Sādhanā Vol. 38, Part 2, April 2013, pp. 169–185. © Indian Academy of Sciences

Semantic intrusion detection with multisensor data fusion using complex event processing

R BHARGAVI* and V VAIDEHI

Department of Information Technology, Madras Institute of Technology,
Anna University, Chennai 600 044, India
e-mail: bhargaviren@gmail.com; vaidehivijay@gmail.com

MS received 8 March 2012; revised 7 September 2012; accepted 12 September 2012

Abstract. Complex Event Processing (CEP) is an emerging technology for processing and identifying patterns of interest from multiple streams of events in real/near real time. Sensor network-based security and surveillance is a topic of recent research where events generated from distributed sensors at an unpredictable rate need to be analysed for possible threats and respond in a timely manner. Traditional software architectures like client/server architecture where the interactions are pull-based (DBMS) do not target the efficient processing of streams of events in real time. CEP which is a push-based system can process streaming data to identify the intrusion patterns in near real time and respond to the threats. An Intrusion Detection System (IDS) based on single sensor may fail to give accurate identification of intrusion. Hence there is a need for multisensor based IDS. A multisensor-based IDS enables identification of the intrusion patterns semantically by correlating the events and context information provided by multiple sensors. JDL multisource data fusion model is a well-known research model first established by the Joint Directorate Laboratories. This paper proposes JDL fusion framework-based CEP for semantic intrusion detection. The events generated from heterogeneous sensors are collected, aggregated using logical and spatiotemporal relations to form complex events which model the intrusion patterns. The proposed system is implemented and the results show that the proposed system out performs the pull-based solutions in terms of detection accuracy and detection time.

Keywords. Wireless sensor networks; complex event processing; event stream; multi sensor data fusion; semantic intrusion detection.

*For correspondence

1. Introduction

Complex event processing is a method of tracking and analysing streams of information about things that happen, and deriving a conclusion from them. Complex event processing provides comprehensive solution for building applications to filter, correlate and process events in real-time so that downstream applications are driven by true, real-time intelligence (Luckham 2010). Complex event processing approach is ideally suited for applications delivering situational awareness and response where the system under observation is event driven, asynchronous push interactions are to be supported, and timely actions are required. Degree of complexity of a system is measured by any of the factors like degree to which the application is expected to change over time, number and types of event sources, number of listeners or consumers of the information, etc.

Sensor networks bridge the gap between the physical world and the virtual world of processing and communication (Holger Karl & Andreas Willing 2005). Wireless Sensor Networks (Holger Karl & Andreas Willing 2005) consists of spatially distributed autonomous sensor nodes that cooperatively monitor an environment by sensing the physical parameters like temperature, pressure, humidity, sound, motion, etc. and communicating among the neighbours. Wireless sensor networks are gaining importance in several industrial and civilian applications. Some of the applications of wireless sensor networks include security and surveillance (He *et al* 2006; Wang *et al* 2009) health care monitoring (Yao *et al* 2011), financial services (Adi *et al* 2006), habitat monitoring (Mainwaring *et al* 2002), fire detection, nuclear reactor monitoring and controlling, etc. Distributed sensors in the surveillance application produce huge data. Processing, analysing and detecting abnormal patterns from this data is very complex in nature. There are two approaches for processing and analysing the data streams generated from heterogeneous sensors. First one is *centralized approach* in which the sensors send the data or events to a central server which runs data processing and mining algorithms. Second one is *distributed approach* in which data is stored and processed in the nodes itself. Special processing and mining algorithms are proposed in the literature for distributed and in-network processing (Wang *et al* 2009). Complex event processing is based on the centralized approach.

The objective of this paper is to develop a system that collects data from numerous sensors about raw events (real world event), convert them into event instances, process them with CEP engine in an ongoing basis by applying aggregation rules to determine the interconnected trends and patterns and respond to the threats in a timely manner. A series of queries could look for a pattern of events that represents an opportunity, problem, or threat, and pass that alert to a downstream application or an administrative person. Our earlier work on complex event processing for object tracking and intrusion detection (Bhargavi & Vaidehi 2011) discusses CEP architecture and how rules can be developed for intrusion detection. However, the issues like integration of heterogeneous networks (wireless network and Ethernet) data, out of order arrival of events that come into picture while building the complete integrated solution are not addressed in our earlier work. In this paper, complete solution that can handle event streams generated from multiple sensors is presented and different scenarios like tailgating, server room entry violation, etc. are modelled using event expressions and validated in real time.

Contributions from this paper include development of a JDL fusion (Steinberg *et al* 1999) based CEP framework for multi sensor data fusion for semantic intrusion detection, where events generated from distributed heterogeneous sensors are processed at different levels to identify the abnormal patterns semantically, formulation of rules for complex events. The complex events model a pattern or a scenario and are formed by aggregating basic or primitive events using logical, temporal and spatial relationships. Contributions of the paper also include integration of

wireless and Ethernet technology for better understanding of the situation awareness, handling synchronization and out of order arrival of data and coming from different sources. The proposed system is implemented and validated in real time with number of intrusion scenarios. Results show that the proposed system out performs the pull-based solutions in terms of detection accuracy and detection time. In the following few paragraphs some of the related works are discussed.

Yao *et al* (2011) have proposed a CEP-based solution to process RFID streams for a healthcare application. They proposed and implemented a framework to model surgical events and critical situations in an RFID enabled hospital. A prototype system was developed to track patient flow and asset flow and abnormalities in real time. CEP rules are used for data filtering and aggregation and complex event detection. Drools CEP engine, open software is used for implementing the rules. Wang *et al* (2009) have proposed an event oriented framework process the RFID data by (i) declarative event specification with temporal constraints; (ii) declarative rules definition to support data transformation and real-time monitoring; and (iii) an RFID complex event detection engine. The event framework provides comprehensive support of RFID applications, including object tracking and real-time monitoring. However, the focus was only on RFID streams.

Pradeep & Khaparde (2010) have proposed and implemented complex event processing approach for processing high level power system events from the perspective of a multi-area system. The patterns in overdrawals and underdrawals, and irlink flows are studied for constituents of the national grid of India and modelled using CEP rules/patterns. ESPER tool (ESPER 2009) is used for implementing the rules. Wu *et al* (2006) have proposed a complex event language that allows development of queries to filter, aggregate, correlate events and transform them into new composite events. They have designed, implemented and evaluated their system called SASE that executes complex event queries over real time RFID streams. The developed system is compared against a relational stream processor, TelegraphCQ. But the proposed system was not applied and tested in a real time live environment.

Mori *et al* (2004) have proposed a scheme for tracking multiple people using pressure sensors and RFID system. An indoor environment with a bed of pressure sensors has been considered for the experiment. RFID system was used to identify the individual persons. To detect the multiple points where the people might exist, the system uses pressure sensors. Load on each tile is measured and then based on some threshold value, highly loaded areas are determined. Number of people inside a certain area is calculated using weight ratio of the areas on the basis of human count inside the room. Wasserkrug *et al* (2008) have proposed a mechanism for event materialization with a probability or uncertainty. They provided a model for representing materialized events and two algorithms for correctly specifying the probability space of an event history. The first algorithm on the construction of a Bayesian network provides an accurate solution, and the second is a Monte Carlo sampling algorithm that approximates the materialized event probabilities. But the proposed work was not tested in a real time environment. Also temporal logic was not incorporated in the proposed work which is an essential component of the CEP.

There are literatures on application of complex event processing to finance (Adi *et al* 2006), Business intelligence, habitat monitoring (Mainwaring *et al* 2002), healthcare (Baldus *et al* 2004), etc. Though CEP is suitable for sensor networks applications there is not much work done on CEP-based solutions to the intrusion detection and person tracking in a sensor network environment. This paper focuses on application of CEP to surveillance application specifically intrusion identification.

The remainder of this paper is organized as follows. Section 2 presents back ground information and some CEP preliminaries. Section 3 explains the proposed model. In section 4, real time implementation of the model is explained. Results are discussed in section 5. Finally, conclusions are provided in section 6.

2. CEP preliminaries

In this section, some basic definitions of event, event processing, event stream, etc. and background information on complex event processing are presented. CEP is an emerging technology for detecting known patterns of events and correlating them to complex events in real-time. An event is 'any thing that happens, or is contemplated as happening' (Luckham & Schulte 2008) in the real world and normally it is of interest to some group of people. A key stroke, a sensor outputs a reading, etc., are couple of examples of an event. Some times these events in turn may produce secondary events internally. Real world occurrences can be defined as events that happen over space and time. Events are of two types (i) basic/primitive events, (ii) complex events. Events have event attributes. An event attribute is a property of the event.

A basic event is atomic, indivisible and occurs at a point in time. Attributes of a basic or primitive event are the parameters of the activity that caused the event. Computer systems process the events by representing them as event objects.

A common model for an event is a tuple represented as

$E = E\,(id, a, t)$
here, id is the unique ID of an event,
$a = \{a_1, a_2, \ldots\ldots, a_m\}$, $m > 0$, is a set of attributes,
t is the time of occurrence of the Event.

For example, an RFID event is denoted as $E = e\,(o, r, t)$, where o is the tag EPC, r is the reader ID and t is the time stamp of the event.

Complex events are composed of basic events. Complex events are defined by connecting basic events using temporal, spatial or logical relations. A common model for a complex event is

$$E = E\,(id, a, c, t_b, t_e)\,,\ t_b >= t_e,$$

where $C = \{e_1, e_2, \ldots.., e_n\}$, $n > o$ is the vector that contains basic events and complex events that cause this event happen; t_b, t_e are starting and ending times of the complex event.

Attributes of complex events are derived from the attributes of the constituent primitive events. Events constructors and event operators are used to express the relationship among events and correlate events to form complex events.

Any basic event or a complex event is specified by an event expression. An event expression is a mapping from histories (domain) to histories (range) (Gehani *et al* 1992).

E: histories $\rightarrow$ histories.

Event expression is formed by combining events with the event constructors. Since event expressions are equivalent to regular expressions it is possible to implement event expressions using finite automata. Table 1 shows the logical event constructors and table 2 shows the temporal event constructors (Yao *et al* 2011). Any complex events can be modelled with the event expressions using these constructors.

An event stream or an event cloud is an infinite sequence of linearly ordered events i.e., {data-tuple, time stamp}. Event processing is a technology to perform operations on specific events like reading, deleting, abstracting, etc. CEP is a technique for analysing streams of event data in real-time, improving situational awareness and enabling immediate response to emerging opportunities and threats. CEP looks at events in the context of other events rather than in isolation.

Table 1. Logical constructors.

AND ($\wedge$)	E**1** $\wedge$ E**2**	Conjunction of two events E**1** and E**2** without occurrence order
OR ($\vee$)	E**1** $\vee$ E**2**	Disjunction of two events E**1** and E**2** without occurrence order
NOT ($\neg$)	$\neg$ E**1**	Negation of E**1**
Sequence	E**1** : E**2**	E**1** occurs followed by E**2**

Table 2. Temporal constructors.

window()	window(E1,t)	E1 occurs for a time period t
	window(E1,n)	E1 occurs n times
within()	within(E1,t)	E1 occurs within less than t
	within(E1,t1,t2)	E1 occurs within interval t1 and t2
at()	at(E1,t)	E1 occurs at t
Every(*)	E1*	Every occurrence of E1
during()	during(E1,E2)	Event E2 occurs during event E1

Event processing agents provide the events to the event processing engine in the suitable form. Event processing engine then processes the events by applying the predefined rules/patterns and then notifies. A rule is a Boolean combination of user defined Boolean functions and Event Query Language (EQL) queries. Interesting event patterns need to be subscribed using the EQL rules/patterns. Once the queries are subscribed, the engine continuously monitors the incoming event stream for the occurrence the event pattern and indicates the rule/pattern hit and an automated response takes place in a way similar to Event Condition Action (ECA) paradigm. This automatic response generation is handled by developing listeners corresponding to every rule/pattern subscribed. Once the events of interest are identified by the event processing engine the listener takes the necessary action to be followed.

3. Proposed semantic intrusion detection system based on multisensor data fusion

3.1 *System architecture*

Figure 1 shows the architecture of the proposed complex event processing system for semantic intrusion detection system.

The proposed CEP system has multiple event receivers. Each event receiver receives the data/events coming from a different data/event source. The event receiver on receiving the data from the source converts them into event streams. Data generated by different sources follows a different format hence event receivers also convert the data from different sources to the specific format suitable for processing further by the event processing engine. The events generated by the event receivers are inserted in to a FIFO or a queue. Events are organized in the queue in the order of their detection time.

To avoid the out of order arrival of the events which is caused by the network delays, the events are stored internally in the queue for some tome T. Hence the events are dequeued after a time T for processing. Any event getting generated at time t will be processed after t+T time by the CEP engine. Any event arriving the queue with a time stamp smaller than the time stamp of the already dequeued events is ignored. This leads to missing of events. Missing events due to

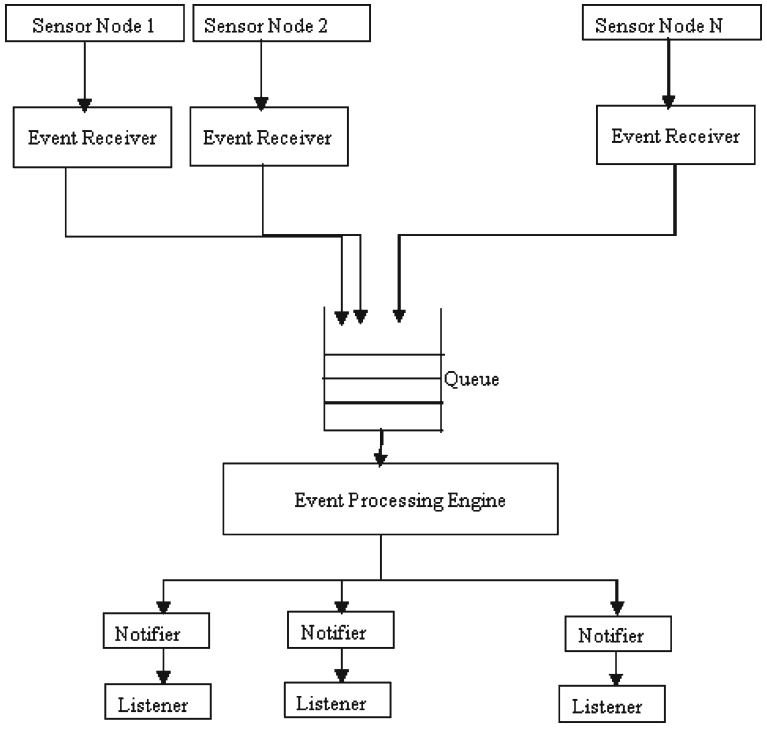

Figure 1. CEP architecture for semantic intrusion detection system.

network delays can be avoided by having larger value of T i.e., by storing the events internally for longer duration. But this will delay the event processing time. Hence there is a trade off between missing events due to network delay and latency.

Event processing engine processes the event streams. All the rules and patterns are to be registered initially. The listeners are intimated whenever the corresponding rule is hit or pattern is matched for which it is configured. Modelling of the complex events using event expressions is explained later in this section. Notifiers are used for intimating the listeners about the rule or pattern occurrences. Listeners are the modules that take necessary action on notifications.

There are several supporting modules in semantic IDS to perform the activities on the occurrence of certain events. These modules are Kalman tracking module, person detection module, authentication module, etc. Person tracking is done using Kalman filter. All these modules are listeners to the CEP engine. Kalman filter takes the position coordinates of a person and time at which the person is detected at place as the input and predicts the next position of the person. Thus, the path followed by a person can be tracked, also we can determine if the person is moving in a place where he is not authenticated to. For person recognition and tailgating image processing modules are used.

3.2 *CEP reference model*

JDL data fusion model is adapted for complex event processing (Bass 2006). Figure 2 shows the CEP reference architecture. The objectives of data fusion in the JDL model extend beyond

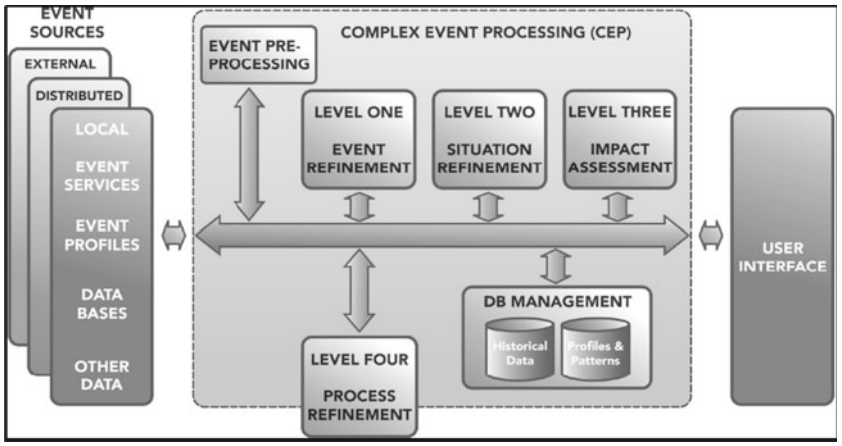

Figure 2. JDL Model for CEP.

merely merging data; the purpose is to achieve dynamic understanding and awareness of situation – including threat impact – and enable adaptation of plans and processes to compensate for such situational threat. To achieve the goals using this reference model events are processed at five different levels of abstraction starting from event preprocessing (Level 0) through Event refinement (Level I), situation refinement (Level II), impact/threat assessment (Level III), and finally Process refinement (Level IV).

Figure 3 shows how the primitive events generated from different sensors can be aggregated to generate a complex event which represents a pattern or a scenario of interest. Sensed information from the sensors that interact with the physical environment/world is collected by the event collection subsystem. Various scenarios representing simple and complex events have been modelled using event expressions.

Following primitive events are considered for the present study.

- Events generated during the interaction between the RFID readers and tags.
- PIR readings generated whenever a person crosses the sensor.
- GPS readings indicating the location of event occurrence.
- Time of event occurrence.
- Images captured by the camera.

Events generated from heterogeneous sources are processed as per the reference model as described below.

3.2a *Event preprocessing (Level 0)*: Event preprocessing is the first step in the CEP. This step deals with the raw data collection from physical sources like sensors and preparing the data for further processing. In the context of Intrusion Detection System data is generated from heterogeneous sensors like PIR sensor, camera, RFID, etc. in the coverage environment. These sensors monitor the environment for any change in the physical phenomenon and capture the changes. The observed data along with the time of event occurrence and location information is captured as an event and communicated to the central server for further processing.

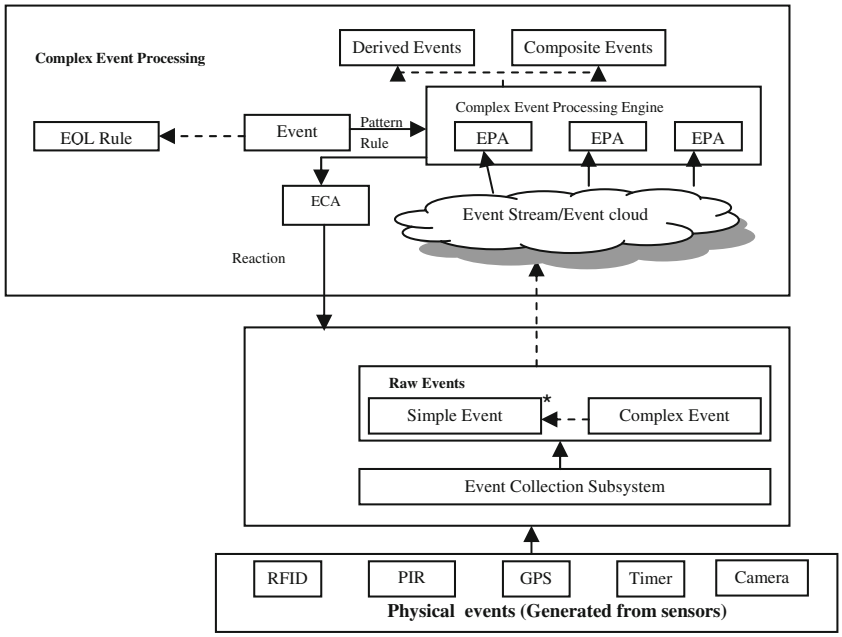

Figure 3. Complex event processing for semantic IDS.

3.2b *Event refinement (Level 1)*: Raw events sent from different sources or generators are in different formats and structures. Some times there may be missing data also. All these inconsistencies are handled in the event refinement level. Basic feature extraction from the raw data handled at this stage. Basic events like RFID event, PIR event, etc. are identified which will be further processed by the upstream processing modules. This is the essential step for better performance of the upstream modules in identifying the patterns that infer intrusion.

3.2c *Situation refinement (Level 2)*: In this stage, selected events of interest are identified from the basic events which will be further aggregated and tracked in the next level. In Semantic IDS this means identification of invalid RFID event, missing RFID event, invalid image event, etc. This involves steps like identification of ranges of values for each event to classify as abnormal event, evaluation of the current event to classify as normal event or abnormal event/event of interest and mark them for further tracking.

3.2d *Impact assessment/threat assessment (Level 3)*: Threat assessment or impact assessment is the next higher level of inference. Here the events identified by level 1 and level 2 are aggregated using logical and spatiotemporal relations to form complex events which model or represent intrusion. For example, in the case of Intrusion Detection System Tailgating event is said to occur if there is a missing RFID event followed by multiple people identified event. Similarly, if an event representing the person entry and another event representing person exit do not happen in duration of 10 minutes time, it is identified as an intrusion.

3.2e *Process refinement (Level 4)*: Process refinement step is used for tuning or adjusting the system. Unlike in levels 0 to 3, in level 4 no event identification or prediction happens. Instead, here the decision variables or models are updated or tuned for better performance.

Few of the complex event scenarios and their modelling using event expressions are discussed below.

Example 1: Unauthorized entry
When a person who does not possess a valid RFID tag tries to enter a location just behind an authorized person it is called as tailgating. This scenario can be captured as explained below.

CEP queries the wireless data for an event where PIR is present but RFID is zero. Once such an event is identified, CEP gets the image from the database corresponding to that event's timestamp and gives the image to Haar Face Detection module. The Haar Face detection algorithm counts the number of persons present and gives the count back to CEP. This scenario can be modelled as follows:

E1: Office entry
E1 = (s1, o1, t1) type(s1) = RFID event
E2: Office entry
E2 = (s2, o1, t2) type(s2) = PIR event
E3: Multiple people sensed by Image sensor.

The complex pattern can now be formulated as

$$\textbf{every}\left((\textbf{E1};\neg(\textbf{E2}))\ \hat{}\ \textbf{E3}\right).$$

Now the rule can be developed to rise an alert on the above complex event using IF – THEN condition

IF (True)
Get the Image with the same Time stamp and Node ID from the Data Base.
Send the location of the Image in the Data Base to the Image processing module for checking number of people.
Receive the response from the Image processing module.
Generate alert if needed.
End

Example 2: Anomaly detection involving temporal relation
If an organization has a policy or rule as: Any person entering the server room should exit in 10 minutes and if any violations to this rule occur it can be captured as follows:

E1: Server room entry
E1 = (s1, o1, t1) type(s1) = Entry.
E2: Server room exit
E2 = (s2, o1, t2) type(s2) = Exit.

The complex pattern can now be formulated as

$$\textbf{within}\,(\textbf{E1};\neg(\textbf{E2}),\ \textbf{10 min}).$$

Now the rule can be developed to rise an alert on the above complex event using IF – THEN condition

IF (True)
  Generate Alert
End

Example 3: Anomaly detection involving temporal and logical relation
If an organization has a rule as; A junior engineer is allowed to enter the server room only when he is accompanied with one system administrator

E1: Junior engineer server room entry
  E1 = (s1, o1, t1) type(o1) = Junior Engineer
E2: Administrator server room entry
  E2 = (s1, o2, t2) type(o2) = Administrator.

The complex pattern can now be formulated as

$$\textbf{within}\left(\textbf{E1} \wedge \neg\,(\textbf{E2}),\ \textbf{5 sec}\right).$$

Now the rule can be developed to rise an alert on the above complex event using IF – THEN condition

IF (True)
  Generate Alert
End

4. System implementation and validation

This section explains the implementation of the proposed semantic intrusion detection system and validation of the system in a simulated environment as well as in a real time environment.

4.1 *Design and implementation*

The complete semantic intrusion detection system is developed using JAVA. JAVA based rule engine ESPER is used for modelling the complex events. The system has two modes of operations namely Simulation mode, and Real time mode. In the Simulation mode the system supports and handles the data generated by a simulator where sensor network wireless and the person's movements are simulated using a Java application. The Real time mode is used when the system is run with the physical sensors. The following sections brief the design and implantation details.

The proposed system has four important modules namely Wireless event receiver, wired event receiver, Event processing module and User interface for querying the data base.

All the Event receivers run in parallel in different threads. Wireless event receiver handles the events generated by the wireless nodes. It reads the data packets received at the sink. The received data packet is parsed to get Sensor ID, PIR, RFID, Sensor Location (LLA), Timestamp. This information is then written in to a queue for further processing. This is done to avoid the loss of data packets.

Wired event receiver is functionally similar to the wireless event receiver. It is used for receiving the events generated by cameras.

The events received by the receivers are then put in the queue for a specified time T. The need for this queue is explained in the system architecture.

Event processing module reads dequeued data from the queue, creates an object instance of the event and sends to the CEP engine. The CEP engine constantly monitors the streaming events for the registered queries and generates the responses to the corresponding listeners.

The User Interface module is used for querying the data base. The functions of this module are to get the Inputs from the user, make a request to the DB and get the results, and finally display the results.

At the start up of the system, all the configurations, initializations are done and all the threads are started. All the supporting modules for tracking and image processing, etc., are interfaced with the event processing engine as listeners. Few Listeners developed are as follows.

countObjectsListener: Counts the number of people present in a location and alerts if the number crosses a threshold.
logMsgsListener: Invokes Kalman Tracking Module for tracking and predicting the future position of a person.
pathListener: Indicates the places being visited by a person.
priorityListener: Indicates the unauthorized entry into privileged areas.
rfidValidityListener: Checks the RFID validity.
sensitiveZoneListener: Invokes person Recognition Module.
sensorValidityListener: Counts the number of faces in tail gating problem.

The developed proposed Semantic Intrusion Detection System validated with simulation environment as well in real time environment with a WSN of RFID, PID sensors, GPS receiver, Timers and a wired network of cameras. The software system for processing the events generated by the sensors is developed using JAVA.

4.2 *Simulation WSN environment*

A Wireless sensor network consisting of seventy sensor nodes and movements of ten people is simulated for generating the events for validating the proposed semantic intrusion detection system. The sensor node deployment and person movements in WSN are done graphically using Applets in java. The main advantage of the simulation is that the number of nodes and persons are not restricted. The deployment can be configured to be either regular or random. Figure 4 shows a snap shot of the deployment scenario. Here the sensor nodes are represented with circles. The green coloured circles represent the inactive sensors and the other coloured circles represent the activated sensors which have been activated by the movements of the people. Path of a person as he/she moves is indicated by straight line path, ramp path or zigzag path, etc. An event is generated by these sensors whenever the sensors detect the presence and movement of a person. PIR is used for detecting the presence of a person. RFID is used for identification of the person. Sensor location is treated as the location of the person. The location information is given by the GPS receiver. Data/events generated by the sensors are formed as wireless packets and send to the server using socket programming. WSN deployment details, details of the authenticated people, and some other required knowledge base, are maintained in the data base server.

4.3 *Real time environment*

Sensor node used in this work consists of two types of sensors namely PIR and RFID reader. The sensor node also has a GPS receiver and timer. Figure 5 shows the wireless sensor node used

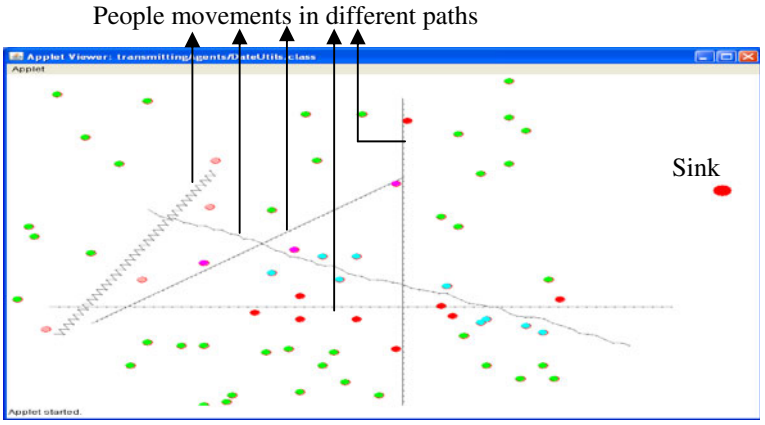

Figure 4. Simulation of WSN deployment and person movements.

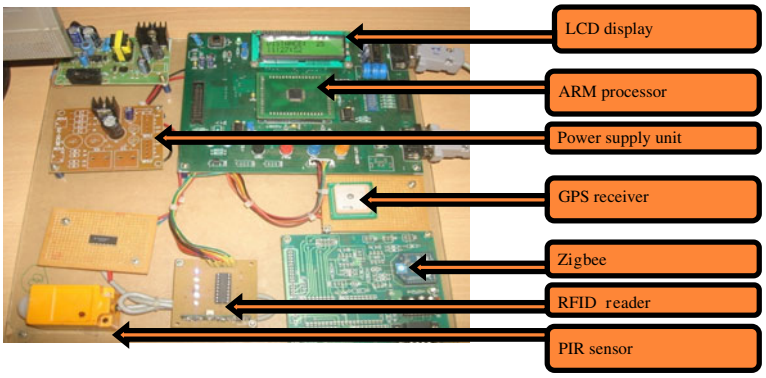

Figure 5. Wireless sensor node.

Figure 6. Interface between sink and server.

in the real rime mode of operation of the system. Figure 6 shows the interface between the sink node or base station and the server where the processing of the events takes place. Sink node is connected to the server using RS232 interface. 10 nodes are used for validation of the proposed system.

5. Results and discussion

The proposed semantic IDS has been validated with several test cases. To validate the tailgating scenario the image captured at the entrance of a room is shown in figure 7. The anomalies observed by the CEP are as follows. Number of events detected by PIR is 3, number of RFID readings is 1. Hence CEP gets data needed information like the location or sensor ID and time at which the anomaly has been detected. With this information the image captured from the corresponding camera is fetched and the number of people in the image is counted. If the count is greater than 1 then it is treated as tailgating occurrence.

At highly secured zones just checking the authorization is not sufficient. There is a possibility for an intruder to wear a valid RFID tag and enter into the secured zone. To identify such intrusion, the captured images are used. When the person enters the zone his RFID, sensor ID and time are captured in the wireless data stream. At the same time PIR sensor enables the camera and the person's photo is also captured with the same time stamp and sensor ID. This image is compared with the available images in the database with the same RFID. If there is a mismatch between these images then it is treated as an intrusion. Figure 8 shows a screen shot of this scenario.

Figure 9 shows the results of Kalman tracking module when a person deviates from the permitted paths for him/her. There are predefined paths between any source and destination. When a visitor enters the campus which is under surveillance the details are collected at the entry point. From these details the predefined path of the visitor is obtained. As the person moves ahead towards the destination the sensors located at different places capture the presence of the visitor and this information is sent to the sink where it will be processed as explained in the previous

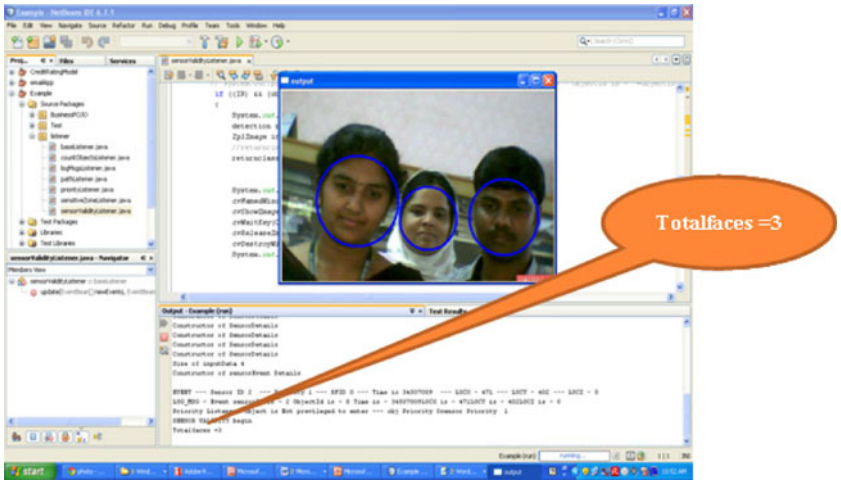

Figure 7. Tailgating snap shot.

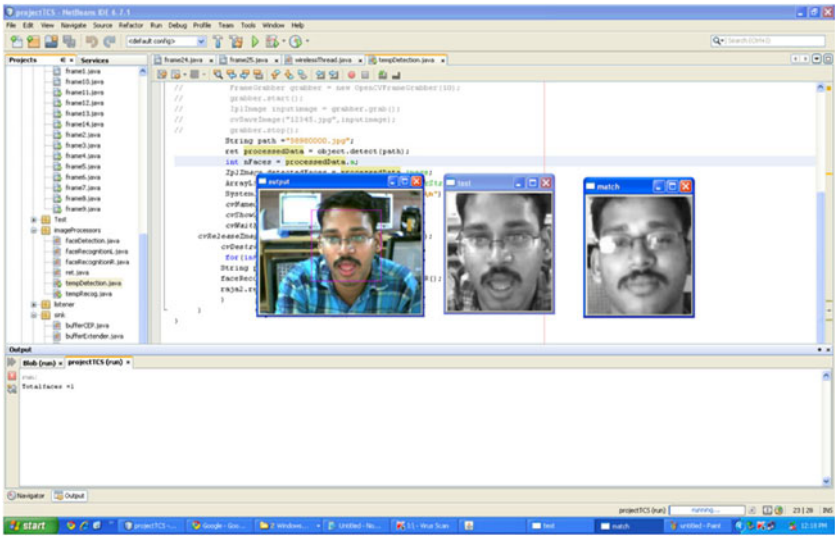

Figure 8. Screen shot of restricted zone authentication.

sections. Kalman tracking gives the predicted position of the person from the previous position inputs. From this information and the predefined paths it can be determined if the person is moving in the authorized place or not.

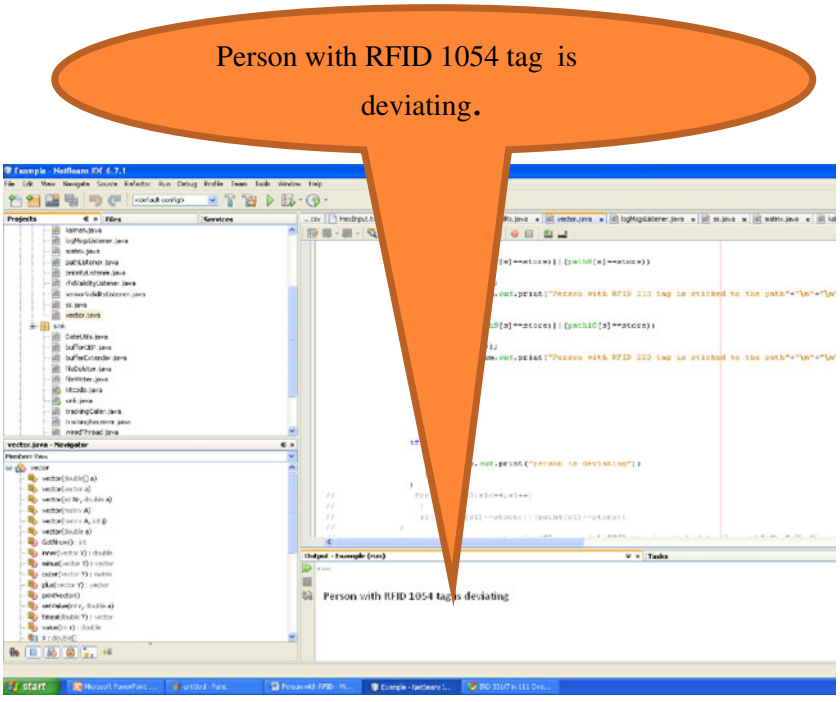

Figure 9. Snap shot of path verification.

The performance of the proposed system is compared with the traditional pull based approach. It is observed that the proposed system performs better in terms of detection time and detection accuracy. In the pull based approach, the queries need to be fired constantly at some time intervals. If the time between successive querying is more then the latency i.e., time between the event occurrence and event identification is more. If the time between successive querying is less then the latency will reduce but it increases the processing and other overheads like interactions with database, etc. With the proposed system the event of interest can be captured as and when it occurs. Figure 10 shows the comparison of pull based system or a traditional DBMS system and the proposed push based system or event driven CEP system in terms of query or rule execution time.

Figure 11 shows the variation of computation time for executing the rules with the number of rules. It is observed that computation time does not vary much as the number of rules is varied.

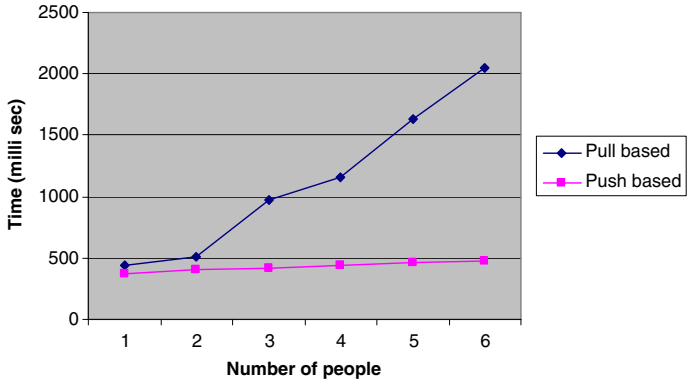

Figure 10. Comparison of pull based and proposed push based system.

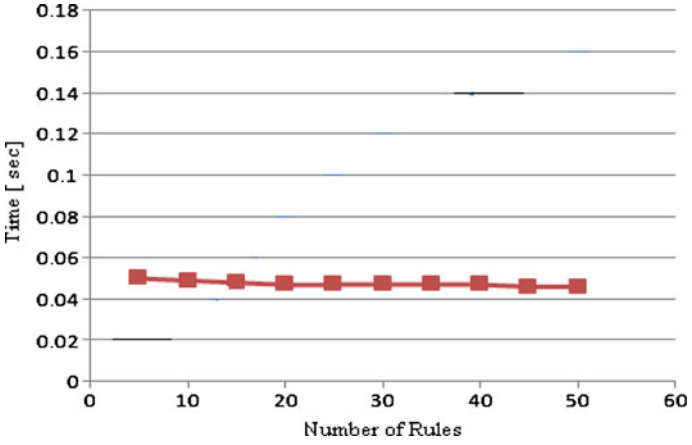

Figure 11. Response time vs number of rules.

6. Conclusion

JDL based Complex Event Processing approach for semantic intrusion detection in surveillance application has been proposed in this paper. Complex Event Processing enables sense-and-respond behaviour, in which incoming events or information is used to assess the current situation and generates a response in a timely fashion. Early identification of significant complex events provides situational awareness and better decision making. The events generated from heterogeneous sensors are collected, and processed at different levels of abstraction by aggregating as per the JDL fusion framework. Patterns representing intrusion are modelled as complex events which in turn are aggregated from base events and other complex events using logical and spatiotemporal relations. This paper also addresses how to handle issues like integration of heterogeneous networks data and out of order arrival of events. The proposed CEP based semantic intrusion detection system is capable of identifying the predefined intrusive patterns from the event streams generated from multiple sensors. The proposed semantic intrusion detection system is implemented as a multi threaded JAVA application. ESPER tool is used as the event processing engine. Rules have been developed to identify the unauthorized entry of the people into the security zones, tailgating issue and few more scenarios. Using the power of CEP, only suspicious people can be identified and tracked. The developed system integrates benefits of CEP for pattern identification and prediction capability of Kalman Filter for tracking suspicious people. The proposed system is implemented and validated in a Real time sensor network environment as well as in a simulated environment with several test scenarios. The performance of the proposed system is compared with the traditional pull based architecture and it is observed that the proposed system out performs the pull based solutions in terms of detection accuracy and detection time.

Acknowledgements

This research project was supported by Tata Consultancy Services (TCS), Innovation Labs, Bangalore. The authors would like to thank for the support.

References

Adi A, Botzer D, Nechushtai G and Sharon G 2006 Complex event processing for financial services. In: *Proceedings of the IEEE Services Computing Workshops (SCW'06)*, Chicago, Illinois, September 18–22, pp. 7–12

Baldus H, Klabunde K and Muesch G 2004 Reliable set-up of medical body-sensor networks. *Wireless Sensor Networks.* In: *Wireless Sensor Networks Journal* 2920/2004, 353–363

Bass T 2006 Fraud detection and event processing for predictive business. Technical report, Tibco

Bhargavi R and Vaidehi V 2011 Complex event processing for object tracking and intrusion detection in wireless sensor networks. *Int. J. Comput. Theory and Eng.* 3(3): 434–438

ESPER TOOL 2009 URL: http://esper.codehaus.org/esper/documentation/documentation.html

Gehani N H, Jagadish H V and Shmueli O 1992 Composite event specification in active databases: Model and implementation. In: *VLDB '92: Proceedings of the 18th International Conference on Very Large Data Bases*, 327–338. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc

He Tian, Sudha Krishnamurthy, Liqian Luo, Ting Yan, Lin Gu, Radu Stoleru, Gang Zhou, Qing Cao, Pascal Vicaire, John A Stankovic, Tarek F Abdelzaher, Jonathan Hui and Bruce Krogh 2006 Vigil Net: An integrated sensor network system for energy-efficient surveillance. *ACM Transactions on Sensor Networks* 2(1): 1–38

Holger Karl and Andreas Willing 2005 Protocols and Architectures for Wireless Sensor Networks, Wiley Publications

Luckham D C 2010 *The power of events: An introduction to complex event processing in distributed enterprise systems*. Boston, MA, USA: Addison Wesley Longman Publishing Co., Inc

Luckham D and Schulte R 2008 *Event processing glossary* – Version 1.1. Event Processing Technical Society. URL: http://www.ep-ts.com/component/option,com_docman/task,doc_download/gid,66/Itemid,84/

Mainwaring A, Polastre J, Szewczyk R, Culler D and Anderson J 2002 *Wireless sensor networks for habitat monitoring*. In WSNA, Atlanta, USA, 88–97

Mori T, Suemasu Y, NoguchiI H and Sato T 2004 Multiple people tracking by integrating distributed floor pressure sensors and RFID system. In: *IEEE International Conference on Systems, Man and Cybernetics*, 5271–5278

Poorani M, Vaidehi V, Rajesh M, Bharghavi, Balamuralidhar and Grish Chandra 2010 Semantic intruder detection system in WSN. In: *Proceedings of International Conference on Advanced Computing (ICoAC)*, MIT, Anna University, 26–32

Pradeep Y and Khaparde S A 2010 Complex event processing of high level events in multi-area power grid: An Indian perspective. In: *Proceedings of Power and Energy Society General Meeting*, 1–6

Steinberg A N, Bowman C L and White F E 1999 Revisions to the JDL data fusion model. In: *Proceedings of the SPIE. Sensor Fusion: Architectures, Algorithms and Applications*, SPIE, 430–441

Wang F, Liu S, Liu P and Bai Y 2006 Bridging physical and virtualworlds: Complex event processing for rfid data streams. In: *10th International Conference on Extending Database Technology (EDBT'2006)*

Wang F, Liu S and Liu P 2009 Complex RFID event processing. *VLDB Journal* 18: 913–931

Wang Xue, Sheng Wang and Daowei Bi 2009 Distributed visual-target-surveillance system in wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics* 39(5): 1134–1146

Wasserkrug S, Gal A, Etzion O and Turchin Y 2008 Complex event processing over uncertain data. In: *DEBS '08: Proceedings of the Second International Conference on Distributed Event-based Systems*, New York, NY, USA, 253–264

Wu E, Diao Y and Rizvi S 2006 High-performance complex event processing over streams. In: *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, 407–418

Yao W, Chao-Hsien Chu and Li Z 2011 Leveraging complex event processing for smart hospitals using RFID. *J. Network Comput. Appl.* 34(3): 799–810