Article:

Chebiyyam, M., Reddy, R.D., Dogra, D.P. et al. (2 more authors) (2018) Motion anomaly detection and trajectory analysis in visual surveillance. Multimedia Tools and Applications, 77. pp. 16223-16248. ISSN 1380-7501

https://doi.org/10.1007/s11042-017-5196-6

Noname manuscript No.
(will be inserted by the editor)

Motion Anomaly Detection and Trajectory Analysis in Visual Surveillance

Manaswi Chebiyyam · Rohit Desam Reddy · Debi Prosad Dogra · Harish Bhaskar · Lyudmila Mihaylova

Received: / Accepted:

Abstract Motion anomaly detection through video analysis is important for delivering autonomous situation awareness in public places. Surveillance scene segmentation and representation is the preliminary step to implementation anomaly detection. Surveillance scene can be represented using Region Association Graph (RAG), where nodes represent regions and edges denote connectivity among the regions. Existing RAG-based analysis algorithms assume simple anomalies such as moving objects visit statistically unimportant or abandoned regions. However, complex anomalies such as an object encircles within a particular region (Type-I) or within a set of regions (Type-II). In this paper, we extract statistical features from a given set of object trajectories and train multi-class support vector machines (SVM) to deal with each type of anomaly. In the testing phase, a given test trajectory is categorized as normal or anomalous with respect to the trained models. Performance evaluation of the proposed algorithm has been carried out on public as well as our own datasets. We have recorded sensitivity as high as 86% and fall-out rate as low as 9% in experimental evaluation of the proposed technique. We have carried out comparative analysis with state-of-the-art techniques to benchmark the method. It has been observed that the proposed model is consistent and highly accurate across challenging datasets.

Keywords Visual surveillance, Anomalous activity detection, Abnormal behavior classification, Trajectory analysis.

M. Chebiyyam, R. D. Reddy, D. P. Dogra
School of Electrical Sciences, IIT Bhubaneswar, India
E-mail: {mbc10, rrd10, dpdogra}@iitbbs.ac.in

H. Bhaskar
Department of Computer Science and Engineering, Khalifa University, U.A.E
E-mail: Harish.Bhaskar@kustar.ac.ae

L. Mihaylova
University of Sheffield, U.K

1 Introduction

Learning-based methods have boosted autonomous analysis of high-dimensional spatio-temporal video data. Conventional visual analysis requires integration of statistical inference with computational models. Advances in capability of object localization, recognition, and tracking in videos have allowed researchers to think of designing autonomous decision making systems in situation awareness applications.

CCTV-based video surveillance has emerged lately and being given more priority for national security in developing countries. Modern development in autonomous surveillance has surpassed the legacy of requiring human administrators to examine several hours of visual feed [1] to summarize socio-economic scenarios [2] or mining critical forensic evidences [3]. This has been widely supported by the growth of video analytic capabilities in applications such as motion detection [4, 5], tracking [6], parsing [7], activity recognition [8], behavioural understanding [2, 3, 9], traffic analysis [10], parking area monitoring [11], abandoned object detection [12, 13], suspicious activity detection [14], and scene understanding [15]. With rapid developments in autonomous visual surveillance, such analytic systems are becoming highly prevalent in environments such as airports, shopping malls, railway stations and subways in modern smart cities.

One of the key steps towards complete situation awareness is anomalous activity detection. The objective is to determine if visually perceived movements of objects can be categorized as normal or abnormal. This can be used to detect compromise in security. However, what constitutes an abnormal or anomalous motion has remained a topic of debate. While providing a formal definition of anomalous activity seems subjective, its quantification is even more complex. Several behavioural analysis methods have been introduced recently within the visual surveillance domain [16–18]. A major share of these studies build models over the visual perception of target motion and quantifies them based on motion characteristics, thus recognizing anomalous activities. In this context, pattern recognition and machine learning guided techniques to understand and comprehend target motion through trajectory analysis, is gaining significance.

Although the high level semantic analysis of a target motion is critical towards anomaly detection or behavioural studies, target detection and tracking still remain as the fundametal steps. It is obvious because analyzing their motion characteristics is not meaningful without accurate localization of the targets. However, a major share of detection and tracking algorithms are mainly challenged by the dynamic nature of surveillance scenes including sudden changes in illumination, occlusion, target density, etc. The literature surrounding target tracking is rich in content and reviews on some of the key techniques can be found in Yilmaz et al. [19] and [20]. According to these surveys, target tracking techniques include but not limited to context-based approaches [21–23], shape-based methods [24], statistical model-based techniques [25, 26], appearance-based tracking [27], tracking in complex backgrounds [28], tracking using particle filtering, [29–31], tracking in the presence of occlusion [32, 33], fusion-based human tracking [34, 35], etc.

1.1 Related Work

Video anomaly detection algorithms have already been proposed by various researcg groups [36–39]. However, classification of such algorithms can be done in various ways. Anomaly detection can be achieved by segmenting a video in temporal domain, where each segment can be classified into a different category of interest [3, 40]. In such algorithms, the overall change in scene dynamics within any chosen time interval is considered as an important feature towards identifying a temporal segment of interest. Any measured deviations from these supervised segments of interest is often classified anomalous [8]. Anomaly detection can be performed at holistic as well as individual levels. While the holistic viewpoint addresses variations among clustered trajectories within a scene, individual trajectories can also be analysed to determine video segments containing possible anomalous motion. However, such methods compute statistical mean of trajectories during training and use them as a reference to find deviation from normal class centroids. These methods are popular in visual surveillance applications [41, 42]. More recently, Brun et al. [11, 15] have proposed a scene partitioning approach to classify trajectories using unsupervised learning. Main idea in their method is to divide the scene into non-overlapping zones and to represent it using a graphical structure similar to the method proposed in [43]. Brun et al. have clustered car trajectories recorded from parking zones [44] and human trajectories obtained inside a busy railway station [45]. The method proposed in [43] has been used to detect very simple anomalies such as a target visiting abandoned or inaccessible regions. Further, a similar method, often refereed as non-conformal recognition technique has also been recently introduced for maritime applications [46]. According to [46], a conformal anomaly prediction method using a kernel-density estimation based non-conformal measure has been applied to detect suspicious behaviours of ships due to sudden changes in direction, speed or anchoring. In another study, a Sequential Hausdorff Nearest-Neighbour Conformal Anomaly Detector (SHNN-CAD) has been proposed by [47] for online learning and sequential anomaly detection using motion trajectories. Despite the algorithm being parameter-light, the performance of the technique in [47] is highly influenced by the learning procedure and the choice of training set.

The study of spatio-temporal models for analysing local context for anomalous event detection has been gaining recent interest. For example, Cheng et.al. [48] presented a video anomaly detection methodology that aims to localize a visual scene using hierarchical feature representation and to detect variations using Gaussian Process Regression. This method exploits deformable interaction templates modelled using Gaussian Process Regression for anomaly detection. Although, the theoretical framework was well-founded, experimental results could only demonstrate about 80% recognition rate on different datasets. In another study by [49], the classification of the visual scene into dominant versus rare activities using spatio-temporal models and local similarity has been addressed. Activity detection in other domains have also attracted researchers. For example, Liu et al. [50–54] have proposed a handful of methods to detect human activities during various types of interactions. The method in [49] engages the Histogram of Oriented Gradient (HOG) features within an online Fuzzy C-Means clustering technique for behaviour understanding. In a similar study by [55], several local descriptors have been used in conjunction with spatio-temporal filtering and local k-nearest neighbour algorithm for composite training and detection of motion anomalies in video. The local features explored within the work of [55]include persistence, direction and motion magnitude. Finally, the use of spatio-temporal context analysis for the accurate anomalous event detection and localization has been proposed in [56]. Context-aware anomalous activity detection has also been studied in [43, 57]. Here, it has been demonstrated that in order to determine scene dynamics that relate to target interactions, it is critical to analyze the target motion saliency for reliable anomalous activity detection. For example, if a target deviates from its normal path or spends additional time within a confined area of interest, the dynamics need not necessarily suggest anomalous activity. The activity may be normal depending on the context. For example, a predefined place within the viewing field may be one of the region(s)-of-interest, where people usually visit and stays for longer duration. Thus, contextual information plays a vital role in deciding anomalous activities.

Despite recent developments in this research domain, the problem of motion anomaly detection still remains to be challenging due to the following reasons: a) To identify what constitutes anomalous within the surveillance context and distinguishing it from normal behaviour, b) To propose a scene independent model that can work under varying scene conditions, c) To deal with the complexity in behavioural patterns of moving targets, and d) To operate within reasonable computational overhead. Majority of the above mentioned state-of-the-art methodologies do not address the first challenge, whereas the method proposed in [43] has been shown to handle very simple motion anomalies. Although the unsupervised approach proposed by Burn et al. [11, 15] can successfully classify trajectories, this method does not provide any insight on the type and category of the anomaly. In addition to that, threshold-based classification approach adopted in [11] may not be applicable under varying environmental conditions.

1.2 Contributions of the Paper

In this paper, a learning-based methodology has been proposed to detect video anomalies. The method encapsulates contextual information of object movements within a surveillance scene to detect two types of anomalous situations. In accomplishing this, we have made the following technical contributions:

- We propose a statistical approach to extract meaningful features from available object trajectories and train multi-class support vector machines to classify a given test trajectory as normal or anomalous.
- We have defined two distinct types of anomalies, namely Type-I and Type-II. Type-I anomaly is observed when a target is encircling or residing within a region for a prolonged duration of time. Type-II anomaly demonstrates a target switching between two or more regions for a sustained period of time. We present formal mathematical definitions of Type-I and Type-II anomalies.
- Validation of the proposed methodology on publicly available challenging datasets and our dataset and providing a comparative analysis with state-of-the-art techniques in this field of study.

The rest of the paper is organized as follows. In Section 3, the detailed methodology of the proposed solution is described. Results and discussions are illustrated in Section 4. Finally, Section 5 concludes the paper.

2 Proposed Methodology

We present our proposed anomaly detection framework in Fig. 1. The proposed anomaly detection technique operates in two phases, training and testing. In the training phase, support vector machines are trained using the statistical features extracted from a given set of trajectories. Therefore, when the scene is changed, we carry out the training of SVM parameters using the new set of trajectories available. During testing, a trajectory is verified against the trained models. Testing of a trajectory is instanteneous and whenever a given test trajectory violetes the statistical model, we generate the alarm. Statistical feature extraction has been carried out using a method proposed in [43]. These steps are described in the following sections.

2.1 Importance-based Scene Segmentation

Consider that a visual surveillance scene at a given instance of time, captured from a CCTV camera, is represented as an image frame I. Given the scene I and a set of trajectories Δ of moving targets inside that scene, the aim is to build a segmentation map S composed of homogeneous regions, wherein the criterion of homogeneity is based on the importance of each region within that scene. Assuming that the original scene I is uniformly divided into rectangular blocks b, the aim is to decompose I into K number of semantically homogeneous regions each of which is identified by the region-correspondence variable $R_b \in 1, ..., K$. Here, the r^{th} region is the set of blocks B_r, whose region correspondence variable equals r, i.e., $B_r = \{b : R_b = r\}$. The problem of transforming the scene I into the segmentation map S can be mathematically formulated as:

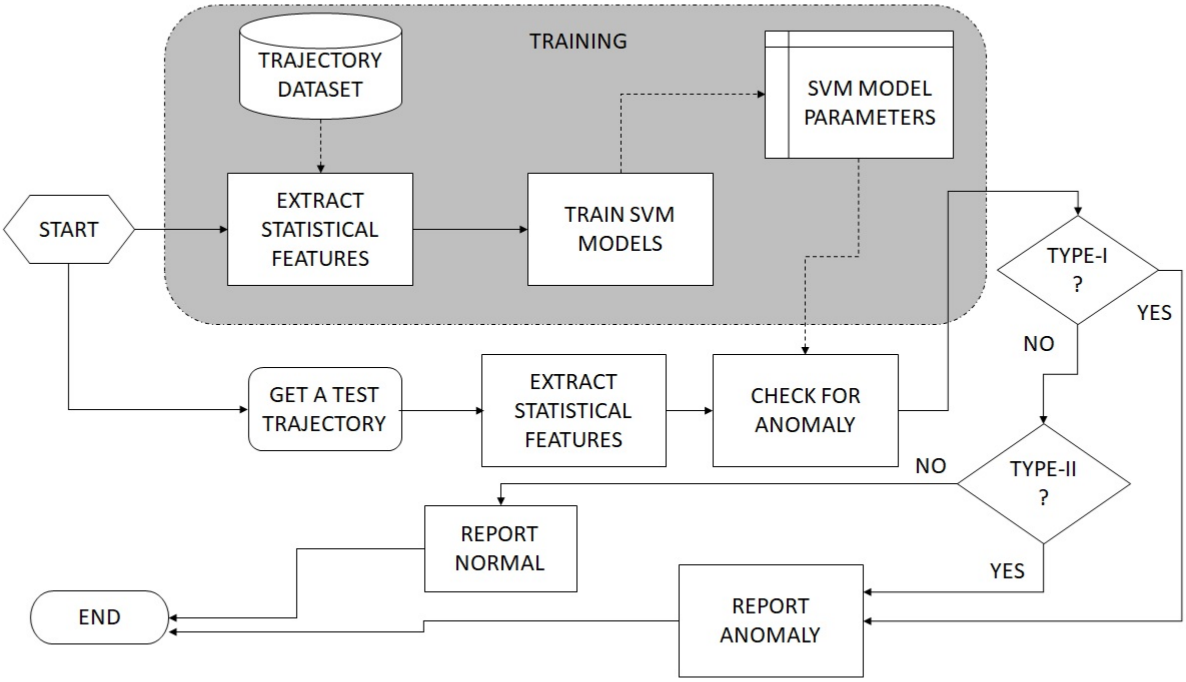

Fig. 1: Flowchart of the proposed anomaly detection algorithm.

$$I \simeq \bigcup_{r} B_r = \bigcup_{b} \{b : R_b = r\} \in 1, ..., K = S. \tag{1}$$

$$R_b = \mathbf{argmax} \overrightarrow{f}(\overrightarrow{b}). \tag{2}$$

$$\forall b f(b) = (i_b) \, . \tag{3}$$

where, $\overrightarrow{f}(\overrightarrow{b})$ represents motion dynamics features extracted from each individual block that is based on the measurement of an importance criterion i_b. Further, blocks with similar importance (i_b) are clustered together in order to build homogeneous-regions within the scene. The importance criterion corresponding to each block is based on motion dynamics features including the velocity of targets within each block and the overall time spend by the target while visiting the block, i.e.

$$i_b = \forall b \frac{\rho_b}{g_b}. \tag{4}$$

where, ρ_b represents the popularity index of the block b and is normalized against the total number of times a block b is visited by different targets (represented as g_b).

$$\rho_b = \rho_b + \frac{\overline{v}^o - v^o_j}{\overline{v}^o}. \tag{5}$$

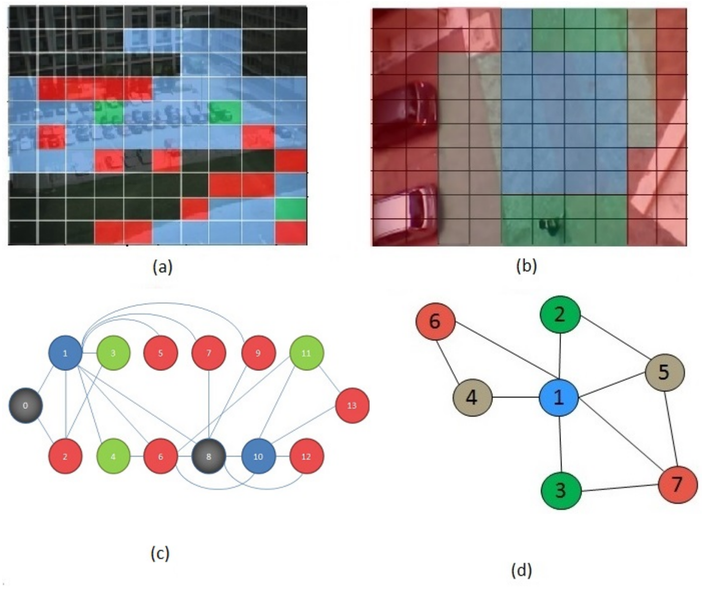

Fig. 2: (a-b) Segmentations of surveillance scenes (MIT and In-HOUSE datasets) using the method proposed in [43]. (c-d) Corresponding RAG representations of the scenes. For example, in (a) Green - mostly accessed blocks (L_1), Blue - frequently visited blocks (L_2), Red - rarely visited blocks (L_3) and Black/Gray - in-accessed blocks (L_4).

Here, it is assumed that the instantaneous velocity (v_j^o) of a target is expected to be lower than its average velocity ($\overline{v}^o$). Further to analyzing the importance distributions from different surveillance scenes, region-labels were chosen to be discretized into 4 classes $\{R_b : \{1, 2, 3, 4\}\}$: interesting blocks L_1 represented as the local maxima within the importance distribution, frequently visited blocks L_2, rarely visited blocks L_3 and in-accessed blocks L_4.

A simple 8-connected component analysis is engaged for clustering the class labels indicating relative importance of blocks in order to generate homogeneous regions. Two examples of the scene segmentation are illustrated in Fig. 2(a-b).

2.2 Scene Representation using RAG

In order to enable seamless inference on the segmented scene S, a RAG represented as $G(V, E)$, where $v \in V$ and $e \in E$ represents nodes and edges of the graph, is constructed. Each homogeneous region in the segmentation map S from the previous step is assigned to a corresponding vertex in the RAG and adjacent regions in the scene are connected through edges. The weight for each vertex $v(w)$ in the RAG G is estimated as follows. Assume z independent trajectory segments pass through a vertex v_i, then its weight $v_i(w)$ is computed using (6), where s_j is the length of the j^{th} trajectory segment passing through v_i. Note that, nodes that are labeled in-accessible, are initialized with zero weight,

$$v_i(w) = \frac{s_1 + s_2 + ... + s_z}{z}. \tag{6}$$

The structure of the constructed RAG represents the overall connectivity of various regions that constitute a surveillance scene. Therefore, given a test trajectory recorded over the same scene, it is possible to benchmark this test trajectory against the RAG and hence categorize its motion into normal or anomaly class.

3 Motion Anomaly Detection

According to the proposed method described so far, a surveillance scene I has been shown to be transformed into a segmented scene S represented using a weighted RAG G, based on motion dynamics features extracted from target trajectories. Some examples of the RAG representation of surveillance scenes are illustrated in Fig. 2(c-d). Although, the proposed method relies on localised importance estimated across blocks of regions; estimating importances using other low-level features such as the use of context [58], trajectory density [11, 15], motion [59, 60], spatio-temporal scene structure [61] are also possible.

In the previous work of [43], it has been shown that the RAG-based representation could be exploited to detect anomalies by categorizing movements of targets into specific regions of interest. However, anomaly in typical surveillance scenes appears in more complex forms. Therefore, in this paper, two complex types of anomalies have been chosen to be handled. Type-I anomaly describes motion activity of a target that is encircling or eventually residing within one specific region of interest for a prolonged duration of time and Type-II anomaly demonstrate a target switching between two or more regions of interest. In Fig. 3, an illustration of the two classes of anomalous activities is depicted. It can be noted that it is possible to generalize any type of motion anomaly by decomposing it as a combination of Type-I and Type-II anomalies. For example, the motion pattern representing a random movement of a target around various regions of a scene can be easily mapped as a combination of Type-I and Type-II anomalies. If such a motion pattern is associated with variations in velocity, the time spent in any regions with or without unexpected repetitive transitions between regions, then such a pattern can be conveniently classified as anomalous. In Sections 3.1 and 3.2, detailed modeling and analysis of Type-I and Type-II anomalies are described.

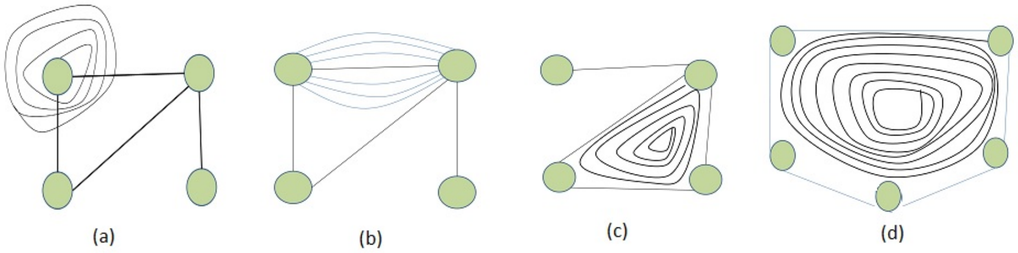

Fig. 3: Demonstration of possible representation of two types of anomalous situations described in this paper. (a) Type-I anomalous situation. (b-d) Type-II anomalous situations.

3.1 Analysis of Type-I Anomaly

In this paper, a Type-I motion anomaly is classified based on the relationship between the motion characteristics of a target that appears within a region of the scene and the type (label) of region within the scene that the target visits. For example, if a target is found to have spent more than usual time within a region that has been previously labelled either as rarely visited (L_3) or not visited at all (L_4), then Type-I anomaly is flagged. Here, the approach could be to compare the test trajectory of a target to the statistical data acquired during training to determine deviations. This paper proposes a probabilistic approach to the classification of Type-I anomaly as defined below. Let a set of κ training trajectories $\{\tau_1, \tau_2, ..., \tau_\kappa\}$ be denoted using (7) and a surveillance scene I be represented using a RAG, say $G(V, E)$, where (8) and (9) represent set of vertices $\{v_1, v_2, ..., v_n\}$ and edges $\{e_1, e_2, ..., e_h\}$, respectively,

$$\Delta = \{\tau_1, \tau_2, \tau_3, ..., \tau_\kappa\}, \tag{7}$$

$$V = \{v_1, v_2, ..., v_n\}, \tag{8}$$

$$E = \{e_1, e_2, ..., e_h\} \tag{9}$$

Each vertex in the RAG represents one homogeneous region that has been labeled during training. An edge (e) between two vertices, is drawn if regions are adjacent. For a given RAG (G) and a set of training trajectories (Δ) of targets within the surveillance scene, the probability $p(v_j|\Delta)$ of a target visiting vertex (v_j) is first estimated. Further, given an unknown trajectory, say τ_{test}, the objective is to exploit the estimated probability $p(v_j|\Delta)$ as features for the classification of the motion patterns into anomalous or normal classes. Consider that the trajectories as given in (7) be referred to as the training set.

A k^{th} trajectory (τ_k) can be considered as a time-series data as given in (10), where (u_i, v_i) pair denotes the location of the target at a given time instant.

$$\tau_k = \{(u_1, v_1), (u_2, v_2), ..., (u_o, v_o)\} \tag{10}$$

The first step is to transform the trajectory of the moving target represented as τ_k into an equivalent path w.r.t the RAG generated from the scene. Let ρ_k denote the equivalent path of τ_k covering m vertices as given in (11), where $|v_j|$ represents the duration of time spent by the target within vertex v_j,

$$\rho_k = (v_1, |v_1|) \rightarrow (v_2, |v_2|) \rightarrow ..... \rightarrow (v_{m_k}, |v_{m_k}|). \tag{11}$$

Further, the probability of a moving target visiting a particular vertex in the RAG can be estimated using a set of paths. That is, given a set of trajectories (Δ), it is possible to estimate the general probability of a target visiting vertex v_j using (12), where $|\rho_i^{v_j}|$ and $|\rho_i|$ represent the duration of time spent by the target within vertex v_j and total duration of its trajectory (τ_i), respectively.

$$p(v_j|\Delta) = \frac{|\rho_1^{v_j}| + |\rho_2^{v_j}| + ... + |\rho_k^{v_j}|}{|\rho_1| + |\rho_2| + ... + |\rho_k|} \tag{12}$$

Similarly, the probability of a target visiting a vertex v_j according to its own trajectory or path (ρ_i) can be estimated using (13)

$$p(v_j|\rho_i) = \frac{|\rho_i^{v_j}|}{|\rho_i|}. \tag{13}$$

Equations (12) and (13) in combination can be used to prepare the training samples for classification. In the proposed approach, a separate classifier for each vertex is trained using the feature vector as given in (14) representing the trajectory path (ρ_i) with its corresponding label vector as represented using (15),

$$F_{\rho_i} = [|\rho_i^{v_1}|, |\rho_i^{v_2}|, ..., |\rho_i^{v_{m_i}}|]^T, \tag{14}$$

$$L_{\rho_i} = [l_{\rho_i}^{v_1}, l_{\rho_i}^{v_2}, ..., l_{\rho_i}^{v_{m_i}}]^T, \tag{15}$$

where

$$l_{\rho_i}^{v_j} = \begin{cases} +1 \text{ when } p(v_j|\rho_i) \geq \beta \times p(v_j|\Delta), \\ -1 \text{ when } p(v_j|\rho_i) < \beta \times p(v_j|\Delta) \end{cases} \tag{16}$$

where β is a multiplication factor that determines the class of a given training trajectory with respect to all trajectories present in the training set. Based on the choice of β, data is prepared for training the classifiers. More precisely, $beta$ is set based on the average time spent by any target within a region and by comparing a given training trajectory's motion dynamics against this average value, a class is determined. Equations (14)-(16) are used to train the classifiers, one for each vertex. Now, given a test trajectory (τ_{test}) or path (ρ_{test}), $F_{\rho_{test}} = [|\rho_{test}^{v_1}|, |\rho_{test}^{v_2}|, ..., |\rho_{test}^{v_m}|]^T$ is computed and used as a feature vector to classify ρ_{test} against the trained m classifiers. If one of these classifiers returned a positive value, the test sample ρ_{test} is assumed to contain Type-I anomaly involving the vertex represented by the index of the respective classifier. It may be noted that β is a critical parameter that can potentially influence the accurate classification of motion patterns. Hence, experimental validation of the effect of the parameter β on the performance of the proposed model is studied and reported in Section 4.5.3.

3.2 Analysis of Type-II Anomaly

As mentioned earlier, a target could visit several regions of a surveillance scene in different motion patterns. However, scenarios depicted in Figs. 3(b-d) can be considered as potential candidates for Type-II anomaly. It is clear that Type-II anomaly is more complex than Type-I anomaly. In contrast to Type-I anomaly, where finding the path within the RAG is critical, Type-II requires finding cycles in a path. Therefore, SVM classifiers are constructed for each candidate cycle in order to determine anomaly. Formally, the detection of Type-II anomaly shall exploit the probability $p(c_j|\Delta)$, where c_j represents a ℓ-cycle segment defined when a loop covering ℓ distinct nodes is identified from a given RAG G and a set of training trajectories (Δ). Further, the objective is to use this probability $p(c_j|\Delta)$, for every possible ℓ-cycles present in G according to a given training set Δ to classify an unknown trajectory τ_{test} as anomalous or normal.

However, detecting cycles of any length (closed walk or simple cycle) in a graph is a NP-complete problem. It has been shown in [62] that this problem can be solved in either $O(VE)$ time or $O(V^\omega \log V)$ time when the length of the cycle is known *a priori*. Similarly, Flum and Grohe in [63] have shown that counting of cycles and paths of length ℓ in both directed and un-directed graphs is not NP-complete for a given ℓ. It has been proved that, when $3 \leq \ell \leq 7$, it is possible to count all ℓ-cycles (a cycle that includes ℓ distinct nodes) in $O(V^\omega)$ time. In order to alleviate the complexity of solving an NP-complete problem and for simplicity, in this paper, it has been chosen to restrict the search for cycles in the RAG to a maximum length of 4, i.e., ≤ 4.

Let, $C = \{c_1, c_2, ..., c_\psi\}$ denote a set of cycles of length ≤ 4 present within a given RAG G. Thus, a number ψ of SVM classifiers are trained, one for each cycle using a set of training trajectories Δ. However, the features required to train these classifiers are relatively more complicated than those used during Type-I anomaly analysis. In addition, searching for ℓ-cycles from a given target trajectory is also a hard problem. Although, a brute-force approach can be employed to solve this, such methods are not efficient and to process a large number of trajectories using this approach would not be computationally tractable. Therefore, the following mechanism has been adopted to find ℓ-cycles from a given trajectory.

A path, say $\rho_i \in \Delta$, is initially represented using (11) with each vertex expanded by its weight. For example, the path given in (11) can be expanded into $\rho_i = (v_1 \rightarrow v_1 \rightarrow ... \rightarrow v_1)_{|v_1|} \rightarrow (v_2 \rightarrow v_2 \rightarrow ... \rightarrow v_2)_{|v_2|} \rightarrow ..... \rightarrow (v_m \rightarrow v_m \rightarrow ... \rightarrow v_m)_{|v_m|}$, where a vertex in this expanded path is assigned a unique but non-linearly quantized label. Thus, Ψ non-linear quantization labels, e.g. $q_1, q_2, ..., q_\Psi$ are selected to represent a path. Such an expansion and non-linear quantization is necessary to determine sparsity of transitions between any pair of nodes. Now, given an expanded representation of the path, it can easily be differentiated from q_j in order to localize the transitions between two nodes u and v. Because of using the non-linear quantization approach, $\frac{\rho_{i+1}-\rho_i}{q_{j-1}-q_j}$ can be used to unambiguously locate transitions between any pair of nodes. Once all such transitions have been identified, cycles can be localized by grouping transitions present within each path. By parsing the path for a contiguous re-appearances of similar transitions, only such desired transitions can be distinguished from others and extracted. In addition, while searching for cycles involving nodes, say u and v, the sparsity in transitions is also checked and if found highly sparse, it is assumed that the path does not indicate any aberrant behavior.

In the proposed method, the frequency of transitions between any pair of nodes within a path is used to estimate the probability of transitions for a given target. Further, given a set of training trajectories (Δ), transitions that represent similar repetitive cycles are marked, features are extracted and fed into the classifiers. Probabilistic measures similar to those introduced in (12) and (13) of Section 3.1 are used to train these classifiers. However, the estimation of general as well as individual probability values are done exclusively. Equations (17)-(18) illustrate the general probability of a ℓ-cycle given all trajectories in the training dataset, where λ represents the minimum number of times a cycle c_j must be repeated by a target in order to consider the trajectory to be anomalous, γ_i represents those instances where cycle c_j appeared more than λ times and η_i denotes the total number of times that the cycle appeared within the whole dataset,

$$\begin{aligned} p(X_{c_j} > \lambda|\Delta) & \\ &= \frac{\gamma_1(X_{c_j} > \lambda+1) + \gamma_2(X_{c_j} > \lambda+2) + ...}{\eta_1(X_{c_j} = 1) + \eta_2(X_{c_j} = 2) + ...} \\ &= p(X_{c_j} = \lambda+1|\Delta) + p(X_{c_j} = \lambda+2|\Delta) + ... \end{aligned} \tag{17}$$

where X_{c_j} indicates the expected number of occurrences of the cycle c_j in a given path,

$$p(X_{c_j} = \lambda + r|\Delta) = \frac{\gamma_1(X_{c_j} > \lambda + r + 1)}{\eta_1(X_{c_j} = 1) + \eta_2(X_{c_j} = 2) + ...}. \tag{18}$$

Similarly, the probability of a ℓ-cycle to appear in a particular path (ρ_i) can be estimated with respect to all the cycles present inside it. That is, Equations (19)-(20) can be used to estimate the probability of a cycle with more than λ repetitions, where ω_i represents the instances when the cycle appeared more than λ times and ϕ_i denotes the total number of times that the cycle was identified in ρ_i,

$$\begin{aligned} p(X_{c_j} > \lambda|\rho_i) & \\ &= \frac{\omega_1(X_{c_j} > \lambda + 1) + \omega_2(X_{c_j} > \lambda + 2) + ...}{\phi_1(X_{c_j} = 1) + \phi_2(X_{c_j} = 2) + ...} \\ &= p(X_{c_j} = \lambda + 1|\rho_i) + p(X_{c_j} = \lambda + 2|\rho_i) + ... \end{aligned} \tag{19}$$

such that,

$$p(X_{c_j} = \lambda + r|\rho_i) = \frac{\omega_1(X_{c_j} > \lambda + r + 1)}{\phi_1(X_{c_j} = 1) + \phi_2(X_{c_j} = 2) + ...} \tag{20}$$

Equations (17) and (19) are used to prepare the training samples necessary for Type-II anomaly classification using ψ classifiers. A feature vector is constructed for each training trajectory with its corresponding label vector in a manner similar to that introduced for Type-I analysis.

$$F_{\rho_i} = [|\rho_i^{c_1}|, |\rho_i^{c_2}|, ..., |\rho_i^{c_\psi}|]^T \tag{21}$$

$$L_{\rho_i} = [l_{\rho_i}^{c_1}, l_{\rho_i}^{c_2}, ..., l_{\rho_i}^{c_\psi}]^T. \tag{22}$$

The corresponding label vector as given in (22) can be generated using (23)

$$l_{\rho_i}^{c_j} = \begin{cases} +1 \text{ when } p(X_{c_j} > \lambda|\rho_i) \geq \beta \times p(X_{c_j} > \lambda|\Delta) \\ -1 \text{ when } p(X_{c_j} > \lambda|\rho_i) < \beta \times p(X_{c_j} > \lambda|\Delta) \end{cases}. \tag{23}$$

Finally, Equations (21)-(23) are used for training the ψ classifiers, one for each cycle. Now, given a test trajectory or path (ρ_{test}) as given in (24) can be computed and used to test ρ_{test} against the ψ classifiers.

$$F_{\rho_{test}} = [|\rho_{test}^{c_1}|, |\rho_{test}^{c_2}|, ..., |\rho_{test}^{c_\psi}|]^T \tag{24}$$

It is assumed that if at least one of the classifiers returned negative, then ρ_{test} is considered abnormal. The mathematical formulation of Type-II anomaly is dependent on only one free parameter λ, which can usually be decided based on the application requirements. For example, to ensure tighter security it may be required to enforce that a target should not encircle a set of regions more than once, where-in λ is set to 1. Although this requirement may vary from one application to another depending on the nature of security, it does not affect the performance of the proposed algorithm.

3.3 Classification

Prediction of unlabelled data is highly important in a typical classification problem. Researchers have used prediction-based techniques in various applications [64–67]. The same logic applies to the present work. Since SVM is a well established predictor, it has been used for the classification of both Type-I and Type-II anomalies. SVM is a set of supervised learning methods used for classification, regression and outliers detection [68]. Though, SVM classifiers are best known for binary classification, however, it has also been used for multi-class separation [69]. With different kernel functions that can be specified for a decision function, SVM is widely used to find the optimal hyper-plane between two classes. In the proposed problem, F_ρ from either Type-I or Type-II anomaly is considered as the feature vector, fed as input x to these SVM classifiers. The objective is to learn a classifier $y = f(x, \alpha)$, where α are the parameters of the function using a linear kernel as the decision function $f(.)$. Given training data (x_i, y_i) for $i = 1, 2, 3......, N$ with $x_i \equiv F_{\rho i}$ and $y_i \in \{-1, +1\}$, a typical 2-D linear classifier is of the form as given in (25), where , ϵ is the normal to the line and b is the bias. Though, ϵ is known as the weight vector, it can be learned by through [68]

$$f(x_i) = \epsilon^T x_i + b. \tag{25}$$

Using the perceptron algorithm, ϵ is initialized to 0. Then an iterative training is performed and Depending on the classification of x_i during iterative processing, ϵ is updated using (26)

$$\epsilon = \epsilon + \alpha \ \text{sign}(f(x_i))x_i. \tag{26}$$

The training process is continued till all data points are correctly classified. At the end of training, final value of ϵ is computed using (27)

$$\epsilon = \sum_{i}^{N} \alpha_i x_i. \tag{27}$$

However, the above training algorithm works only if the data is linearly separable. Finally, the line of classification can be determined using (28), where x_i's are the support vectors of the classifier, i.e.

$$f(x) = \sum_{i} \alpha_i y_i (x_i^T x) + b \tag{28}$$

Since $\epsilon^T x + b = 0$ on the classifier, ϵ is normalized such that $\epsilon^T x_+ + b = 1$ and $\epsilon^T x_- + b = -1$ hold for positive and negative support vectors, respectively. The margin between these support vectors is given as $\frac{2}{||\epsilon||}$. It is possible that multiple solutions for ϵ could be obtained that can satisfy above conditions. However, there is a trade-off between the margin and number of incorrect classification on training data. Thus, learning parameters of the SVM can be formulated as a quadratic optimization problem to separate two classes with maximum distance, subject to linear constraints, e.g. $y_i(\epsilon^T x_i + b) - 1 + \xi_i \geq 0$ and $\xi_i \geq 0$, with unique minimum as given in (29), where ξ is the error in classification. A small η gives larger margins and allows constraints to be easily ignored and a larger η value makes the constraints hard to ignore. Usually, SVM finds a large discrimination hyper-plane when a maximum margin is used.

$$|\epsilon|^2 + \eta \sum_{i}^{N} \xi_i \tag{29}$$

4 Experimental Results

In this section, experimental details and results that validate the performance of the proposed methodology and comparisons against the state-of-the-art using various dataset are reported.

4.1 Description of Datasets

Three datasets including the Grand Central Station dataset (CUHK) [45], MIT trajectory dataset [44] and a custom-built in-house dataset were chosen for performance evaluation. All chosen datasets encapsulate scenarios that illustrate various types of anomalies as described earlier. While the in-house dataset represents both Type-I and Type-II anomalies, the public datasets have scenarios that only demonstrate Type I anomaly.

- The CUHK dataset sequences were recorded in a public place consisting of a large number of targets moving in real-world under no supervision. The CUHK sequence is a 34 minutes long video with nearly 700 target-trajectories. After systematic post-processing, the original set of trajectories were merged into 40 clean trajectories.
- The MIT dataset was recorded for analyzing movements of cars within a parking area. The dataset is composed of nearly 40000 trajectories without manual ground-truth information. The trajectories were obtained using an automatic tracking algorithm.
- Since, none of the available public datasets contained Type-II anomaly, an in-house dataset was used for the validation. The custom-built in-house dataset was recorded for 70 minutes using a static camera hosted on top of a building. We used static video cameras (25 FPS) to record the videos. These cameras were setup on top of the buildings to record videos. These cameras supported HD as well as SD quality video recording. However, we down-scaled the video frames to 640x480 during processing. The dataset was prepared to contain 105 trajectories of targets representing public movement on a busy working day. Some of the chosen trajectories were recorded in a supervised manner covering actions that demonstrate both Type-I and Type-II anomalies.

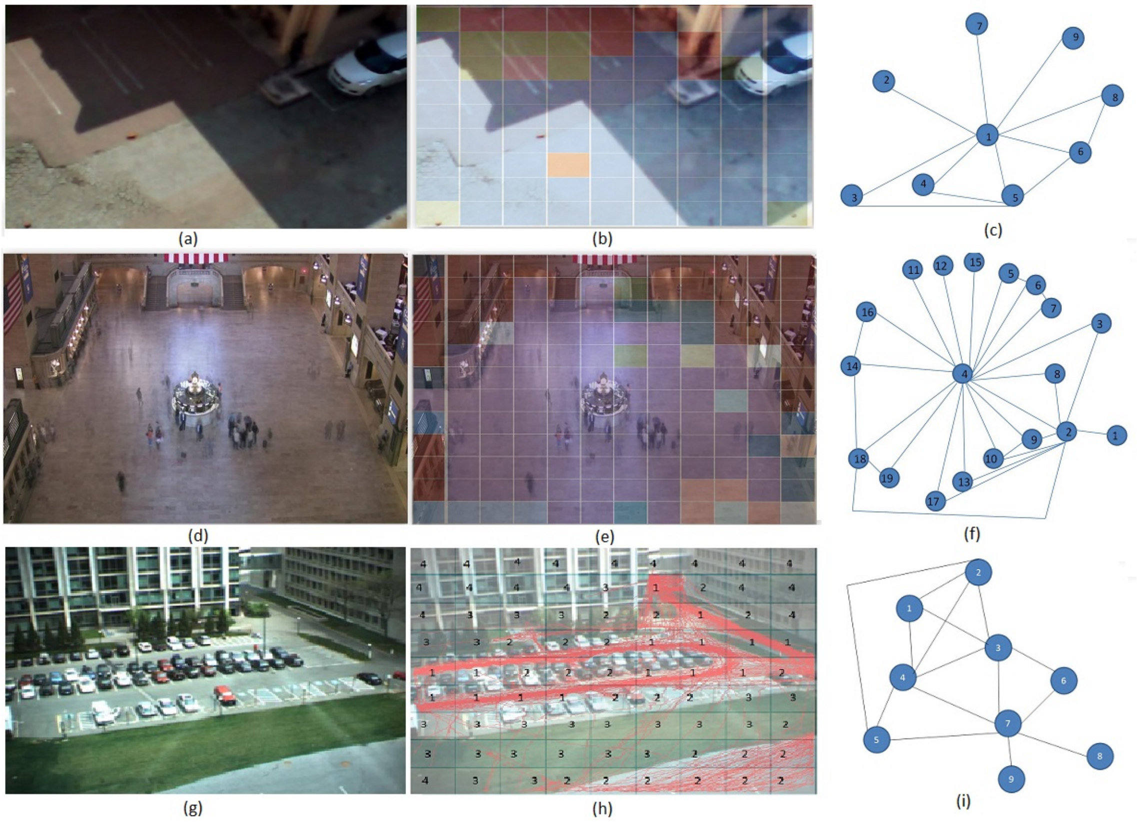

Fig. 4: (a-c) Original scene, segmentation, and corresponding RAG of the custom dataset. (d-f) Original scene, segmentation, and corresponding RAG of the CUHK dataset. (g-i) Original scene, segmentation, and corresponding RAG of the MIT dataset.

Target-trajectories have been extracted using the target detection and tracking algorithm proposed in [22]. The Context Tracker (CT) [22] was used to extract the motion trajectories of the moving targets from these datasets. This method was found to be robust against small camera movements and variations in lighting conditions. Trajectories of individual targets were extracted independently. In all experiments, about 80% of the total trajectories have been used for training, that is, to create the RAG and for estimating the probability values as defined in the algorithm mentioned in the previous section. The remaining 20% of target-trajectories were used for testing. Experiments were validated by measuring the recognition accuracy using 10-fold cross validation, where-in the whole dataset was divided into 10 parts and among them 8 parts were considered for training and 2 for testing. Ground-truth data on Type-I and Type-II anomalies were generated manually. During training, in order to create the RAG, the method proposed in [43] was used. In Fig.4, images representing the original background, segmentation, and the corresponding RAG representation generate using [43], are illustrated.

Table 1: Results of classification on Type-I and Type-II anomalies using $\lambda = 1$ and $\beta = 1$.

Trajectory #	Classification					
	Type-I			Type-II		
	Ground Truth	Detected	Performance	Ground Truth	Detected	Performance
92	5,8	5,6,8	FP:9	(6,8)(5,6)	(5,6)(6,8)	FP:1
93	5,8	5,6,8	FN:0	none	(1,6)(5,6)	FN:0
94	1,6	1,5,6	TN:96	(5,6)(1,6)(6,1,5,6)	(1,6)(5,6)(6,1,5,6)	TN:4
95	1,5	1,3,5	TP: 24	(1,5)	(1,3)(1,5)	TP:10
96	1,5	1,5	Precision:72.7%	none	none	Precision: 90.9%
97	1,5,6	1,5,6		(1,5)	(1,5)	
98	1,5,6,8	1,5,6,8,9		(1,3)(1,6)(6,1,5,6) (1,8,6,1)(1,8,6,5,1)	(1,3)(1,6)(1,7) (6,1,5,6)(1,8,6,1)(1,8,6,5,1)	
99	5,6	5,6	Recall:100%	(1,6,5,1)	(1,6,5,1)	Recall:100%
100	1,5,6,8	1,3,5,6,8		(1,5)(1,8)(5,6)	(1,5)(1,7)(1,8)(4,5)(5,6)	
101	none	none	Accuracy:93%	none	none	Accuracy:93.3%
102	5,6	4,5,6		(1,5)	(1,5)(1,6)	
103	5,6	3,4,5,6		(1,5)	(1,3)(1,5)	
104	1	1		none	none	
105	1,5	1,5		(1,5)	(1,5)	
106	none	none		none	none	

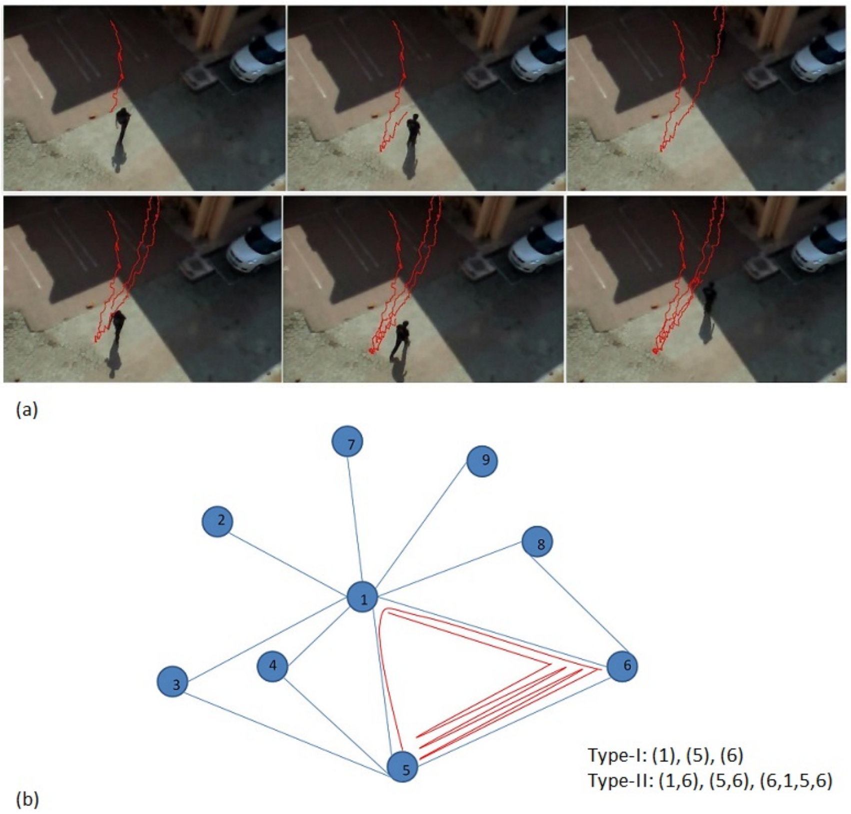

Fig. 5: (a) A test trajectory (#94) of the in-house dataset showing Type-I and Type-II anomalous events. (b) RAG representation of the scene and corresponding intra-vertex and inter-vertex movements causing Type-I and Type-II anomalies.

4.2 Results Using Public Dataset

First, experimental results of the proposed method using the CUHK and MIT public datasets are presented. Out of the 40 trajectories extracted from the CUHK dataset, 30 have been used for training and the remaining 10 for testing. However, no cross validation was done for this experiment due to limited number of clean trajectories that were used to build the model. The original scene, result of segmentation, and the corresponding RAG representation of the scene are presented in Fig. 4(d-f). As illustrated in the RAG presented in Fig. 4(f), 19 different possibilities of Type-I anomaly, equivalent to the number of nodes in the RAG, could be determined. Similarly, it has also been verified from the RAG that Type-II anomaly with $c_j = \{2, 3, 4\}$ may appear in the graph in 32, 72, and 5 ways, respectively. However, no target-trajectory in real-world representing the Type-II anomaly could be found on this dataset. $p(v_j|\varDelta)$ and $p(v_j|\rho_i)$ vales were used to train classifiers associated with Type-I anomaly. The value of β as mentioned in 16 has been varied between 0.5 and 3.0 during experimentation. The results of classification are presented in Table 2.

Table 2: Results of classification (Type-I anomaly) of 10 test trajectories of CUHK dataset using $\beta = 1$.

Trajectory #	Classification (Type-I)		Performance
	Ground Truth	Detected	
31	18,19	18,19	FP:3
32	none	none	FN:0
33	none	none	TN:174
34	15,18	15,18,19	TP: 8
35	14	14	Precision:72.7%
36	none	none	
37	none	none	
38	15,17	15,17	Recall:100%
39	none	none	
40	4	4,11,17	Accuracy:98.3%

On the MIT dataset, a subset of 400 trajectories have been used to validate the proposed algorithm. Out of the 400, 320 were used to construct the RAG and to estimate the features necessary for training the SVM classifier as depicted in Fig. 4(i). The remaining 80 trajectories have been used for the classification of Type-I anomaly. It is clearly evident from the RAG shown in Fig. 4(i), that Type-I anomaly can possibly appear in 19 different ways. As per the available ground truth, there were 72 instances of Type-I anomalies present in the whole test set covering the 80 test trajectories. The proposed algorithm was successful in detecting 109 instances of anomalies assuming $\beta = 1$. When compared against the ground truth, the proposed method recorded false positive, false negative, true positive, and true negative values of 50, 14, 58, and 331, respectively. Precision, recall, and accuracy values were found to be 53.7%, 80.5%, and 85.8%, respectively.

Examples of Type-I anomaly occurring in trajectories of the CUKH and MIT datasets are presented in Fig. 6. It may be observed that the target in question (represented by trajectory #35 of CUHK dataset) has spent more time in region 14 as compared to the average staying probability inside that region from the training data. A similar observation could was also recorded for trajectory #362 of the MIT dataset and the results are shown in Fig. 6(c-d). The results shown in Fig. 6(c-d) highlight the occurrence of Type-I anomaly in nodes 1, 2, and 4, respectively. In general, it has been observed that the rate of false positives generated by the proposed algorithm could be higher in comparison to a significantly low number of false negatives (0 in CUHK dataset

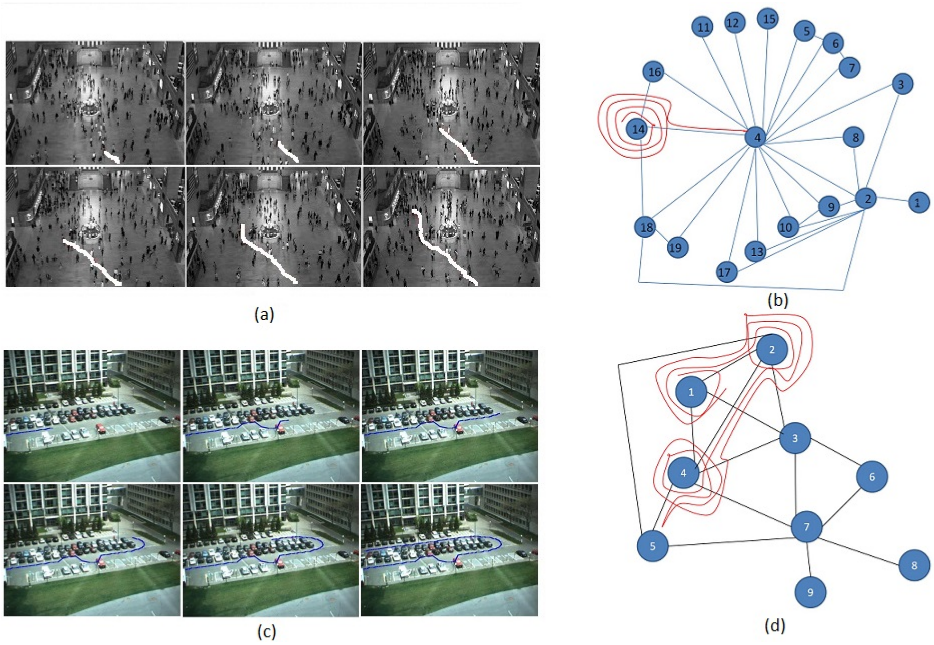

Fig. 6: (a) Frames corresponding to a Type-I anomaly in a test trajectory (#35) of the CUHK dataset. (b) RAG representation of the scene and corresponding intra-vertex movements causing the anomaly. (a) Frames corresponding to a Type-I anomaly in a test trajectory (#362) of the MIT dataset. (b) RAG representation of the scene and corresponding intra-vertex movements causing the anomaly.

and 14 in MIT dataset); thus making the proposed method quite suitable to support visual analytic solutions in real-time.

4.3 Results Using Custom Dataset

In addition to results on the public datasets, experimental evaluation on a custom-built in-house dataset has also been performed. The in-house dataset consists of 105 target-trajectories, out of which 90 have been used for RAG creation and feature extraction. Within the in-house dataset, Type-I anomalies can possibly appear in 9 ways matching the number of nodes present in the RAG as shown in Fig. 4(c). On the other hand, the possibilities of Type-II anomaly is a factor of the cycle length. Therefore, searching for all possibilities is computationally intractable and hence experiments have been limited to search for Type-II anomaly assuming cycles of length ≤ 4. Out of the total 39 instances of Type-II anomalies, cycles with 2, 3, and 4 nodes have appeared 12, 24, and 3 times, respectively. Finally, the total number of Type-I and Type-II anomalies present in the in-house dataset as per the ground-truth, are 9 and 39, respectively.

In the analysis of Type-I anomaly, $p(v_j|\mathit{\Delta})$ and $p(v_j|\rho_i)$ are estimated using the training set and presented to the classifier. Finally, features are computed from the test set and are fed to the classifier. Results of classification are presented in Table 1. For Type-II anomaly, the SVM classifier is trained using $p(X_{c_j} > \lambda|\mathit{\Delta})$ and $p(X_{c_j} > \lambda|\rho_i)$ for cycles of maximum length 5. Further, it has been assumed that $1 \leq \lambda \leq 6$ and β is varied between 0.5 and 3.0. The target-trajectories for $c_j = \{2, 3, \&4\}$ are analysed separately and the results of classification are summarized in Table 1. Examples of Type-I and Type-II anomalies from a single test trajectory are presented in Fig. 5. It may be observed that the chosen test trajectory has Type-I anomaly within the vertices 1, 5, and 6, whereas, inter-vertex movements between 1-6, 5-6, and 6-1-5-6 are directly related to Type-II anomalies. It can be verified from Table 1, that the proposed method is capable of localizing both types of anomalies with accuracy as high as 93%.

4.4 Comparative Analysis

In this section, a comparative analysis of the proposed approach against the state-of-the-art method of Brun et al. [11] using the various datasets are presented. Since the proposed approach and Brun's method in [11] differ in their basic methodology and objectives, a direct comparison of results using previously used metrics may not be possible. However, both methods apply scene segmentation and trajectory classification. Therefore, variations in sensitivity and fall-out rate were considered to be good measures for comparison. In Table 3, the summary comparative results using the MIT dataset is presented.

Following inferences can be drawn from the results presented in 3. Significantly less false positives have been recorded using the proposed method as compared to the baseline method of [11]. However, as β is increased beyond 1.0, the rate of change of true positives decreases as explained earlier. On the contrary, false positive cases are more in comparison to the different variations of Burn's method.

4.5 Effect of Key Parameters and Models

In this section, experimental results demonstrating the effect of some of the key parameters and models are reported.

4.5.1 Varying SVM Parameters

The data fed to the classifiers are linearly separable since the problem has been formulated in such a way that, any outlier to the consensus movement (depending on the value of β) formed within the set of training trajectories

Table 3: Summary of comparisons carried out with Brun's method [11] and its variations applied on MIT dataset on a 10 part cross validation scheme.

Method and Variations	# of Zones or Regions	Sensitivity	Fall-Out
Brun's Method + Simple Threshold	55	0.80 0.85 0.90	0.3 0.31 0.34
Brun's Method + Vertex Degree	15	0.80 0.85 0.90	0.29 0.33 0.36
Brun's Method + Inverted Weight	15	0.80 0.85 0.90	0.32 0.42 0.43
Brun's Method + Cluster Based	35	0.80 0.85 0.90	0.20 0.25 0.37
Proposed Method ($\beta = 0.5$)	9	0.86	0.19
Proposed Method ($\beta = 1.0$)	9	0.80	0.15
Proposed Method ($\beta = 2.0$)	9	0.34	0.09

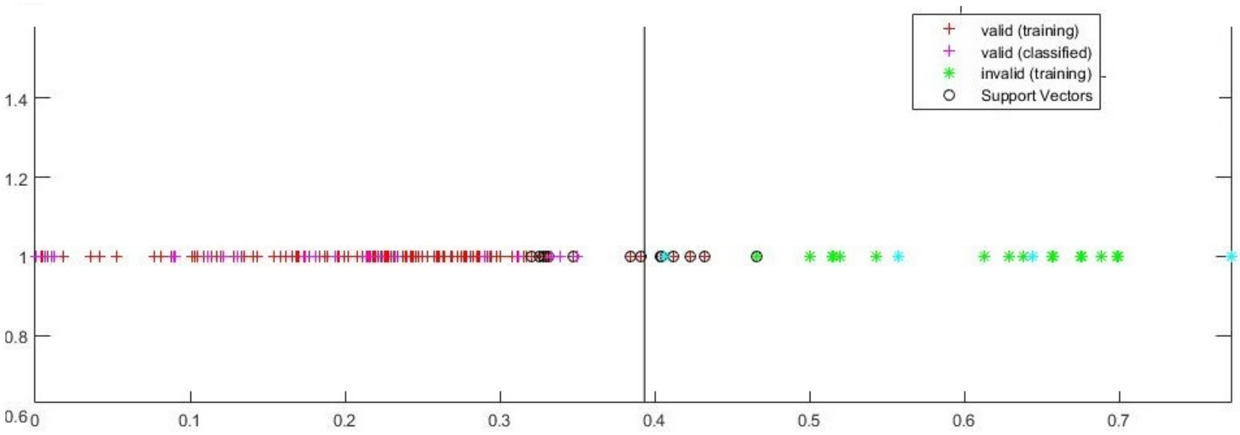

Fig. 7: Demonstration of linearly separable data used in the classification (MIT dataset with respect to region 1).

could be considered as the extreme values while estimating the test trajectory probabilities. In any case, a soft margin of 1 has been included in the classification stage to compensate for any errors accumulated from the previous steps and in order to make classification more generic. In Fig. 7, the evidence that the data used for classification is linearly separable, is presented.

4.5.2 Effect of Different Classifiers

In addition to experimenting with the SVM classifier, as an alternative mechanism of classification, the effect of the k-NN based classifier in conjunction with the proposed method has been studied. It can be observed from Table 4 that the k-NN based classifier is not a better choice compared to the SVM counterpart since it fails to detect more true cases. It may also be observed that, the value of k has very little effect on the rate of classification.

Table 4: SVM vs kNN Classifiers on MIT dataset (Type-I) analysis with $\beta = 1.0$.

Classifier	Abnormality @Node 1 GT: 21	Abnormality @Node 7 GT: 25
SVM	23	27
3-NN	16	20
4-NN	16	20
5-NN	16	19

4.5.3 Effect of β

The sensitivity of the proposed method can be influenced by the choice of the parameter β. Therefore, in order to study the effect of the parameter β on the proposed method, β is chosen to be varied between 0.5 to 3.0 while the true positive and false positive rates are measured. Fig. 8 presents the ROC curves obtained across the different datasets. It can be verified that, high true positive rates are possible in all three cases. However, when β is increased beyond 3.0, the rate of increase of false positives, is reduced. This could be because, when $\beta > 1.0$, the classifier is enforced to be biased towards negative samples as given in (16). Therefore, the rate of change of false positives becomes lower in comparison to the rate of change of true positives. On the contrary, when β is reduced below 1.0, the sensitivity of the proposed method is increased thus classifying a test trajectory as abnormal even under marginal deviation of the trajectory from its average statistics., thus, increasing the number of false positives.

In addition, the effect of β that is used to determine the class of a trajectory during preparation of data for training is also presented. It can be observed that, the number of invalid segments at a particular vertex reduces as β increases. This is in line with the assumption that increasing the value of β offers more flexibility to define a training trajectory as abnormal at a particular vertex. However, increasing the value of β could also introduce more false negatives which is usually not desired.

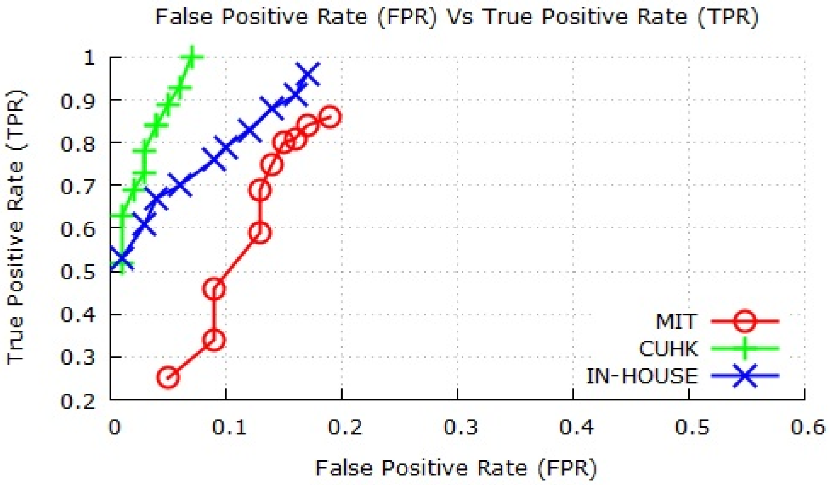

Fig. 8: ROC curves obtained by varying β between 0.5 and 2.5.

4.5.4 Computational Complexity

Given that the number of zones or regions can vary in the Brun's algorithm, the computational overhead is expected to increase as the number of vertices increased. Since the proposed method uses an optimized partitioning of the scene, testing a trajectory against the trained system is computationally acceptable and suitable for online analysis. However, the time required to train the classifiers in the proposed technique is high because of the complex nature of the feature extraction process. It has been observed that, the proposed method takes 31 seconds to compute the graph and train the classifiers on the MIT dataset. Testing takes on an average 1-2 seconds, which is often acceptable given that the training can be done offline.

5 Conclusion

In this paper, a novel approach to detect the presence of various types of anomalies in the motion of targets within a surveillance scene using a RAG, is proposed. It has been shown that, the features describing the intra-vertex and inter-vertex movement dynamics, estimated from a set of training trajectories, can be used to train multi-class SVM to categorize types of anomalies present in a surveillance scene. The proposed algorithm has been tested on two benchmark datasets and on a custom-built in-house dataset. Experimental results have validated the superiority of the proposed method against a chosen competing baseline method is classifying both Type-I and Type-II anomalies.

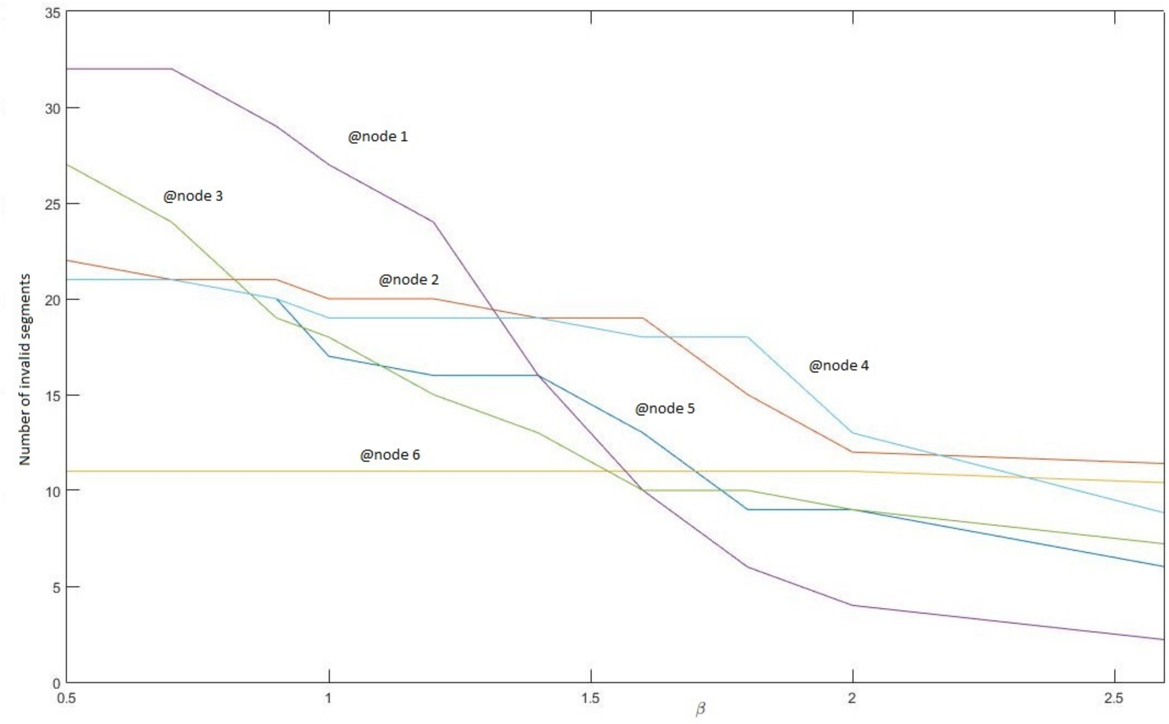

Fig. 9: Variation in number of invalid trajectory segments with respect to various nodes/regions of the RAG verses β value.

The future of this method can have potential applications in analysing crowd behaviour in outdoor as well as indoor environments. Further, a single camera based solution can be designed using the proposed methodology to assist security agencies responsible for maintaining smooth and safe operations inside crowded scene such as in railway stations, airports, busy road junctions, etc. Finally, possible extensions can be made for searching complex anomalies and be combined with target identification.

References

1. D. Gowsikhaa, S. Abirami, R. Baskaran, Automated human behavior analysis from surveillance videos: a survey, Artificial Intelligence Review 42 (4) (2014) 747–765.
2. M. Takai, Detection of suspicious activity and estimate of risk from human behavior shot by surveillance camera, in: Proceedings of the Second World Congress on Nature and Biologically Inspired Computing, 2010, pp. 298–304.

3. K. Ouivirach, S. Gharti, M. Dailey, Incremental behavior modeling and suspicious activity detection, Pattern Recognition 46 (3) (2013) 671 – 680.
4. J. Zhang, Z. Liu, Detecting abnormal motion of pedestrian in video, in: Proceedings of the International Conference on Information and Automation, 2008, pp. 81–85.
5. N. Kiryati, T. Raviv, Y. Ivanchenko, S. Rochel, Real-time abnormal motion detection in surveillance video, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–4.
6. W. Niu, J. Long, D. Han, Y. Wang, Human activity detection and recognition for video surveillance, in: Proceedings of the IEEE International Conference on Multimedia and Expo, Vol. 1, 2004, pp. 719–722.
7. B. Antic, B. Ommer, Video parsing for abnormality detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2415–2422.
8. R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, G. Coleman, Detection and explanation of anomalous activities: representing activities as bags of event n-grams, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 1031–1038.
9. T. Hospedales, J. Li, S. Gong, T. Xiang, Identifying rare and subtle behaviors: A weakly supervised joint topic model, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (12) (2011) 2451–2464.
10. M. Krishna, J. Denzler, A combination of generative and discriminative models for fast unsupervised activity recognition from traffic scene videos, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 640–645.
11. L. Brun, B. Cappellania, A. Saggese, M. Vento, Detection of anomalous driving behaviors by unsupervised learning of graphs, in: Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, 2014, pp. 405–410.
12. G. Tzanidou, I. Zafar, E. Edirisinghe, Carried object detection in videos using color information, IEEE Transactions on Information Forensics and Security 8 (10) (2013) 1620–1631.
13. K. Lin, S. Chen, C. Chen, D. Lin, Y. Hung, Abandoned object detection via temporal consistency modeling and back-tracing verification for visual surveillance, IEEE Transactions on Information Forensics and Security 10 (7) (2015) 1359–1370.
14. M. Elhamod, M. Levine, Automated real-time detection of potentially suspicious behavior in public transport areas, IEEE Transactions on Intelligent Transportation Systems 14 (2) (2013) 688–699.
15. L. Brun, A. Saggese, M. Vento, Dynamic scene understanding for behavior analysis based on string kernels, IEEE Transactions on Circuits and Systems for Video Technology 24 (10) (2014) 1669–1681.
16. T. Ko, A survey on behavior analysis in video surveillance for homeland security applications, in: Proceedings of the IEEE Applied Imagery Pattern

Recognition Workshop, 2008, pp. 1–8.

17. B. Morris, M. Trivedi, A survey of vision-based trajectory learning and analysis for surveillance, IEEE Transactions on Circuits and Systems for Video Technology 18 (8) (2008) 1114–1127.
18. S. Saleh, S. Suandi, H. Ibrahim, Recent survey on crowd density estimation and counting for visual surveillance, Engineering Applications of Artificial Intelligence 41 (2015) 103–114.
19. A. Yilmaz, O. Javed, M. Shah, Object tracking: A survey, ACM Computing Surveys 38 (4) (2006) 1–45.
20. A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: An experimental survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (7) (2014) 1442–1468.
21. J. Batista, P. Peixoto, C. Fernandes, M. Ribeiro, A dual-stage robust vehicle detection and tracking for real-time traffic monitoring, in: Proceedings of the Intelligent Transportation Systems Conference, 2006, pp. 528–535.
22. T. Dinh, N. Vo, G. Medioni, Context tracker: Exploring supporters and distracters in unconstrained environments, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2011, pp. 1177–1184.
23. Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, F. Gustafsson, Track-before-detect algorithm for tracking extended targets, IEE Proceedings on Radar, Sonar and Navigation 153 (4) (2006) 345–351.
24. M. Han, A. Sethi, W. Hua, Y. Gong, A detection-based multiple object tracking method, in: Proceedings of the IEEE International Conference on Image Processing, Vol. 5, 2004, pp. 3065–3068.
25. H. Veeraraghavan, P. Schrater, N. Papanikolopoulos, Robust target detection and tracking through integration of motion, color, and geometry, Computer Vision and Image Understanding 103 (2) (2006) 121–138.
26. S. Oron, Locally orderless tracking, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 1940–1947.
27. Z. Khan, I. Gu, Joint feature correspondences and appearance similarity for robust visual object tracking, IEEE Transactions on Information Forensics and Security 5 (3) (2010) 591–606.
28. G. Silveira, E. Malis, Real-time visual tracking under arbitrary illumination changes, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–6.
29. M. Ekman, Particle filters and data association for multi-target tracking, in: Proceedings of the 11th International Conference on Information Fusion, 2008, pp. 1–8.
30. S. Srkk, A. Vehtari, J. Lampinen, Rao-blackwellized particle filter for multiple target tracking, Information Fusion 8 (1) (2007) 2–15.
31. Z. Khan, T. Balch, F. Dellaert, A rao-blackwellized particle filter for eigentracking, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 2004, pp. 980–986.

32. J. Pan, B. Hu, J. Zhang, Robust and accurate object tracking under various types of occlusions, IEEE Transactions on Circuits and Systems for Video Technology 18 (2) (2008) 223–236.
33. P. Brasnett, L. Mihaylova, N. Canagarajah, D. Bull, Sequential monte carlo tracking by fusing multiple cues in video sequences, Image and Vision Computing, 25 (8) (2007) 1217–1227.
34. J. Cui, Y. Liu, Y. Xu, H. Zhao, H. Zha, Tracking generic human motion via fusion of low-and high-dimensional approaches. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43 (4) (2013) 996–1002.
35. Y. Liu, J. Cui, H. Zhao, H. Zha, Fusion of low-and high-dimensional approaches by trackers sampling for generic human motion tracking, in: Proceedings of the International Conference on Pattern Recognition, 2012, pp. 898–901.
36. V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 1975–1981.
37. B. Zhao, L. Fei-Fei, E. Xing, Online detection of unusual events in videos via dynamic sparse coding, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2011, pp. 3313–3320.
38. R. Leyva, V. Sanchez, C. Li, Video anomaly detection based on wake motion descriptors and perspective grids, in: Proceedings of the IEEE International Workshop on Information Forensics and Security, 2014, pp. 209–214.
39. Y. Cong, J. Yuan, Y. Tang, Video anomaly search in crowded scenes via spatio-temporal motion context, IEEE Transactions on Information Forensics and Security 8 (10) (2013) 1590–1599.
40. M. Krishna, P. Bodesheim, M. Krner, J. Denzler, Temporal video segmentation by event detection: A novelty detection approach, Pattern Recognition and Image Analysis 24 (2) (2014) 243–255.
41. F. Nater, H. Grabner, L. Gool, Temporal relations in videos for unsupervised activity analysis, in: Proceedings of the British Machine Vision Conference, 2011, pp. 21.1–21.11.
42. D. Kuettel, M. Breitenstein, L. Van Gool, V. Ferrari, What's going on? discovering spatio-temporal dependencies in dynamic scenes, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 1951–1958.
43. D. Dogra, R. Reddy, K. Subramanyam, A. Ahmed, H. Bhaskar, Scene representation and anomalous activity detection using weighted region association graph, in: Proceedings of the International Conference on Computer Vision Theory and Applications, 2015, pp. 17–25.
44. X. Wang, K. Ma, G. Ng, W. Grimson, Trajectory analysis and semantic region modeling using a nonparametric bayesian model, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

45. B. Zhou, X. Wang, X. Tang, Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 2871–2878.
46. R. Laxhammar, Chapter 4 - anomaly detection, in: V. N. Balasubramanian, S. Ho, V. Vovk (Eds.), Conformal Prediction for Reliable Machine Learning, Morgan Kaufmann, Boston, 2014, pp. 71–97.
47. R. Laxhammar, G. Falkman, Online learning and sequential anomaly detection in trajectories, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (6) (2014) 1158–1173.
48. K. Cheng, Y. Chen, W. Fang, Video anomaly detection and localization using hierarchical feature representation and gaussian process regression, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, pp. 2909–2917.
49. M. Roshtkhari, M. Levine, Online dominant and anomalous behavior detection in videos, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013, pp. 2611–2618.
50. Y. Liu, L. Nie, L. Han, L. Zhang, D. Rosenblum, Action2Activity: recognizing complex activities from sensor data, in: Proceedings of the International Conference on Artificial Intelligence, 2015, pp. 1617–1623.
51. Y. Liu, L. Nie, L. Liu, D. Rosenblum, From action to activity: Sensor-based activity recognition, Neurocomputing, 181 (2016) 108–115.
52. L. Liu, L. Cheng, Y. Liu, Y. Jia, D. Rosenblum, Recognizing Complex Activities by a Probabilistic Interval-Based Model, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016, pp. 1266–1272.
53. Y. Lu, Y. Wei, L. Liu, J. Zhong, L. Sun, Y. Liu, Towards unsupervised physical activity recognition using smartphone accelerometers, Multimedia Tools and Applications, 76 (8) (2017) 10701–10719.
54. Y. Liu, X. Zhang, J. Cui, C. Wu, H. Aghajan, H. Zha, Visual analysis of child-adult interactive behaviors in video sequences, in: Proceedings of the International Conference on Virtual Systems and Multimedia, 2010, pp. 26–33.
55. V. Saligrama, Z. Chen, Video anomaly detection based on local statistical aggregates, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 2112–2119.
56. N. Lia, X. Wua, D. Xud, H. Guoa, W. Fenga, Spatio-temporal context analysis within video volumes for anomalous-event detection and localization, Elsevier Neurocomputing 155 (1) (2015) 309319.
57. D. Dogra, A. Ahmed, H. Bhaskar, Interest area localization using trajectory analysis in surveillance scenes, in: Proceedings of the International Conference on Computer Vision Theory and Applications, 2015, pp. 31–38.
58. I. Sebanja, D. Megherbi, Automatic detection and recognition of traffic road signs for intelligent autonomous unmanned vehicles for urban surveillance and rescue, in: Proceedings of the IEEE International Conference on Technologies for Homeland Security, 2010, pp. 132–138.

59. X. Li, T. Breckon, Combining motion segmentation and feature based tracking for object classification and anomaly detection, in: Proceedings of the 4th European Conference on Visual Media Production, 2007, pp. 1–1.
60. B. Wenger, S. Mandayam, P. Violante, K. Drake, Detection of anomalous events in shipboard video using moving object segmentation and tracking, in: Proceedings of the IEEE AUTOTESTCON, 2010, pp. 1–6.
61. N. Nayak, Y. Zhu, A. Roy-Chowdhury, Exploiting spatio-temporal scene structure for wide-area activity analysis in unconstrained environments, IEEE Transactions on Information Forensics and Security 8 (10) (2013) 1610–1619.
62. N. Alon, R. Yuster, U. Zwick, Finding and counting given length cycles, Algorithmica 17 (1997) 209–223.
63. J. Flum, M. Grohe, The parameterized complexity of counting problems, SIAM Journal on Computing 33 (4) (2004) 892–922.
64. Y. Liu, Y. Liang, S. Liu, D. Rosenblum, Y. Zheng, Predicting urban water quality with ubiquitous data, in: arXiv preprint, 2016, 1610.09462.
65. Y. Liu, Y. Zheng, Y. Liang, S. Liu, D. Rosenblum, Urban water quality prediction based on multi-task multi-view learning, 2016.
66. Y. Liu, L. Zhang, L. Nie, Y. Yan, D. Rosenblum, Fortune Teller: Predicting Your Career Path, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016, pp. 201–207.
67. D. Preotiuc-Pietro, Y. Liu, D. Hopkins, L. Ungar, Beyond binary labels: political ideology prediction of Twitter users, in: Annual meeting of the association for computational linguistics, 2017.
68. S. Abe, Support vector machines for pattern classification, Vol. 53, Springer-Verlag, 2005.
69. Z. Jiang, Support vector machines for multi-class pattern recognition based on improved voting strategy, in: Proceedings of the Chinese Control and Decision Conference, 2010, pp. 517–520.