the paper.

II. Related Work

A. Outlier Detection

Several surveys on characterization and classification of outlier detection techniques [1] have suggested that clustering based outlier detection techniques have got numerous advantages such as low computational complexity, ability to be deployed in an unsupervised and dynamic environments, high detection and low false positive rates and the ability to incorporate multi-variate and trendy data. *Hyper-ellipsoid* clustering is one of the popular outlier detection in WSNs [6][8][9]. It has the potential to perform equivalent to the state-of-the-art Support Vector Machine (QS-SVM) based methods [10][11] in terms of detection rates (DR) and false positive rates (FPR) [12]. These clustering techniques are computationally simpler, introduce very low latencies and can incorporate correlation between the attributes [12]. Our implementation of outlier detection is based on the *iterative* boundary estimation model mentioned in [6] called *Forgetting Factor Iterative Data Capture Anomaly Detection* (FFIDCAD). The algorithm computes the boundary of the local model for multidimensional data in an *on-line* manner. Each node performs real time adjustment to its hyper-ellipsoidal cluster boundary based on the most recent measurements. Moreover by using a forgetting factor, it allows to effectively track any non-stationary environmental behavior.

B. Event Detection and Identification (EDI)

An event is a sequence of outliers with spatial and temporal correlation in a streaming data set [3][4]. Event identification is about determining the relative contribution of each of the attribute in a declared *event*. Event detection and identification go hand-in-hand and the identification phase is invoked right after the detection of a *local* event. A computationally efficient, accurate and distributed identification technique leads to an efficient utilization of the limited energy resources of the network by decreasing the overall communication overhead.

Identification techniques based on dimensionality reduction through Eigen Value Decomposition (EVD) can effectively perform *event identification*, but EVD involves a huge computation time and large storage space [13] which makes it infeasible for a *distributed* implementation on energy constrained nodes. Moreover, its hardware implementation also entails complicated circuitry [13] in FPGA based implementations. We propose to use General Hebbian Algorithm (GHA) for event identification in WSNs. GHA was initially proposed by Erkki Oja in 1982 [14] for the modeling of neural networks. GHA algorithm offers an effective online way to compute the eigen vectors and their corresponding eigen values. GHA has previously been used in various diverse fields but to the best of authors' knowledge, GHA based *event identification* has never been studied, simulated or implemented in WSNs.

III. System Model and The Proposed Implementation

A. Outlier Detection

Consider a WSN where each node is capable of measuring n attributes and the data gets hierarchically routed to the gateway node S_g. Let $\mathbf{A_k} = \{\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_k}\}^\mathbf{T}$ be an $n \times k$ data vector where k is the total number of data samples in the memory and n is the number of attributes. The $n \times 1$ mean m_k of streaming data can be updated in an on-line manner in the following way

$$m_{a,k+1} = \frac{km_{a,k} + a_{k+1}}{k+1} \tag{1}$$

The continuously updated covariance matrix S_k is given as

$$m_{a^2,k+1} = \frac{km_{a^2,k} + a_{k+1}a_{k+1}^T}{k+1} \tag{2}$$

$$S_k = m_{a^2,k} - (m_{a,k}\, m_{a,k}^T) \tag{3}$$

Once the covariance of the data is calculated, a hyper-ellipsoid is found whose boundary encapsulates 98% of the data under the assumption that the data has a normal distribution [15]. The boundary of such an hyper-ellipsoid is given by the inverse of the chi-squared statistic with d-degrees of freedom, i.e. $t^2 = (\chi_d^2)_p^{-1}$ with $p = 0.98$. Thereafter, Mahalonobis distance of each data sample is computed and the data points lying outside the boundary of the hyper ellipsoid are declared as outliers. An outlier is declared if:

$$(a - m_k)^T S_k^{-1} (a - m_k) > t^2 \tag{4}$$

The mean $m_{a,k+1}$ can also be incrementally calculated using Eq. (5). This exponential moving average technique uses λ as a *forgetting factor* which adds *tracking* capability in the algorithm making it feasible for implementation in unsupervised dynamic environments [16]. The suggested value of λ is between 0.99 to 0.999 [16]. The covariance inverse S_k^{-1} can then incrementally be updated by using Eq. (6).

$$m_{k+1,\lambda} = \lambda m_{k,\lambda} + (1-\lambda)x_{k+1} \tag{5}$$

$$S_{k+1}^{-1} = \frac{kS_k^{-1}}{\lambda(k-1)} \times \tag{6}$$

$$\left[I - \frac{(x_{k+1} - m_{k\lambda})(x_{k+1} - m_{k\lambda})^T S_k^{-1}}{\frac{(k-1)}{\lambda} + (x_{k+1} - m_{k\lambda})^T S_k^{-1} (x_{k+1} - m_{k\lambda})} \right]$$

B. Local Composite Event Detection and Identification (EDI)

In our implementation, a local event is declared at a node if more than q consecutive outliers are detected to ensure *temporal correlation* of outliers before declaring an event. The value of q depends on sampling rate and and the required temporal resolution. For a sampling rate of 10 samples per second, we use $q = 20$ in our simulations and implementation. Event identification is invoked only after a local *event* has been declared at a node. Our algorithm carries out identification in an on-line manner by computing the eigen vectors and the corresponding eigen values of the covariance matrix of the outliers in the detected event. The eigen value decomposition of the incrementally updated covariance matrix S_o can be written as follows:

$$S_o = \Phi^T \Lambda \Phi \tag{7}$$

where Φ is a $n \times n$ matrix containing eigen vectors and Λ is a $n \times n$ diagonal matrix with corresponding eigen values.

If S_o is symmetric, positive and has an invertible square root, then the data can be whitened by finding the appropriate *weights* (projections of the data set onto the eigen vectors) E_w corresponding to the eigen values:

$$X = A - m \tag{8}$$

$$E_w = \Lambda^{-1/2}\Phi^T X = S_o^{-1/2} X = \Lambda^{-1/2} y \tag{9}$$

where A is incoming data vector, X is zero mean data vector, and $y = \Phi^T X$ gives the de-correlated data. We will shortly see that we do not need to calculate S_o because our GHA based algorithm will automatically provide us with the required eigen values and vectors.

1) Generalized Hebbian Algorithm: For a linear MIMO system with inputs $x_1(t), x_2(2), ..., x_n(t)$ of weights $w_1(t), w_2(t), ..., w_n(t)$ we have:

$$y = \sum_{i=1}^{n} w_i x_i \tag{10}$$

In case of streaming data, the right hand side of above equation leads to an infinite increment with time. Therefore, the weights w_i are normalized as follows at each iteration:

$$w_i(t+1) = \frac{w_i(t) + \eta(t) x_i(t)}{\left[\sum_{i=1}^{n} [w_i(t) + \eta y(t)]\right]^{1/2}} \tag{11}$$

where η is a small constant which determines the convergence time. For a general case equation (11) simplifies [14] to

$$w_i(t+1) = w_i(t) + \eta y(t)[x_i(t) - y(t) w_i(t)] + O(\eta^2) \tag{12}$$

Equation (12) updates the weights at every iteration with a negative feedback such that an increase in the output y decreases the weights. The algorithm converges mainly because of the fact $\sum_{i=1}^{n} w_i(t)^2$ is close to one. For a detailed mathematical analysis of the convergence of the weights to the principal components, the reader is refereed to the classic papers by Oja and Karhunen [14].

2) Using GHA for Event Identification: The eigen vector matrix Φ in eq. (9) is calculated using GHA algorithm as follows: Let $\mathbf{W(t)} = \{\mathbf{w_1(t)}, \mathbf{w_2(t)}, \cdots, \mathbf{w_n(t)}\}$ be a $n \times n$ weight matrix, initialized to some random W and η be the learning rate. Algorithm 1 summarizes the steps for calculating and updating the eigen vector matrix Φ in an on-line manner as j^{th} consecutive outlier is detected in an event.

Algorithm 1 General Hebbian Algorithm (For Eigen Vectors)

```
1: initialize random W; /*Weight Matrix*/
2: initialize η; /*Small Value of Learning Rate*/
3: initialize imax;
4: for each consecutive detected outlier X_j do
5:     X_j = A_j − m_k;
6:     for i = 1 to imax do
7:         Y = W ∗ X_j
8:         dW = [Y ∗ X_j^T] − Diag(Y ∗ Y^T) ∗ W
9:         W = W + η ∗ dW
10:    end for
11: end for
12: Return W
```

In steps 4-12, weight matrix W is iteratively updated where $Diag(M)$ is an operator that diagonalizes matrix M. The algorithm stops when $imax$ iterations have been performed. If $imax$, i.e. total number of iterations, is large enough and the learning rate is appropriate, W converges to eigen vector matrix Φ in eq. (9). We've used $\eta = 0.05$ and $imax = 100$ in our simulations and implementation on WizziMotes. After getting the eigen vector matrix Φ, the eigen values of the covariance matrix can be obtained simply as:

$$\Lambda = Diag(\Phi S \Phi^T) \tag{13}$$

Among the q consecutive detected outliers lying outside boundary of the hyper-ellipsoid ($t^2 = (\chi_d^2)_p^{-1}$ with $p = 0.98$), the latest q^{th} outlier is decorrelated by projecting onto the calculated eigen vectors ($\Phi^T X$) and then whitened ($\Lambda^{-1/2}\Phi^T X$) to get an $n \times 1$ weight matrix E_w containing the identification ratios of the individual attributes. The percentage contribution or *identification ratio* of i^{th} attribute C_i towards the detected *event* can then simply be calculated as

$$C_i = \frac{E_w[i]}{\sum_{i=1}^{n} E_w[i]} \times 100 \tag{14}$$

Based on this contribution information, now each node can decide whether the detected event is composite or atomic. Owning to a distributed implementation, no sensor data is being saved on the sensor nodes; such an implementation is quite difficult (if not impossible) with batch implementations of EVD. Moreover, with GHA the complexity with respect to the dimension of data vectors (i.e. the number of sensors on a sensor node) increases in a linear order as opposed to quadratic increase with eigenvalue decomposition. Therefore, principal components can be computed with lower computational resources as compared to GHA, which makes it more attractive for hardware implementation.

C. Event Reporting Packet (ERP) format

During the event detection and identification phase, the following packet format facilitates in transferring *composite* and *global* event reports to cluster heads. It is assumed that a node transmits its event report once a local event is detected its vicinity (an event triggered implementation). The term *cluster heads* or CHs is used for representing *parent* nodes i.e. the nodes responsible for collecting data from *children*. Following is a description of the individual fields in the proposed packet format.

- *EVENT FLAG*: A **1 bit** field indicating the presence or absence of an event.
- *NODE ID*: A variable length node identification number. The number of bits in this field is of the order of $log_2(N_n)$, where N_n is the number of nodes in the network (typically ranges between **8-11 bits** for a small scale network).
- *ATTRIBUTES*: This field contains the attributes contributing in a declared event and thus indicates the

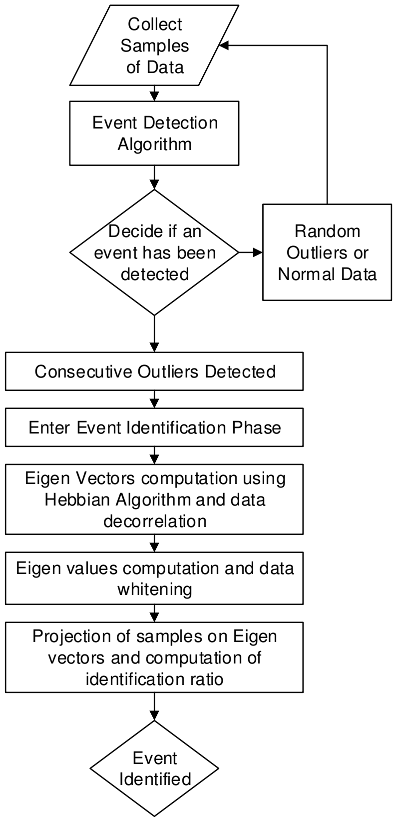

Fig. 1. Algortihm Description

presence of multi-attribute events. For instance, in case of a mine explosion, both gas and temperature will contribute toward the event. Moreover, multi-attribute events are not always perfectly correlated in time. To tackle "cause and effect" in multi-attribute events, each node waits for τ_L seconds before declaring a multi-attribute events. The parameter τ_L is environment and sensor dependent (and thus is user specified). This field requires **n bits** where n is the maximum number of attributes being monitored by any node in the network.

- *SPATIAL CONFIDENCE LEVEL*: This field of the packet is only updated by the parent nodes or CHs in case of multiple ERP receptions from children nodes (section III-D). The number of bits in this field is equal to bits in *NODE ID* field.
- *IDENTIFICATION RATIOS:* This field includes the Identification Percentages (IPs) of the detected event derived from the EDI algorithm in form of whole numbers from 0-100. Thus, if n attributes are being monitored, this part of the packet will accumulate to **7n** bits.
- *EVENT TAG:* This field is used to differentiate between multiple event reports from multiple CHs. If number of cluster heads in the network is N_c, $log_2(N_c)$ bits are required for this field.

So, the total number of bits in ERP's of leaf nodes will be $8n + log_2(N_n) + 1$ and for parent heads the packet size will be $8n + log_2(N_c) + 2 \times log_2(N_n) + 1$.

D. Global Composite Event Detection and Identification

Global event detection is invoked only if a local event has been detected. If a CH receives an ERP packet indicating an event, it performs the following steps before forwarding the ERP up the hierarchy:

1) The moment a CH receives an event packet from one of its child nodes, it starts a timer and waits for an interval τ_G to hear from other child nodes for an event report. The parent then estimates τ_G by exploiting the known positions of its children nodes. This step is performed for all child nodes (n_c times).
2) The CH then extracts the event information from the ERPs and looks for spatial and temporal correlation of the detected events.
3) It also compares the extracted information with its own event information to identify the attributes contributing towards a "global" event.
4) Each parent node increments the spatial confidence level field if multiple children report similar events, and averages out the individual identification ratios of the attributes reported by each node
5) After performing the above mentioned steps from 1-4, the CH then packetizes the aggregated information into an ERP (according to format discussed in Section III-C) and sends it up the hierarchy towards its parent node.

For a network of total 1024 nodes, 992 leaf nodes and 32 cluster heads (assuming each node measuring 5 attributes), each leaf and CH node needs just 51 and 66 bits respectively to communicate the event information up the hierarchy.

IV. Complexity Comparison

Tables I and II show comparison of the complexity and communication overhead of our proposed EDI technique with 5 QS-SVM based algorithms presented in [10], [11] and [17] respectively. All of these algorithms also consider both spatio-temporal and attribute correlations of data for global event detection. The notation for entries in Table I is as follows: n is the total number of measurements, d is number of attributes, $v << 1$ is fraction of events in data and u$\in$ $(0, 1)$ is the regularization parameter for SVM which sets the lower limit to the number of false detections in any QS-SVM algorithm. Moreover, $nv << n$ and $(nv)^3 << n^3$. In Table II, N_o is be the number of outliers and N_e be the number of events detected in the network. In our scheme, very concise event related information propagates through the network thus giving the communication complexity of order $\mathcal{O}(N_e)$ instead of $\mathcal{O}(n)$ that is the case with above mentioned SVM schemes where each measurement needs to be communicated to the base-station for processing. Note that $n >> N_e$ in real world scenarios. Moreover, WSN sensor data is floating point data and sending all the data vectors to the base-station for event detection and identification leads to a huge overhead.

V. Simulation Results and Discussion

The simulation for complete EDI process has been divided into *detection* and *identification* phases. Fig. 1 presents the

Proposed EDI approach	Computational Complexity
Outlier Detection	$\mathcal{O}(nd^2)$
Projection along d dimensions	$\mathcal{O}(nv)$
Clustering along d dimensions	$\mathcal{O}(nv)$
Decision for outlier and event	$\mathcal{O}(1)$
Total computational complexity of proposed technique	$\mathcal{O}(nd^2 + 2vn + imax * nv) \approx \mathcal{O}(nd^2)$
QS-SVM Techniques	**Computational Complexity**
ST-QS-SVM [17]	$\mathcal{O}(n^2)$
STA-QS-SVM [10]	$\mathcal{O}(n^2 + nd^2)$
STA-TASV,STA-TSV [11]	$\mathcal{O}(n^2 + nd^2)$
STA-CA [11]	$\mathcal{O}(n^2 + nud^2)$

TABLE I. COMPARISON OF COMPUTATION COMPLEXITY WITH QS-SVM BASED ALGORITHMS

Proposed EDI approach	Communication Complexity
Maximum communication overhead for parent node close to base-station	$\mathcal{O}(N_e)$
QS-SVM Techniques	**Communication Complexity**
ST-QS-SVM [17]	$\mathcal{O}(n)$
STA-QS-SVM [10]	$\mathcal{O}(n)$
STA-TASV,STA-TSV,STA-CA [11]	$\mathcal{O}(n)$

TABLE II. COMPARISON OF COMMUNICATION OVERHEAD WITH QS-SVM BASED ALGORITHMS

block diagram representation of the simulated algorithm. At first, when a data sample is collected, an on-line outlier detection algorithm determines if the current measurement points to a normal reading or an outlier. In case an outlier is detected, the algorithm goes into local event detection phase and looks for a series of outliers. If an *event* is declared (i.e. more than q consecutive outliers are detected), the algorithm enters the event *identification* phase described in Section III-B.

Simulations have been performed on real world dataset which was taken from a multi-hop WSN deployment [18] using TelosB motes, consisting of humidity and temperature measurements over a 6 hour period at 5 second intervals. The proposed event detection and identification algorithm was applied on Nodes 1 and 3 of TelosB dataset because both clearly contained an *event*. Principal components were computed with both EVD and with GHA algorithm. Two correlated attributes namely temperature and humidity have been tested. The *event* detection algorithm successfully declared the event and passed the data to the event *identification* phase. Figure 2 shows time series plot for both nodes 1 and 3 respectively along with the detected outliers. It is obvious that the proposed technique successfully detects both outliers and events. Figure 3 shows the results of event *identification* obtained. We compare the results of our GHA based identification technique with the regular EVD based approach used in [19]. The results shown by figure 3 conclude that the percentage contributions of measured using our proposed identification approach are more than 95% accurate when compared to state-of-the-art EVD based techniques [19], proving it to be a successful event identification method for WSNs. The comparison of the detection and false positive rates of our scheme with some state-of-the-art SVM based schemes is given in Table III.

VI. HARDWARE IMPLEMENTATION

It can be observed from the simulation results that the proposed *event identification* scheme successfully determines the ratio for different attributes with significantly less computational resources. The general Hebbian algorithm can be

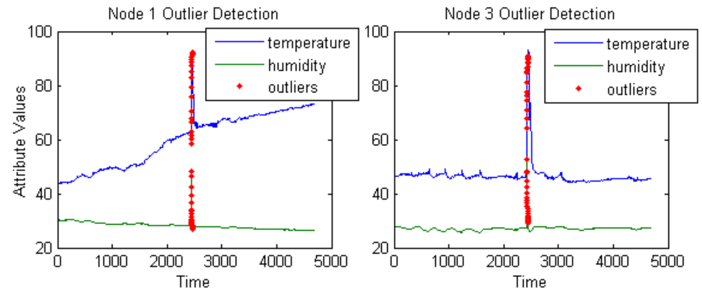

Fig. 2. Results of outlier and event detection for TelosB Nodes 1 and 3

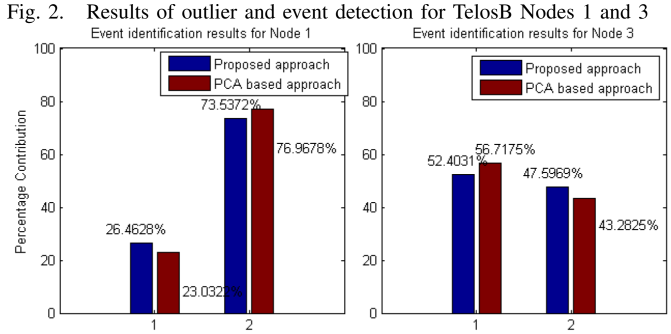

Fig. 3. Results of event identification for TelosB Nodes 1 and Node 3 alongwith a comparison with regular PCA based approach mentioned in [19]

10 times more computationally efficient when compared with conventional EVD algorithms [20]. Previous evaluations based on hardware implementations [20] have shown that upto 99.8% improvement in memory requirements, 20.5 times improvement in power consumption and 10.3 times improvement in logic resources can be achieved by the stream-based hebbian eigen filter as compared to EVD. This low complexity and computational efficiency of the presented *event identification* scheme makes it attractive for implementation in resource constrained WSNs. We use CC430 based WizziMotes [7] running DASH7 protocol [21] in order to test our proposed event identification technique. We selected two attributes namely illumination and temperature. The readings from the illumination sensor were normalized to give values close to 1 when the mote was placed in complete darkness. In the same way the readings from temperature sensor were normalized to give temperature of the atmosphere in C^o. We performed two experiments in order to test our algorithm. In both the experiments, two sensor nodes were used, one being the *sink* node connected with a PC working as a Cluster Head (CH) of the second *child* node.

The validation of EDI algorithm is explained as follows; each node updates the covariance and mean of the monitored attributes in an on-line manner to continuously check the readings for anomalous behavior on the basis of Mahalanobis distance. Once a local *event* is declared by the child node, on-line GHA based *event identification* is invoked which calculates the individual contribution of the attributes in the detected local *event* and sends the required information to the sink node using the packet format mentioned in Section III-C. The sink node decodes the event report packet, compares it with it's own event information and the results of composite

Technique	Detection Rate	False Positive Rate
Proposed EDI approach [17]	98.88%	0.01%
ST-QS-SVM [17]	16.67%	10.85%
STA-QS-SVM [10]	91.67%	0.5%
STA-TASV [11]	92.45%	0.48%
STA-TSV[11]	99%	0.9%
STA-CA [11]	91.67%	0.1%

TABLE III. COMPARISON OF DETECTION AND FALSE POSITIVE RATES WITH QS-SVM BASED ALGORITHMS

event detection are monitored on the PC through a UART interface on the *sink* node. Following are the two scenarios in which we tested our algorithm. The results obtained in both experiments are also discussed.

1) Testing Scenario No. 1: In scenario 1, the *identification* algorithm is tested by turning off the lights of the room in which the motes were deployed. The idea was to find out if the algorithm identified close to 100% contribution of the sole attribute responsible for the *event*. The results showed 1.286% and 98.814% contribution for temperature and illumination respectively confirming the effectiveness of the proposed scheme.

2) Testing Scenario No. 2: In order to find out the efficiency of the algorithm in scenarios where multiple correlated attributes contribute toward an *event*, we illuminated two incandescent bulbs in a close vicinity of both the motes. The results showed 74.87% contribution from illumination and 25.13% contribution from temperature attributes respectively which proved that both light intensity and temperature have contributed to the event in a correlated manner.

The above results show that the proposed scheme gives accurate identification ratios in cases where both independent as well as correlated attributes contribute towards an event. In future, we will be implementing our complete composite and global event detection and identification solution in a larger WSN framework consisting of our COTS sensor nodes for performing more thorough evaluations of our proposed schemes in terms of required memory resources and the overall network lifetime.

VII. CONCLUSION

This paper proposes a low complexity and computationally efficient *event identification* scheme for wireless sensor networks. We show, for the very first time, that *event identification* can be performed by using a low complexity implementation of dimensionality reduction technique. The proposed approach makes use of Generalized Hebbian Algorithm (GHA) to determine the relative contribution of each attribute in an *event* without compromising the accuracy of the results. The proposed identification algorithm has also been implemented and tested on off-the-shelf WizziMotes [7] proving the effectiveness of the proposed scheme by giving extremely accurate results in terms of the *identification* ratio of the individual attributes.

ACKNOWLEDGEMENT

This research work has been supported by LUMS Startup and FIF grant.

REFERENCES

[1] N. Shahid, I. Naqvi, and S. Qaisar, "Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey," *Artificial Intelligence Review*, pp. 1–36, 2012.

[2] I. Memon and T. Muntean, "Cluster-based energy-efficient composite event detection for wireless sensor networks," in *SENSORCOMM 2012, The Sixth International Conference on Sensor Technologies and Applications*, pp. 241–247, 2012.

[3] M. Bahrepour, N. Meratnia, and P. J. Havinga, "Use of ai techniques for residential fire detection in wireless sensor networks," 2009.

[4] N. Shahid and I. H. Naqvi, "Energy efficient outlier detection in wsns based on temporal and attribute correlations," in *International Conference on Emerging Technologies (ICET)*, pp. 1–6, IEEE, 2011.

[5] N. Shahid, I. H. Naqvi, and S. B. Qaiser, "Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey," *Artificial Intelligence Review*, 2012.

[6] M. Moshtaghi and et. al., "Incremental elliptical boundary estimation for anomaly detection in wireless sensor networks," in *IEEE International Conference on Data Mining (ICDM)*, p. 467476, IEEE, 2011.

[7] "Wizzi mote." Available: http://www.wizzilab.com/solutions/wizzikit/, 2014.

[8] M. Moshtaghi and et. al., "Clustering ellipses for anomaly detection,," *Pattern Recognition*, vol. 44, no. 1, pp. 55–69, 2011.

[9] M. Moshtaghi and et. al., "An efficient hyperellipsoidal clustering algorithm for resource-constrained environments," *Pattern Recognition*, 2011.

[10] N. Shahid, I. H. Naqvi, and S. Bin Qaisar, "Quarter-sphere svm: attribute and spatio-temporal correlations based outlier & event detection in wireless sensor networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2048–2053, 2012.

[11] N. Shahid, I. Naqvi, and S. Qaisar, "Real time energy efficient approach to outlier & event detection in wireless sensor networks," in *IEEE International Conference on Communication Systems (ICCS)*, pp. 162–166, 2012.

[12] N. Shahid and et. al., "Joint event detection & identification: A clustering based approach for wireless sensor networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, (Shanghai), 2013.

[13] S.-J. Lin, K.-H. Lin, and W.-J. Hwang, "FPGA implementation for GHA based texture classification,"

[14] E. Oja, "Simplified neuron model as a principal component analyzer," *Journal of mathematical biology*, vol. 15, no. 3, pp. 267–273, 1982.

[15] D. M. J. Tax and R. P. Duin, "Data description in subspaces," in *IEEE International Conference on Pattern Recognition*, 2000.

[16] M. Moshtaghi, C. Leckie, S. Karunasekera, J. C. Bezdek, S. Rajasegarar, and M. Palaniswami, "Incremental elliptical boundary estimation for anomaly detection in wireless sensor networks," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 467–476, IEEE, 2011.

[17] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, "Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 3, pp. 518–533, 2010.

[18] S. Suthaharan and et. al., "Labelled data collection for anomaly detection in wireless sensor networks," in *IEEE ISSNIP*, 2010.

[19] J. Gupchup, A. Terzis, R. Burns, and A. Szalay, "Model-based event detection in wireless sensor networks," *arXiv preprint arXiv:0901.3923*, 2009.

[20] B. Yu, T. Mak, X. Li, L. Smith, Y. Sun, C.-S. Poon, *et al.*, "Stream-based hebbian eigenfilter for real-time neuronal spike discrimination," *BioMedical Engineering OnLine*, vol. 11, no. 1, pp. 1–19, 2012.

[21] "Dash 7." Available: http://www.indigresso.com/wiki/, 2014.