An Online Shadowed Clustering Algorithm Applied to Risk Visualization in Territorial Security

Rafael Falcon and Amiya Nayak
Electrical Engineering and Computer Science
University of Ottawa
800 King Edward Ave., Ottawa ON, Canada K1N 6N5
{rfalc032, anayak}@eecs.uottawa.ca

Rami Abielmona
Research & Engineering
Larus Technologies Corporation
58 Antares Drive, Unit 1, Ottawa ON, Canada K2E 7W6
rami.abielmona@larus.com

Abstract—The identification and processing of the risk sources that prevail in a sensor-monitored area is crucial to guarantee the uninterrupted and efficient operation of the surveillance system. In particular, a time-varying schematic depiction of the risk associated with each object will allow the human expert to draw meaningful conclusions about the system dynamics. In this paper, we introduce an online clustering algorithm for risk visualization in a Territorial Security environment. The clustering machinery leans upon shadowed sets due to their robustness and interpretability. The proposed algorithm is able to process data arriving in real time as it only memorizes a small subset of them. It is strong to noisy and abnormal samples and represents each cluster as a shadowed set. Experiments conducted in a simulated Critical Infrastructure Protection scenario confirm the feasibility and robustness of the proposed technique.

I. INTRODUCTION

Preventing, detecting, and responding to unauthorized persons and/or goods crossing a perimeter is a security concern of individual, corporate, and international scope. State-of-the-art perimeter security solutions use physical barriers, sensors (e.g., indoor motion, cameras, audio/vibration), and human personnel (e.g., a camera operator, entrance guard, and patrolling security guards). These procedures are effective in a limited scenario, where a few entrance points are constrained by well-delineated physical boundaries.

Territorial security, however, deals with large regions of strategic importance, such as international borders, transportation (e.g., airports, rail yards, public transit), and critical infrastructure (e.g., military bases, nuclear facilities, emergency services, etc.) [17]. When trying to monitor such large geographically-distributed infrastructure, a number of challenges present themselves. First, a linked network of this size is inflexible and expensive to setup. Second, security operators frequently suffer from overload, stress, and inattention due to the substantial influx of data. Finally, it is increasingly important for territorial security systems to allow for the sharing of knowledge to specified entities [10].

Current perimeter security systems demonstrate the effectiveness of fixed, wired sensor networks. A network of sensor nodes positioned along a perimeter collects disparate and diverse types of relevant data. As examples of perimeter security sensor suites: color and infrared cameras enable the detection and recognition of intruders; acoustic sensors sense activities near a barrier, such as digging; while sonar and radar provide the location of marine or air-based objects. The identification of the risk sources that prevail in a given deployment area and their watchful monitoring is thus crucial to guarantee the uninterrupted and efficient operation of the perimeter security solution.

As previously identified by the authors [3], a proactive approach to managing potential faulty units is required for a stable perimeter security solution. To that effect, they proposed a multi-modular architecture that evolves the local risk perception time over time. Though its overall design is largely automated, it enables a human expert who occasionally monitors the system to override the current formulation of risk, either globally or for a particular object, through the interaction with the risk visualization and risk assessment modules.

The risk visualization module displays clusters of objects in the multidimensional space defined by the risk features under consideration (e.g., battery level, distance to an intruder). The underlying clustering algorithm, termed “Evolving Shadowed Clustering” (ESC), is built on the formalism of shadowed sets [14], an appealing generalization of fuzzy sets. Periodic in-field reports from the sensor network arrive at a centralized location (e.g., a base station), where the collected stream is split into multiple data snapshots for further analysis and processing. A data snapshot contains the values of the risk features reported by every object in the system at a particular time. Hence, ESC operates on a single snapshot at a time and attempts to reveal the continuity pattern between the previous and the current snapshots. It also raises timely alerts whenever the structural properties of the corresponding cluster distributions have undergone a significant change.

In spite of the encouraging results accomplished with this snapshot-based partition of the data stream, two subtle drawbacks that obscure the accurate modeling of the real-time perimeter security solution behavior can be pinpointed. The first one is related to time synchronization issues, as sensors are scheduled to transmit their measurements to the base station at particular discrete intervals (in order to minimize processing latency) and this triggers the need for an expensive and periodic network-wide agreement on the common time slots. The second disadvantage is the underlying assumption that all sensing nodes have the same transmission rate. If a sen-
sor suspects that an event is taking place in its surroundings, it will try to sample the environment more often and submit its reports to the base station, but its previously received reading will be replaced with the one arriving at the end of the current time slot, for only one vector of risk feature values can be encapsulated for each sensor within the same snapshot.

The alternative to a snapshot-based risk visualization at the base station is a sample-based one. That is, each sensing unit is regarded as a standalone source of information and thus processed independently and asynchronously. In that way, there is no need for coordinated data transmission from all the sensors and each can transmit at its own rate. This sample-based data stream partition also provides a different perspective of the system behavior, as its local and global risk perceptions are affected with every incoming data sample. Yet the algorithmic machinery behind the existing clustering method (ESC) cannot operate on a sample-by-sample basis. This paper introduces an incremental clustering approach, termed Online Shadowed Clustering (OSC), that will be used in lieu of ESC for risk visualization in a sample-based scenario.

The rest of the paper is structured as follows. Sect. II is related work. A general description of the risk-driven framework is offered in Sect. III. Shadowed sets are briefly introduced in Sect. IV before the shadowed clustering algorithms, ESC and OSC, are dissected in Sect. V. Experiments are unfolded in Sect. VI and conclusions stated in Sect. VII.

II. RELATED WORK

A. Risk Management Frameworks

Authors in [18] and [5] adopt continuous Hidden Markov Models (HMM) to evaluate system risk based on predefined qualitative categories (e.g. assets, facets) and a weighted sum approach. However, they rely to a large extent on a priori knowledge about the system (e.g., event occurrence probability, impact of the observations, initial state distribution, state transition rate matrix, etc.). Furthermore, a HMM is required for each asset’s facet, which makes the system’s representation entangled and sensitive to slight departures from the envisioned initial models. The works in [1] and [16] are concerned with risk modeling in critical infrastructures. The former aims at modeling the security properties of interdependent systems and measures their risk levels and assurances. The latter envisions a service decomposition graph for risk assessment and an online monitoring tool of three risk parameters. The initial snapshot is made up of the cluster centroids, which are conveyed as the initial centroids of the next snapshot into weighted examples, represented by the cluster centroids, which are conveyed as the initial centroids of the next snapshot of data. The speedup that is gained from this knowledge propagation is negated by the occurrence of shifts and drifts within the data streams. In [9], the clustering methodology requires a parameter which constrains the number of clusters being partitioned and arbitrarily chooses initial cluster centers. Both of these are seen as drawbacks to truly online clustering since the desired interval for the number of clusters cannot be usually determined a priori and the algorithm becomes sensitive to that initial cluster partition.

Authors in [7] and [6] presented online variants of FCM in which data was processed (still in chunks) as it arrived. In [7], a single-pass FCM (SPFCM) method was introduced in which large datasets were separated into smaller partial ones. The initial snapshot is made up of the cluster centroids, which are subsequently condensed into weighted points. The latter are clustered along with the new points arriving in the next snapshot. In [6], a streaming FCM (SFCM) method, whose operation is quite similar to that of SPFCM, was proposed. The idea of varying the amount of history used by the streaming algorithm is introduced. Although both SPFCM and SFCM performed nearly as well as FCM, they have their limitations: SPFCM requires the reordering of the incoming data which degrades its performance and SFCM is quite sensitive to the user-set parameter representing the number of history chunks to utilize for incoming snapshots.

III. RISK MANAGEMENT FRAMEWORK FOR TERRITORIAL SECURITY

Fig. 1 displays the architectural view of the risk management framework introduced in [3]. Yet in this version, each incoming data sample will be processed as it arrives, i.e. the data stream will not be split into multiple discrete snapshots.

Risk feature extraction provides an initial characterization of the set of risk features \( F = \{ F_1, F_2, \ldots, F_n \} \) that the system will monitor (e.g. coverage radius, intrusion distance), leaning on either expert knowledge or more sophisticated feature mining techniques from data streams.

Risk visualization allows the user to watch clusters corresponding to the sensor nodes in the risk feature space. By doing so, the human expert is able to alter the level of the perceived threat for a particular sensor or group of sensors. This knowledge is immediately incorporated into the system.

Cluster visualization is a general-enough representation of
the system’s prevalent dynamics and thus can be applied to any risk source (e.g. cyber-attack, measurement error, etc.). Shadowed sets are a natural extension to fuzzy sets and we used them to represent online clusters given their robustness against noise and outliers as well as their fast convergence and high interpretability [11].

**Risk assessment** quantifies the overall risk posed by any system unit across all features on the basis of fuzzy and shadowed evaluations of their local risk. It is sensitive to adaptive user feedback in modeling the local risk perceptions and adjusts its behavior upon the failure of any system unit.

**Risk monitoring** creates and maintains a contingency plan to lower the overall system risk and triggers alarms whenever a threatening condition arises.

**Risk forecasting** updates its built-in model for risk prediction on the basis of the previous observations and the system’s current state and behavior.

**Security manager** coordinates actions that must be taken upon the environment to mitigate the danger, e.g. replacing a damaged sensor or sprinkling water over a blazing area.

This paper discusses in detail the risk visualization module. A necessary introduction to the shadowed set methodology, upon which this module hinges, is outlined in the next section.

### IV. Shadowed Sets

Shadowed sets are algorithmically induced by fuzzy sets [14]. They tackle the uncertainty arising from the assignment of quantitative membership degrees \( \mu_X(x) \in [0, 1] \) to any element \( x \) of the concept \( X \). A shadowed set performs two cuts over the membership grade space of a fuzzy set \( X \), thus splitting it into three disjoint regions: the core \( \text{CORE}(X) \), the exclusion \( \text{XCL}(X) \) and the shadowed \( \text{SDW}(X) \). Each element \( x \in X \) belongs to exactly one region, i.e. \( \text{CORE}(X) = \{ x | \mu_X(x) \geq 1-\lambda \} \), \( \text{XCL}(X) = \{ x | \mu_X(x) \leq \lambda \} \) and \( \text{SDW}(X) = \{ x | \lambda < \mu_X(x) < 1-\lambda \} \), where \( \lambda \in (0,0.5) \) is a threshold value. The creation of a shadowed set is depicted in Fig. 2.

The membership degrees of the core elements are elevated to 1, those of the exclusion members are lowered to 0 and those of the shadowed elements remain as in the original fuzzy set \( X \). These three zones are functionally equivalent to the positive, boundary and negative regions in rough sets [13].

The optimal value of \( \lambda \) is sought after the principle of uncertainty relocation, i.e. the reduction and elevation of membership in the exclusion and core zones must be compensated with the uncertainty remaining in the shadowed region.

Shadowed sets are more interpretable granules than fuzzy sets. They bear the qualitative strength of rough sets with the added value of numerical discernibility in the shadowed regions provided by fuzzy sets. For these reasons, they are adopted as the representational mechanism for clusters of sensor nodes across the risk feature space in Sect. V.

### V. Shadowed Clustering Algorithms

Evolving fuzzy clustering [15] has been recently spotted as a promising tool to learn knowledge structures (clusters) in dynamic environments. Our visualization module hinges on an online clustering approach based on the static Shadowed C-Means (SCM) method [11]. The structure and number of the clusters may change across any two consecutive time slots.

#### A. Shadowed C-Means (SCM)

SCM is a partitive clustering algorithm in which each cluster is represented as a shadowed set. It needs the desired number of clusters \( c \) as input and returns a set of cluster prototypes \( V_1, V_2, \ldots, V_c \), a collection of shadowed thresholds \( \lambda_1, \lambda_2, \ldots, \lambda_c \) and a partition matrix \( U = (u_{ik}) \) holding the membership grade of each data pattern \( X_k \) to every prototype \( V_i \), where \( k = 1..N, i = 1..c \) and \( N \) is the number of patterns.

The degrees of membership in SCM are computed as in (1), where \( m \) is the fuzzifier parameter (used to control the overlap among clusters) and \( d_{ik} \) is the distance from \( X_k \) to \( V_i \).

\[
  u_{ik} = \frac{1}{\sum_{j=1}^{c} \left( \frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)}} 
\]

During the prototype formation process, SCM weighs the patterns based on the region they belong to, as in (2).
due to their robustness and interpretability, we developed an incremental clustering algorithm to overcome the ESC limitations that were highlighted in Section I. The advantages of OSC include the ability to model the system’s dynamics in a more realistic fashion as each individual data sample influences the overall risk formulation and an increased real-time responsiveness of the network implementation to intrusion attempts in a secure perimeter.

The Online Shadowed Clustering (OSC) scheme is, like ESC, deeply rooted on the shadowed set formalism so as to deal effectively with noise and outliers as well as to provide an interpretable granulation mechanism for cluster formation. OSC requires memorization of a (usually small) set of data samples in order to account for the update of the λ threshold of the shadowed set attached to each cluster. That is, it maintains a time window TW where the most recent $TWS = TW$ data samples are stored. OSC is formalized in Algorithm 1.

1) Membership and Typicality Grades: OSC utilizes both membership and typicality grades, like in the Possibilistic Fuzzy C-Means [12], to better capture two important facets of the relationship between a data sample $X$ and a cluster prototype $V_i$. The membership grade $u_{iX}$, originally introduced with Fuzzy C-Means (FCM) [2] and computed by (1), expresses the extent to which $X$ belongs to the cluster described by $V_i$ in relation to all other clusters. The typicality grade $t_{iX}$, on the other hand, indicates how well can $X$ represent the cluster prototyped by $V_i$ without consideration of any other knowledge structure (cluster) in the feature space, as shown in (3).

$$t_{iX}[ii] = \frac{1}{1 + \left(\frac{\rho_{iX} \cdot d_{iX}^2}{\gamma_i} \right)^{1/(\eta - 1)}}$$

where

$$\gamma_i = \frac{\sum_{X \in TW} \rho_{iX} \cdot d_{iX}^2}{\sum_{X \in TW} \rho_{iX}}$$

The parameter $\gamma_i$ is used for distance ratio maintenance and specifies the distance at which a data samples is “half way through” of becoming a typical member of the cluster. It is computed on the basis of the memorized data samples and updated after every OSC iteration (see line 41 of Alg. 1).

In OSC, the membership and typicality grades are subsumed into the combined grade $\rho_{iX}$ and computed through (5). Both indicators are weighed by the constants $a$ and $b$, respectively, which are strictly problem-specific. Notice that the presence of the “sign” function is necessary to handle the single-cluster case, where $u_{iX} = 1$ regardless of the values of $X$ and $V_i$ due to the probabilistic calculation framework underpinning FCM.

$$\rho_{iX}[ii] = \frac{\text{sgn}(c[ii] - 1) \cdot a \cdot u_{iX} + b \cdot t_{iX}}{\text{sgn}(c[ii] - 1) \cdot a + b}$$
Algorithm 1 Online Shadowed Clustering (OSC)

**Input:** current cluster distribution $V[i]$, data sample $X[i]$

**Output:** new cluster distribution $V[i+1]$

1. $i \leftarrow 0$;
2. while incoming data sample $X[i]$ arrives do
3.   $i \leftarrow i + 1$;
4.   if $i \equiv 1$ then
5.     $V_i[i] \leftarrow\text{createNewCluster}(X[i])$;
6.     updateTimeWindow($X[i]$);
7.   else
8.     /* grade calculation for new data sample */
9.     compute membership grades $u_{i,X}[i]$ by (1);
10.    compute typicality grades $t_{i,X}[i]$ by (3);
11.    compute combined grades $\rho_{i,X}[i]$ by (5);
12.    updateTimeWindow($X[i]$);
13.    /* clusters react to new data sample */
14.    update $\lambda$ with $X \in \text{TW}$ by (6) iff $|\text{TW}| \geq N_1$;
15.    update cluster prototypes $V_i[i]$ by (7);
16.    /* detect shifts in data stream */
17.    if $X[i] \in \text{XCL}(V_i[i])$, $\forall i \in \{1..c[i]\}$ then
18.       $c[i] \leftarrow c[i] + 1$;
19.       $V_i[c[i]] \leftarrow\text{createNewCluster}(X[i])$;
20.    end if
21.    /* detect drifts in data stream */
22.    update the starvation index of all clusters by (8);
23.    for each cluster $V_i[i]$, $i \in \{1..c[i]\}$ do
24.       if $\phi_i[i] > \bar{\phi}[i] + \delta[i]$ then
25.          drift detected: trigger alarm;
26.       end if
27.    end for
28.    /* delete non-representative clusters */
29.    if $\phi_i[i] > \Omega$ then
30.       deleteClusterPrototype($V_i[i]$);
31.       $c[i] \leftarrow c[i] - 1$;
32.    end if
33.    /* update information in the TW */
34.    update sample grades by (1), (3), (5) $\forall X \in \text{TW}$;
35.    update $\gamma_i$ by (4) $\forall i \in \{1..c[i]\}$;
36.    end if
37. end while

2) Shadowed Set Threshold Update: The update of the $\lambda_i$ threshold associated with each shadowed set representation of a cluster is a critical step in the algorithm. The value of $\lambda_i$ clearly delineates the three regions (core, shadowed, exclusion) of the shadowed set and will determine how memorized and incoming data samples will influence the update of the cluster prototype $V_i$. We follow the “uncertainty relocation” principle outlined by Pedrycz [11] to automate the calculation of the $\lambda$ thresholds. The optimal $\lambda_i$ for each cluster is the one that minimizes expression (6).

$$O(\lambda_i) = \left| \sum_{X \in \text{XCL}(V_i)} \rho_{i,X} + \sum_{X \in \text{CORE}(V_i)} (\rho_{i,max} - \lambda_i) \right| - \text{card}(\text{SDW}(V_i))$$

Notice that OSC uses the combined grades $\rho_{i,X}$ instead of the membership grades $u_{i,X}$ of all data samples in the time window $\text{TW}$ to update $\lambda_i$ as long as $|\text{TW}| > N_1$, where $N_1$ is a built-in parameter usually set to half of the time window size. This prevents the shadowed set regions from being shaped by the influx of an insufficient number of data samples.

3) Recursive Cluster Prototype Update: Once the $\lambda$ thresholds of the clusters have been recalculated in response to the incoming data sample, the next step is to update the cluster prototypes. OSC features a recursive calculation of the new location of prototype $V_i[i]$ based on its previous location $V_i[i-1]$, a memorized term $B_i[i-1]$ that stands for the cumulative influence ratio brought about by all previous data samples, and the impact of the current data sample in the system $\Delta A[i]$ and $\Delta B[i]$. These two latter terms can be computed “on the fly”. The expression governing the relocation of the cluster prototypes is as follows:

$$V_i[i] = \frac{B_i[i] - 1 \cdot V_i[i-1] + \Delta A[i]}{B_i[i] - 1 + \Delta B[i]}$$

where

$$\Delta A[i] = \begin{cases} 
X, & X \in \text{CORE}(V_i[i]) \\
\rho_{i,X}^{\text{min}} \cdot X, & X \in \text{SDW}(V_i[i]) \\
\rho_{i,X}^{\text{ii}}, & X \in \text{XCL}(V_i[i]) \\
1, & X \in \text{CORE}(V_i[i]) 
\end{cases}$$

$$\Delta B[i] = \begin{cases} 
\rho_{i,X}^{\text{min}} \cdot X, & X \in \text{SDW}(V_i[i]) \\
\rho_{i,X}^{\text{ii}}, & X \in \text{XCL}(V_i[i]) 
\end{cases}$$

and $B_i[i] = B_i[i-1] + \Delta B[i]$. Notice how OSC follows in the SCM footsteps by weighing the influence of the incoming data sample $X[i]$ according to the region that it belongs to, with data samples in the core region exercising a strong bias and those in the exclusion region having a negligible bearing. This is where robustness to noisy and outlier samples stems from, as the prototype is...
not drifted in the direction of samples that poorly describe the cluster. Parameters \( m \) and \( \eta \) are outlined in Table I.

4) Drift and Shift Detection: A shift (abrupt transition of a cluster in the feature space) is naturally detected in an incremental clustering scenario when the incoming data sample \( X[i] \) is tagged as an outlier. This occurs only if \( X[i] \) belongs to the exclusion region of ALL existing clusters, as indicated in lines 19-22 of Alg. 1. In such case, a new cluster is formed with \( X[i] \) as its provisional prototype.

A drift (gradual transition of a cluster in the feature space) can be detected in OSC with the help of the starvation index. This indicator \( \phi_i \) is computed for each cluster \( i = 1..c[i] \). When a new cluster is created, its starvation index is set to zero. The arrival of an incoming data sample \( X[i] \) triggers an update of the \( \phi_i \) as displayed in (8).

\[
\phi_i[i] = \begin{cases} 
\max(\phi_i[i-1] - 1, 0), & X \in \text{CORE}(V_i[i]) \lor (X \in \text{SDW}(V_i[i]) \lor (\#j \neq i : X \in \text{SDW}(V_j[i])) \\
\phi_i[i-1] + 1, & \text{otherwise}
\end{cases}
\]

(8)

In other words, only clusters who have been reinforced through the addition of the incoming sample into its core or its shadowed region (exclusively) are not starving in the current time slot. All other clusters will witness their starvation index go up by one unit. When \( \phi_i \) exceeds a permissible threshold \( \Omega \), the cluster is no longer considered relevant and its removal from the system is enforced, as in lines 33-37 of Alg. 1.

VI. EXPERIMENTS

The online shadowed clustering algorithm under discussion has been applied to a novel cooperative networking scenario depicted in [4], in which a mobile robot augments the wireless sensor network with self-healing capabilities by repairing its coverage as soon as a damaged sensor arises. The simulation has been conducted in Microsoft Robotics Development Studio (MRDS)\(^1\) on an Intel Core i7 CPU 860 @ 2.80 GHz with 6 GB of RAM under Windows 7 Home Premium with a 64-bit architecture.

A territorial perimeter around a critical infrastructure is set up by laying out a group of stationary platforms (in white), as shown in Fig. 3. The brown node outside the perimeter is an intruder which will try to compromise the secured area. Each platform is equipped with a built-in, battery-powered sonar device, which allows monitoring its immediate vicinity. The ID of a platform equals that of its only-mounted sensor. Two risk sources (features) per deployed platform are considered: the battery level and the intrusion distance, each bearing an associated risk perception. Whenever the total risk for a sensing unit \( S \) goes beyond a tolerable cap, the mobile robot located inside the perimeter (see Fig. 3) will add the coordinates of \( S \) to its visiting list and eventually move there to better cover the environment in an attempt to detect the intruder and take further actions. The robot also replenishes the battery of the visited sensor.

\(^1\)http://www.microsoft.com/robotics/

![Fig. 3. The simulated outdoor territorial security scenario.](image)

**TABLE I**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>2</td>
<td>[2;4]</td>
<td>Fuzzifier of the membership grade</td>
</tr>
<tr>
<td>( \eta )</td>
<td>2</td>
<td>[2;4]</td>
<td>Fuzzifier of the typicality grade</td>
</tr>
<tr>
<td>( a )</td>
<td>2</td>
<td>[2;5]</td>
<td>Weight of the membership grade</td>
</tr>
<tr>
<td>( b )</td>
<td>2</td>
<td>[2;5]</td>
<td>Weight of the typicality grade</td>
</tr>
<tr>
<td>( TWS )</td>
<td>50</td>
<td>[30;40;50;60;70;80]</td>
<td>Time window size</td>
</tr>
<tr>
<td>( N_1 )</td>
<td>25</td>
<td>TWS/2; TWS/3; TWS/4;</td>
<td>Min. # of memorized samples</td>
</tr>
<tr>
<td>( \Omega )</td>
<td>8</td>
<td>[6;20]</td>
<td>Starvation threshold</td>
</tr>
</tbody>
</table>

The online shadowed clustering algorithm under discussion has been applied to a novel cooperative networking scenario depicted in [4], in which a mobile robot augments the wireless sensor network with self-healing capabilities by repairing its coverage as soon as a damaged sensor arises. The simulation has been conducted in Microsoft Robotics Development Studio (MRDS)\(^1\) on an Intel Core i7 CPU 860 @ 2.80 GHz with 6 GB of RAM under Windows 7 Home Premium with a 64-bit architecture.

A territorial perimeter around a critical infrastructure is set up by laying out a group of stationary platforms (in white), as shown in Fig. 3. The brown node outside the perimeter is an intruder which will try to compromise the secured area. Each platform is equipped with a built-in, battery-powered sonar device, which allows monitoring its immediate vicinity. The ID of a platform equals that of its only-mounted sensor. Two risk sources (features) per deployed platform are considered: the battery level and the intrusion distance, each bearing an associated risk perception. Whenever the total risk for a sensing unit \( S \) goes beyond a tolerable cap, the mobile robot located inside the perimeter (see Fig. 3) will add the coordinates of \( S \) to its visiting list and eventually move there to better cover the environment in an attempt to detect the intruder and take further actions. The robot also replenishes the battery of the visited sensor.

\(^1\)http://www.microsoft.com/robotics/

The risk visualization module governed by the OSC algorithm is offering a visual feedback to the user about the structural dynamics of the perimeter security solution. A data sample containing the feature vector reported by any platform in the outdoor environment is clustered by OSC every 1/8 sec. The simulation spans 45 sec. During that time, the battery level of each node was decreased in proportion to the amount of network traffic in the lower layers and the intruder node performed an attempt to breach the secured perimeter by approaching the nodes in the left side of Fig. 3.

A. Algorithmic Parameters

Table I unveils the parameter list required by OSC. Their values have been determined after an empirical analysis in which relevant ranges were considered.

B. Performance Metrics

The following performance metrics have been used in the simulation study:

1) Average Cluster Stability: For each cluster, we compute its accuracy \( \alpha \), compactness \( \rho \) and stability \( \theta \) as follows:

\[
\alpha(V_i) = \begin{cases} 
-\infty & |\text{CORE}(V_i)| + |\text{SDW}(V_i)| = 0 \\
\frac{|\text{CORE}(V_i)|}{|\text{CORE}(V_i)| + |\text{SDW}(V_i)|} & \text{otherwise}
\end{cases}
\]

(9)
$$\sigma(V_i) = \begin{cases} 
1 & \text{SDW}(V_i) = \emptyset \\
0 & \text{CORE}(V_i) = \emptyset \\
\frac{1}{|\text{SDW}(V_i)|} \sum_{x_k \in \text{SDW}(V_i)} u_{ik} & \text{otherwise} \\
\frac{1}{|\text{CORE}(V_i)|} \sum_{x_k \in \text{CORE}(V_i)} u_{ik}
\end{cases}$$

(10)

$$\theta(V_i) = w \cdot \alpha(V_i) + (1 - w) \cdot \sigma(V_i)$$

(11)

The accuracy $\alpha_j$ of a cluster $V_j$ indicates the proportion of representative patterns in it. The compactness $\sigma_j$ denotes how far in membership are the core and shadowed zones on average. The stability $\theta_j$ amalgamates both measures according to the user-defined importance $w$.

Then the average cluster stability $\overline{\theta}$ is the mean of the stability of all clusters $\frac{1}{c} \sum_{i=1}^{c} \theta_i$.

2) Quality of the Clustering:

$$\vartheta[ii] = \frac{\sum_{i=1}^{c}[ii] |\text{CORE}(V[i][ii])|}{1 + |\bigcup_{i=1}^{c}[ii] \text{SDW}(V[i][ii])|}$$

(12)

This indicator rewards clusterings with a high number of patterns in the core regions and a small number of patterns in the shadowed regions.

3) Structural Variation Index:

$$\Psi[ii] = \sum_{s=1}^{c[ii]} \sum_{t=1}^{c[ii]} \psi(w_{st})$$

(13)

where $w_{st}$ is the membership grade of $V_s[i][ii]$ to $V_s'[i][ii - 1]$ computed as in (1) whereas $\psi(\cdot)$ is a functional that quantifies the degree of structural variation of $V_s[i][ii]$ w.r.t the shadowed set representing $V_s'[i][ii - 1]$. Fig. 4 shows its schematic portrayal. There is no structural variation if the prototype in the present snapshot falls within the core or the exclusion regions of the shadowed set associated with the cluster prototype in the previous data snapshot. The variation degree increases as we approach 0.5, the focal point of the shadowed region.

![Fig. 4. Membership function of $\psi(w_{st})$. The threshold $\lambda_s[i][ii - 1]$ is used.](image)

C. Results

Fig. 5 displays the cluster distribution returned by OSC after 57 data samples have been presented incrementally to the system. While all sensor nodes are very close to each other in the battery level space (since their power sources are almost full at the outset of the surveillance scenario), their distances with respect to the intruder node (feature 2) clearly divide them into two groups, which are well captured by OSC.

Notice how the prototype of the cluster in the bottom region is not spatially located in the centre of gravity of its three members but fairly close to one of them. This was, actually, the first data sample read by the system and the prototype was laid there. The attraction of the two subsequent cluster members was not strong enough to drive the prototype far away from its original location, thus demonstrating the resilience with which it deals with abnormal data. Even more negligible was the influence of the prototypes in the upper part of the figure, as they are atypical to the cluster and thus cast into its exclusion region.

![Fig. 5. OSC results at time slot 57 in the TerrSec scenario.](image)

Fig. 6 displays the fluctuations in average cluster stability throughout the OSC execution. Despite the well-known fact that all incremental clustering algorithms are sensitive to the order in which the data samples are presented and that OSC only updates the cluster distribution based on the impact caused by the incoming data sample, it is encouraging to see how the clusters were quite stable (over 50%) for most of the 8-vector snapshots. The steep decline in stability occurred between time slots 200-208 (snapshot 26) and had to do with a sudden change in the intruder distance feature caused by the advance of the intruder node towards the secure perimeter.

![Fig. 6. Average cluster stability reported by OSC.](image)
The quality of the clustering \( \vartheta \), see expression (12), was also heterogeneous, as portrayed in Fig. 7. OSC captures very well the dynamics of the perimeter-monitored scenario by reallocating the memorized data samples into the core, shadowed or exclusion regions of the existing clusters as their prototypes change in response to the incoming data samples. Again, notice that OSC only creates a new cluster in the vicinity of the current sample and not around past samples. The shift caused by the intruder node in the system around vicinity of the current sample is well reflected in the \( \vartheta [ii] \) indicator too.

Future research efforts will concentrate on eliminating OSC’s current need for memorization, a more thorough parametric sensitivity analysis as well as its empirical comparison versus ESC [3] and other online clustering algorithms in the literature.

ACKNOWLEDGEMENT

The support received from the MITACS Accelerate Program and NRC-IRAP is gratefully acknowledged.

REFERENCES