Efficient and effective automated surveillance agents using kernel tricks

Tarem Ahmed, Xianglin Wei, Supriyo Ahmed and Al-Sakib Khan Pathan

SIMULATION published online 7 November 2012
DOI: 10.1177/0037549712460908

Simulation

Simulation: Transactions of the Society for Modeling and Simulation International
0(0) 1–16

DOI: 10.1177/0037549712460908
sim.sagepub.com
SAGE SCS

Efficient and effective automated surveillance agents using kernel tricks

Tarem Ahmed[1,2], Xianglin Wei[3], Supriyo Ahmed[2] and Al-Sakib Khan Pathan[1]

Abstract

Many schemes have been presented over the years to develop automated visual surveillance systems. However, these schemes typically need custom equipment, or involve significant complexity and storage requirements. In this paper we present three software-based agents built using kernel machines to perform automated, real-time intruder detection in surveillance systems. Kernel machines provide a powerful data mining technique that may be used for pattern matching in the presence of complex data. They work by first mapping the raw input data onto a (often much) higher-dimensional feature space, and then clustering in the feature space instead. The reasoning is that mapping onto the (higher-dimensional) feature space enables the comparison of additional, higher-order correlations in determining patterns between the raw data points. The agents proposed here have been built using algorithms that are adaptive, portable, do not require any expensive or sophisticated components, and are lightweight and efficient having run times of the order of hundredths of a second. Through application to real image streams from a simple, run-of-the-mill closed-circuit television surveillance system, and direct quantitative performance comparison with some existing schemes, we show that it is possible to easily obtain high detection accuracy with low computational and storage complexities.

Keywords

agent, automated, camera, intrusion, kernel, surveillance

1. Introduction

Physical security is unfortunately of prime concern in today's world, and an extensive network of multimodal surveillance and security networks is prevalent in many places. They range from analogue closed-circuit television (CCTV) systems to sophisticated networks of infrared and motion sensors in sensitive areas such as banks and museums. The London Underground and London Heathrow Airport have more than 5000 cameras for the purposes of physical security, for example.[1] The task of simultaneously monitoring multiple camera images becomes tedious and monotonous for a human operator, and this consequently increases the risk of suspicious activity going unnoticed. Indeed, studies have shown that the optimal concentration span for a human being ranges between 25 and 30 minutes.[2] In the most unfortunate of situations, given the labour cost involved with hiring the requisite number of human operators to monitor a visual surveillance network, the feeds may be monitored only sparingly or not at all, often merely serving as an archive to retroactively refer back to once an untoward incident is known to have occurred.[3]

The field of automated visual surveillance has recently attracted a lot of interest, with many large research projects such as a European Erasmus Mundus project[4] granted on the subject. Beginning with the classic agent-based surveillance system proposed by Remagnino et al.,[5] a variety of software-based agents have been proposed recently for so-called third-generation automated surveillance systems.[6]. Monari et al. proposed an agent-based software architecture which detects and tracks moving objects across multiple camera views using a decentralized and collaborative sensor network, applying image processing algorithms for event and object detection at node level.[7]

[1]Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[2]Department of Electrical and Electronic Engineering, BRAC University, Dhaka, Bangladesh
[3]Department of Computer Science and Engineering, PLA University of Science and Technology, Nanjing, China

Corresponding author:
Al-Sakib Khan Pathan, Department of Computer Science, International Islamic University Malaysia (IIUM), Jalan Gombak, Kuala Lumpur 53100, Malaysia.
Email: sakib.pathan@gmail.com

Chao and Jun presented a multi-agent distributed surveillance system with each surveillance terminal acting as an agent that processes raw information from the sensors it governs, and transmits condensed results through an IP network.[8] Aguilar-Ponce et al. developed a scheme where cooperative agents perform object detection and tracking using clusters of wireless visual sensors.[9] The goals of current research are to develop algorithms that attract the attention of a human operator in real-time based on end-user requirements, process information arriving from a multi-sensor environment at high rates, and use low-cost standard components.[1,6] These requirements call for a low computational cost algorithm, which is fully automatic after possibly a minimal setup, and which is self-adaptable to a changing environment.[10]

The literature survey has revealed the following gaps in existing knowledge, as we discuss in detail later in Section 2. First, most existing algorithms involve high complexities, and need significant memory and storage resources. Second, no quantitative performance analysis/comparison is provided by most researchers. Third, most commercial systems require specific-purpose hardware and sophisticated equipment. In this paper, we present three software-based agents to perform automated detection of unnatural activity in visual surveillance systems. The proposed agents are built upon algorithms which involve computational, storage and memory complexities that are independent of time, making the agents naturally suited for online use. In addition, the proposed agents have been shown to work with the simplest, run-of-the-mill surveillance systems that may already be deployed. We have collected real data from such a system, and simulated using Matlab™. The proposed agents have been compared with two representative schemes selected from two families of methods currently being used in automated surveillance. Direct quantitative comparisons of detection performances between the proposed agents and the benchmark systems have been performed. It has been shown that the proposed agents achieve a higher detection rate for a given false-alarm tolerance level compared with the benchmark schemes, in addition to meeting the timing constraints enforced by a real-time applications.

The proposed agents are built using kernel machines[11] that employ the 'kernel trick'.[11] Kernel-based agents are built using algorithms that work by first mapping input data onto a (often much) higher-dimensional feature space, and then operating in the feature space instead. The reasoning is that mapping onto the (higher-dimensional) feature space enables the comparison of additional, higher order correlations in determining patterns between the raw data points.[11] We first present the kernel-based online anomaly detection (KOAD) algorithm.[12] KOAD is a learning algorithm that sequentially constructs and adapts a dictionary of features that approximately spans the subspace of normal behaviour.[13] The fundamental idea is if the multi-dimensional space is overdetermined with low intrinsic dimensionality of network traffic, regions occupied by the traffic features may be represented using a relatively small dictionary of approximately linearly independent feature vectors.[13] In addition, the size of the dictionary will be much smaller than the number of traffic measurements, thus leading to dimensionality and complexity reduction. Each new observation is compared with this dictionary by evaluating a distance measure between the observation and the cluster described by the dictionary, in the feature space.[12] The agent signals an anomaly when this distance metric exceeds a threshold. KOAD was first successfully applied to the problem of automated surveillance by Ahmed et al.[14]

The KOAD agent requires a couple of critical parameters to be manually set. This issue was subsequently addressed with the advent of the Kernel Estimation-based Anomaly Detection (KEAD) algorithm of Ahmed,[15] which incorporated autonomous setting for all important algorithm parameters. KEAD is based on the technique of kernel density estimators,[16,17] which are a popular method of obtaining estimates of minimum volume sets,[18–20] and of estimating the probability density function (PDF) of a random variable from a given sample. KEAD is also a recursive, learning algorithm that signals anomalies in real-time.

The Kernel Principal Component Analysis (KPCA) algorithm[21] is the kernel version of the standard principal components analysis (PCA) technique.[22] While KPCA is inherently a block-based algorithm intended for offline use, we modify it here into a sliding window implementation to propose an online agent that makes instantaneous decisions.

While the KOAD algorithm was previously applied to the problem of autonomous intruder detection in surveillance networks in Ahmed et al.,[14] this paper extends the work from Ahmed et al.[14] on a number of major aspects. The two new algorithms KEAD and KPCA, with various complementary advantages to KOAD, have been applied to the problem. A larger data set has also been used throughout. The performances of all three proposed agents KOAD, KEAD and KPCA have then been quantitatively compared with representative schemes from two existing families of systems that are commonly used for automated surveillance. The results obtained using the three kernel-based agents have been compared with those obtained using the standard PCA technique, and with a scheme using a data compression-based similarity metric between images that was proposed by Au et al.[23] Many existing methods of autonomous intruder detection in surveillance systems are based on the standard PCA technique and on Au et al.'s compression-based similarity metric, as is discussed later. The earlier work from Ahmed et al.[14] had just applied the KOAD agent, with no direct quantitative

performance comparison with any existing means of automated surveillance. In this paper we also provide a thorough analysis of the sensitivities of the various parameters involved, and show how to set the crucial standard deviation parameter (the so-called bandwidth) when a Gaussian kernel is used. These features were absent not only in Ahmed et al.[14] where the KOAD agent was first applied to the problem of automated surveillance, but also in Ahmed et al.[12] and Ahmed[15] where the original KOAD and KEAD algorithms, respectively, were first presented.

1.1 Outline of the paper

The rest of this paper is organized as follows. Section 2 describes some of the related work in automated visual surveillance schemes. Section 3 introduces the benchmarks that we compare our proposed methods against, and the mathematical techniques that we base the development of our proposed agents on. Section 4 presents the three kernel-based agents that we are advocating in this paper. Section 5 describes our experimental setup and Section 6 presents our experimental results with analyses. Section 7 concludes and outlines the future potential of our work.

2. Related work

2.1 Works based on software agents

A variety of software-based agents have been recently proposed for automated surveillance. Monari et al. presented the concept of an agent-based software architecture for automated, wide-area video surveillance.[7] The authors use a decentralized, collaborative sensor network with the objective of detecting and tracking moving objects across multiple camera views. Image processing algorithms are used to detect events and objects at the sensor node level. If physical motion is detected, an agent-based, multi-sensor processing cluster is created. Each instantiated cluster is then responsible for observing one object in the scene, and dynamic sensor clusters autonomously manage object handover.

Chao and Jun presented a multi-agent, distributed, video surveillance architecture.[8] Each terminal acts as an agent that locally processes the raw information from the sensors it governs, and only condensed results are transmitted through an IP network to reduce unnecessary load. The use of an active 'blackboard' is advocated as the data exchange center among information processing and control units, to ensure that all control information can be transmitted and exchanged when needed.

Aguilar-Ponce at al. developed a scheme where cooperative agents perform object detection and tracking using clusters of wireless visual sensors scattered in an area.[9,24] The proposed architecture consists of several object processing units that are wirelessly connected in a cluster fashion. The cluster heads analyse all of the information and communicate with the scene processing units. The area under surveillance is divided into several sub-areas with one camera is assigned to each sub-area. A region agent monitors each given sub-area. The camera in the relevant sub-area first performs background subtraction and a computed foreground mask is passed to the region agent, which then creates object agents responsible for tracking the detected objects. Object processing units automatically perform the object detection and tracking. The tracking information and foreground mask are sent to a scene processing unit that finally analyses this information and determines if a threat pattern is present at the scene and an alarm needs to be raised.

2.2 Works based on machine learning and PCA

There has recently been a lot of interest in applying machine learning principles to automated visual surveillance. However, most schemes that we have come across either involve significant computational complexities or significant memory and storage resources, thereby limiting their utility as an online solution for a high data rate application. It must be noted that the computational, storage and memory complexities of KOAD and KEAD are independent of time, making the agents naturally suited for online use. With our sliding window implementation, the complexity of KPCA is also bounded. Examples of algorithms for automated intruder detection in surveillance systems based on machine learning principles include the technique of Sudo et al., where the authors represent arriving images from a single video camera from spatio-temporal features without the use of any heuristics. They begin by extracting the moving areas in each frame, and create a background model by analysing the time sequences of each pixel as a mixture distribution. The sequence is divided into sets of a constant number of frames to yield the feature sets. The technique of PCA is then used to compress the dimensionality of the feature space, and the result is fed into a one-class Support Vector Machine (SVM). The discrimination function of the one-class SVM (OCSVM) is used to identify anomalies.

Singh et al.[25] have presented an automated system for object detection, movement tracking, and activity monitoring across frames in video streams, using background modelling. They present a system that both detects a human object in a frame, and also segments the object so that it can be tracked in subsequent frames. Segmentation of objects from the background is performed using a Gaussian mixture model. The tracking algorithm considers the human form as a whole across frames, instead of individually tracking the human parts such as limbs. Object features such as center of mass, size, and bounding box are used to estimate a match in consecutive frames. As the object is segmented and tracked, a Bayesian inference framework is used for event analysis. The event

recognition algorithm assumes that the shape of each type of object is known. Our proposed approach in this paper is contrary to this signature-based approach,[26] in that no prior information regarding the unusual activity or alien objects to be detected, is assumed. The experiments reported by Singh et al. were conducted using a single camera view, and unusual activity was detected using the detected objects and object tracking results. Singh et al. only provide performance comparisons with very rudimentary methods such as Gaussian mixture models (GMM) and temporal differencing, concepts on which they base their very proposed method on. No performance comparison with other competing algorithms is provided.

Various approaches where moving objects are detected in video sequences directly using PCA, have also been proposed in literature, with Verbeke and Vincent[27] being among the most popular. They isolate the most significant information from the images and express it in a lower-dimensional space, in which classifying motion areas and still areas becomes easier. To map to this lower-dimensional space, they apply PCA on the input data, and only keep the first two principal components. They compute the false-negative and false-positive rates of their proposed algorithm, and the values reported show effective performance.

Wang et al.[28] proposed a two-stage process involving incremental two-dimensional PCA (2DPCA) and maximum-likelihood estimation for tracking foreground moving objects in a dynamic background. The aim of the first stage is object characterization through dynamic learning. The aim of the second stage is object tracking, assuming that the foreground object regions are available together with previous learned objects. For target object characterization, incremental 2DPCA is used to characterize the image regions containing the target objects. Assuming separable kernels along rows and columns, recursive formulae are developed for an incremental 2DPCA algorithm. In this paper we have proposed Kernel PCA, and two other kernel-based agents (KOAD and KEAD), and compared performance with the fundamental PCA-based scheme.

2.3 Works based on information theory

Au et al.[23] presented a scheme where novel images are stored, and future images are compared against it using a similarity metric known as the normalized compression distance (NCD) which is based on mutual information. They achieve sparsity by only storing the novel images. Our proposed KOAD and KEAD agents achieve sparsity by using the dictionary of images that approximately spans the space of normality. Comparison with Au et al.'s scheme shows that KOAD and KEAD achieve much greater degrees of sparsity, as indicated by a much smaller dictionary compared with the number of images that Au et al.'s scheme needs to store. Au et al. ran their experiments on an empty office corridor, during a weekend. Thus their proposed system was applied to a lightly-loaded scenario, which matches the present focus of our work. The NCD measure that Au et al. introduced has led to substantial further work in outlier detection from image sequences,[26,29] as is described later in Section 3.2.

Nowak and Jurie[30] proposed a system that learns a similarity measure for previously unseen objects. They developed an activity-based semantic scene model for an area that is viewed by a video surveillance system. Semantics of their model include entry/exit zones, paths, routes and stop zones. A set of methods are presented that allow automatic learning of the scene elements from observations. The first steps in motion tracking require the separation of objects of interest from the background. A sequence of video frames is used to define an adaptive pixel-wise model for the background based on a Gaussian mixture model for each pixel, in intensity, RGB pixel values or normalized RGB space. Pixels are classified at each frame as either foreground or background according to the most likely Gaussian model. A connectivity algorithm is then applied to identify possible objects in motion. In surveillance applications, activity analysis is based on a manual segmentation of the scene, so special configuration is needed is each surveillance system to allow event analysis. The unsupervised nature of the proposed algorithms allows the implementation of a visual surveillance system that 'observes' and 'learns' its environment by an activity-based semantic scene model that consists of primitive elements. Their system learns the measure from pairs of training images labelled either 'same' or 'different', and learns the characteristic differences between local descriptors sampled from pairs of same and different images.

2.4 On-going works

The field of automated visual surveillance has recently attracted a lot of interest, with many large research projects such as a European Erasmus Mundus project[4] granted on the subject. Some of the most recent research on outlier detection in image sequences is being carried out at the Computer Vision Laboratory at the Swiss Federal Institute of Technology (ETH) Zurich in Zurich, Switzerland,[31,32] and at the Center for Sensor Web Technologies at Dublin City University in Dublin, Ireland.[33,34] Gowsikhaa et al. have added the supplemental capabilities of face, hand and gesture recognition, to determine suspicious specific actions such as looking over a neighbour's script during an examination.[35] Most of these researchers, however, only present their own algorithms, without providing any quantitative performance comparisons on labelled data.

Breitenstein et al. present a data-driven, unsupervised approach for unusual scene detection.[31] Their method is not object-class specific, but is based on simple, static features. It is aimed at detecting atypical configurations

within a scene. Their method is applicable for video systems with low frame rates, which is typical with cameras used in surveillance networks. The target is to automatically learn the usual scenario for a camera's field of vision, and then detect unusual events, while employing incremental learning techniques to adapt to changes in the data stream itself. The algorithm uses the most representative subset of a hierarchical feature set, defining usual scenes based on the concept of meaningful nearest neighbours[36] instead of building explicit models.

The follow-up work by Schuster et al.[32] presents another learning method for real-time, automatic identification of unusual events in video streams. This work performs the additional task of also identifying potential regions within the identified frame, which is displaying the unusual behaviour. Instead of explicitly modelling specific unusual events, the proposed approach incrementally learns the usual scenarios from the data source, and simultaneously points out potential unusual regions within the image that has been flagged as an outlier. Feature extraction from the images is done using histograms of oriented gradients (HoG).[37] Schuster et al.[32] extends the ideas regarding abnormality detection using meaningful nearest neighbour from Breitenstein et al.[31] A purely data driven approach is used akin to Breitenstein et al.,[31] and a model of usualness is built by storing representative clusters of observed data. A purely data-driven approach is portable, and works in different applications without human intervention. In addition, it adapts to changes, the so-called data drift, automatically. This portability across applications and adaptive behaviour to changes in the nature of normality itself, are characteristics that are also satisfied by the agents proposed in this paper.

It must be mentioned here, however, that neither Breitenstein et al.[31] nor Schuster et al.[32] carry out any marking of their data sets a priori in order to establish or define any sort of 'ground truth'. Furthermore, only the true positive instances are mentioned in both works, with no evaluation of the true positive accuracy rate (i.e. missed detections or false negatives), and no discussion of the false positives. In contrast, such analysis is thoroughly performed for all the agents proposed in this paper.

Kuklyte et al. proposed a general purpose framework for detection of unusual events in video streams,[33] that is based on the unsupervised method for unusual scene detection from webcam images introduced by Breitenstein et al.[31] Kuklyte et al.[33] extended the work of Breitenstein et al.[31] on two primary aspects. First, they generalized the algorithm of Breitenstein et al.[31] from being applicable solely to discrete images or frames taken every few seconds, to being applicable to continuous data streams, and from a variety of sensor types. This is achieved by using so-called time–space block feature vectors, and performing unsupervised classification using Euclidean similarity measure. Second, they integrated data-fusion methodologies into the abnormal event detection framework of Breitenstein et al.[31] This improves reliability and enables the detection of events that single modality data analysis alone is not able to provide. Kuklyte et al. conducted experiments using an audio-visual camera placed in the corridor outside their research laboratory, and the images presented look similar to the images obtained from our real-life outdoor setting. Kuklyte et al.[34] also investigated the parameters involved with the system presented in their earlier work.[33] They claimed in later work[34] that their proposed system can be easily deployed and automatically adapt to any environment without any manual adjustment. This condition is also met by the agents proposed in this paper, and is experimentally demonstrated by application to a simple, already-deployed system. Kuklyte et al. have deliberately used descriptors of low computational complexity to enable the system to run in real-time and to enable implementation directly in the camera hardware. Our proposed agents also have low computational and memory complexities, and implementations over centralized as well as distributed architectures with low actual runtimes have been demonstrated experimentally.

Gowsikhaa et al.[35] have added the supplemental capabilities of face, hand and gesture recognition, to determine suspicious specific actions such as looking over a neighbour's script in an examination hall. Human faces and head motion are detected using artificial neural networks after performing background estimation and foreground extraction. A combination of motion detection, edge detection and skin color detection is used to identify the hands of students, and this helps to detect contact between students during an examination. The proposed system uses video processing schemes that are easy to install. The experimental results presented indicate high efficiency and detection rates in a lightly loaded environment.

2.5 Commercial systems

The popular commercial system DETER from Honeywell Inc.[38] has been built using machine learning principles, based on the techniques described by Morellas et al.[39] Most commercial applications tend to use specific-purpose hardware and require a network of sophisticated equipment. There are quite expensive, often need professional help for installation and maintenance and they are optimized to perform specific tasks in specific environments.[33] Examples are ObjectVideo,[40] NICE Systems[41] and Ipsotek.[42] These systems also use expensive internal software, the algorithmic bases for which are not publicly available. In contrast, our proposed agents work with the cheapest possible surveillance system, as demonstrated on data from an example already-deployed, run-of-the-mill system.

2.6 Our contributions

Most schemes that we have come across involve high complexities, and need significant memory and storage resources, thereby limiting their utility as an online solution for a high data rate application. Moreover, no quantitative performance analysis/comparison is provided by most researchers. We have seen that most commercial systems require specific-purpose hardware and customized sophisticated equipment, and require expensive professional help for installation and maintenance.

In this paper, we propose agents for which the computational, storage and memory complexities are independent of time, making them naturally suited for online use. In addition, the proposed schemes are shown to work with an example of a simple, run-of-the-mill surveillance system that may already be deployed. Furthermore, direct quantitative comparisons of detection performances between the proposed agents and selected benchmark schemes are provided by means of receiver operating characteristic (ROC) curves.[43]

3. Benchmarks and mathematical preliminaries

3.1 PCA

The technique of PCA may be used to separate the space occupied by set of input vectors into two disjoint subspaces, corresponding to normal and anomalous behaviour.[44] An anomaly may then be flagged in the timesteps where the magnitude of the projection onto the residual anomalous subspace, θ_t, exceeds a threshold.

We decided to compare our proposed agents with PCA for two primary reasons. First, PCA has been the basis for many recent autonomous intruder detection systems, as was reviewed in Section 2.[27,28,45] Second, while kernel machines cluster the points in the feature space mapped onto by the chosen kernel function, PCA clusters directly in the input space itself, thereby providing a complementary approach.

3.2 Compression-based similarity metric

Au et al.[23] have presented a scheme where a set of novel images are stored, and arriving images are compared with this set. A scene is considered to be anomalous when the maximum similarity between the given scene and all previously viewed scenes evaluates to below a given threshold. Similarity is measured using the NCD measure that was proposed by Li et al.[46] The algorithm of Au et al. first compresses each image individually, and then compresses concatenated pairs of images that are being directly compared. The similarity metric, ρ, is then evaluated comparing the compressed file size of the concatenated pair of images, and the compressed file sizes of the two images individually. The premise is that the size of the compressed version of the concatenation of two similar files should be smaller than that of the concatenation of two dissimilar files. Au et al. ran their experiments on an empty office corridor, during a weekend. Thus, their scheme was applied to a lightly loaded scenario. This experimental setting matches our present focus.

The NCD measure has been shown to be a versatile and broadly applicable tool for pattern analysis, and problem formulations based on it can be very general, parameter-free, robust to noise, and portable across applications and data formats.[47] Cebrian et al.[48] have also shown that the NCD measure is robust to noise. Cilibrasi and Vitanyi[49] have demonstrated the effectiveness of the NCD measure across a diverse range of applications spanning the fields of genomics, virology, natural language processing, literature, music, handwritten digit recognition and astronomy. Cohen et al. have proposed another information-theoretic algorithm based on NCD to track meaningful changes in image sequences.[29] Yahalom has developed a novel algorithm for web server intrusion detection systems (IDSs) that does not rely on signatures of past attacks, using an NCD-based metric.[26]

3.3 Kernel methods

Agents based on the so-called 'kernel trick' involve using a kernel function that maps the input data onto a feature space of much higher dimension,[11] with the expectation that points depicting similar behaviour would form thicker clusters in the richer space. A suitable kernel function, when applied to a pair of input vectors, may be interpreted as an inner product in the feature space.[11] This subsequently allows inner products in the feature space (inner products of the feature vectors) to be computed without explicit knowledge of the feature vectors themselves, by simply evaluating the kernel function:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \quad (1)$$

where $\mathbf{x}_i$, $\mathbf{x}_j$ denote the input vectors and ϕ represents the mapping onto the feature space. Using kernel functions thus allows simple comparison of higher-order statistics between the input vectors.

3.4 Kernel density estimation

Also known as the Parzen Window method or the Parzen–Rosenblatt window,[16,17] kernel density estimation (KDE) is a popular nonparametric method of estimating the PDF of a random variable from a finite data sample. Given an independent, identically distributed sample set $\{\mathbf{x}_i\}_{i=1}^{n}$ drawn from an unknown probability distribution f, the value of PDF f at any point t may be estimated as

$$\hat{f}(t) = \frac{1}{n}\sum_{i=1}^{n} k(\mathbf{x}_i, \mathbf{x}_t) \tag{2}$$

where $\boldsymbol{k}(,)$ denotes the kernel function.

4. Proposed kernel-based agents

4.1 KOAD algorithm

If the points $\{\mathbf{x}_t\}_{t=1}^{T}$ show normal behaviour in the input space, then the corresponding feature vectors $\{\phi(\mathbf{x}_t)\}_{t=1}^{T}$ are expected to (also) cluster. Then it should be possible to explain the region of normality in the feature space using a relatively small dictionary of approximately linearly independent elements $\{\phi(\tilde{\mathbf{x}}_j)\}_{j=1}^{m}$. Feature vector $\phi(\mathbf{x}_t)$ is said to be approximately linearly dependent on $\{\phi(\tilde{\mathbf{x}}_j)\}_{j=1}^{m}$ with approximation threshold ν, if the projection error δ_t satisfies[13]

$$\delta_t = \min_{\mathbf{a}} \left\| \sum_{j=1}^{m} a_j \phi(\tilde{\mathbf{x}}_j) - \phi(\mathbf{x}_t) \right\|^2 < \nu. \tag{3}$$

where $\mathbf{a} = \{a_j\}_{j=1}^{m}$ is the optimal coefficient vector. Here $\{\tilde{\mathbf{x}}_j\}_{j=1}^{m}$ represent those $\{\mathbf{x}_t\}_{t=1}^{T}$ that are entered into the dictionary. The size of the dictionary, m, is expected to be much less than T, thereby leading to computational and storage savings.

Observe that (3) involves an L2 norm, which may be simplified exclusively in terms of the inner products of $\phi(\tilde{\mathbf{x}}_j)$ and $\phi(\mathbf{x}_t)$, and thus evaluated using the kernel function without explicit knowledge of the feature vectors themselves:

$$\delta_t = \min_{\mathbf{a}_t} \left\{ \mathbf{a}_t^{\mathrm{T}} \tilde{\mathbf{K}}_{t-1} \mathbf{a}_t - 2\mathbf{a}_t \tilde{\mathbf{k}}_{t-1}(\mathbf{x}_t) + k(\mathbf{x}_t, \mathbf{x}_t) \right\} \tag{4}$$

where $[\tilde{\mathbf{K}}_{t-1}]_{i,j} = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ and $[\tilde{\mathbf{k}}_{t-1}(\mathbf{x}_t)]_j = \boldsymbol{k}(\tilde{\mathbf{x}}_j, \mathbf{x}_t)$ for $i, j = 1 \ldots m_{t-1}$. The optimum sparsification coefficient vector $\mathbf{a}_t$ that minimizes δ_t at time t is then

$$\mathbf{a}_t = \tilde{\mathbf{K}}_{t-1}^{-1} \cdot \tilde{\mathbf{k}}_{t-1}(\mathbf{x}_t). \tag{5}$$

The expression for error δ_t may then be simplified to

$$\delta_t = k_{tt} - \tilde{\mathbf{k}}_{t-1}(\mathbf{x}_t)^{\mathrm{T}} \cdot \mathbf{a}_t. \tag{6}$$

The KOAD agent operates at each timestep t on a measurement vector $\mathbf{x}_t$. It begins by evaluating the error δ_t in projecting the arriving $\mathbf{x}_t$ onto the current dictionary (in the feature domain). This error measure δ_t is then compared with two thresholds ν_1 and ν_2, where $\nu_1 < \nu_2$. If $\delta_t \leq \nu_1$, KOAD infers that $\mathbf{x}_t$ is sufficiently linearly dependent on the dictionary, and represents normal behaviour. If $\delta_t > \nu_2$, it concludes that $\mathbf{x}_t$ is far away from the realm of normality and raises a 'Red1' alarm to immediately signal an anomaly.

If $\nu_1 < \delta_t \leq \nu_2$, KOAD infers that $\mathbf{x}_t$ is sufficiently linearly independent from the dictionary to be considered an unusual event. It may indeed be an anomaly, or it may represent an expansion or migration of the space of normality itself. In this case, KOAD does the following: it raises an 'Orange' alarm, keeps track of the contribution of the relevant input vector $\mathbf{x}_t$ in explaining subsequent arrivals for a further ℓ timesteps, and then takes a firm decision on it. Further details regarding KOAD are available in Ahmed et al.[12]

4.2 KEAD algorithm

The KEAD agent formally states the problem as follows.[15] Given a sequence of multidimensional data points $\{\mathbf{x}_i\}_{i=t-L}^{t+L} \in \mathbb{R}^D$, the objective is to determine whether $\mathbf{x}_t$ is a realization of probability distribution $P_{n,t}$ or of probability distribution P_a. It is assumed that the points $\{\mathbf{x}_i\}_{i=t-L}^{t+L} \in \mathbb{R}^D$ are independent observations from the mixture distribution P_t:

$$P_t = (1-\pi)P_{n,t} + \pi P_a \tag{7}$$

where π is the mixing faction. The component distributions $P_{n,t}$ and P_a correspond to normal and anomalous traffic at time t, respectively. Distribution $P_{n,t}$ is assumed to be slowly time-varying, while P_a is time-invariant.

Assuming that the distribution governing the normal points, $P_{n,t}$, is stationary in the interval $\{t-L : t+L\}$, leads to the following expression for the KDE at $\mathbf{x}_t$:

$$\tau_t = \frac{1}{2L+1}\sum_{i=t-L}^{t+L} k(\mathbf{x}_i, \mathbf{x}_t) \tag{8}$$

where $\boldsymbol{k}(,)$ represents the kernel function. Now KDE τ_t is expected to be relatively low if $\mathbf{x}_t$ arises from the distribution governing the anomalous points, P_a, compared with the case when $\mathbf{x}_t$ arises from the distribution governing the normal points, $P_{n,t}$. KDE τ_t may thus be selected as a statistic for making a block-based (offline) anomaly decision on $\mathbf{x}_t$.

One may then use the dictionary $\mathcal{D}_{t-1}$ and the matrix $\mathbf{A}_t$ of optimal sparsification coefficient vectors $\mathbf{a_t}$ for the past L timesteps to obtain the online detection statistic $\hat{\tau}_t$:[15]

$$\hat{\tau}_t = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{j=1}^{m_{t-1}} \mathbf{a}_{\ell j} \cdot k(\tilde{\mathbf{x}}_j, \mathbf{x}_t) = \frac{1}{L}\sum_{i=1}^{L} \mathbf{A}_{t-1} \cdot \tilde{\mathbf{k}}_{t-1}(\mathbf{x}_t). \tag{9}$$

KEAD proceeds at every timestep t by first computing the optimum sparsification coefficient vector $\mathbf{a}_t$ from (5) and the projection error δ_t from (6), and the online detection statistic $\hat{\tau}_t$ from (9). The projection error δ_t may also be inferred as a sparsification statistic in this case. The sparsification statistic δ_t is then compared with a sparsification parameter ν. If $\delta_t \geq \nu$, $\mathbf{x}_t$ is inferred to be approximately linearly independent of the space spanned by the

dictionary at time t. Input vector $\mathbf{x}_t$ is consequently added to the dictionary. In contrast, if $\delta_t < \nu$, $\mathbf{x}_t$ is inferred to be approximately linearly dependent on the dictionary. The dictionary is then kept unchanged.

To make a decision regarding whether $\mathbf{x}_t$ is normal or anomalous, KEAD compares online detection statistic $\hat{\tau}_t$ with detection threshold η_t. If $\hat{\tau}_t \geq \eta_t$, the KDE of P_t at $\mathbf{x}_t$, that was computed using the dictionary and window of past L sparsification coefficient vectors, is high enough, and $\mathbf{x}_t$ is inferred to represent normal traffic. In contrast, if $\hat{\tau}_t < \eta_t$, the KDE of P_t is low. In this case, $\mathbf{x}_t$ either represents an anomaly, or $\{\mathbf{x}_i\}_{i=t-L}^{t}$ is not a sufficiently representative sample of $\{\mathbf{x}_i\}_{i=t-L}^{t+L}$ for the online detection statistic $\hat{\tau}_t$ to be an accurate estimate of true τ_t. The following is done in such a situation: an 'Orange' alarm is raised at time t, $\mathbf{x}_t$ is stored for the next $\ell < L$ timesteps, and then a firm decision is taken on it.

An orange alarm that was raised at time t is resolved at the end of timestep $t+\ell$ in the following manner. Online detection statistic $\hat{\tau}$ is re-computed using $\mathbf{A}_{t+\ell}$ and the kernel values of the $\mathbf{x}_t$ that had initially caused the 'Orange' alarm, with dictionary $\mathcal{D}_{t+\ell}$. The lag ℓ allows the window of sparsification coefficient vectors $\mathbf{A}$ to slide forward $\ell+1$ steps, while the dictionary is also allowed $\ell+1$ further modifications. By delaying the final decision by $\ell < L$ timesteps, the algorithm allows for the decision to be based on data sequence $\{\mathbf{x}_i\}_{i=t-L+\ell}^{t+\ell}$ instead of on sequence $\{\mathbf{x}_i\}_{i=t-L}^{t}$. Further details regarding KEAD are available in Ahmed.[15]

4.3 KPCA algorithm

KPCA is an extension of standard PCA that allows nonlinear regression.[21] Instead of determining the principal components of the input data vectors themselves as in standard PCA, KPCA determines the principal components in the feature space mapped onto by the chosen kernel function. The KPCA agent is developed as follows. The standard PCA technique is first formulated in a form that involves only inner products, as is possible to do. This allows the substitution with kernel functions. The principal components are then found by solving the eigenvalue problem for the input vector kernel matrix, also known as the Gram matrix.[50] When applied to image sequences, KPCA takes into account higher-order correlations between pixels in different images.

As KPCA is inherently a block-based agent intended for offline use, we apply it to our real-time detection problem by implementing it in the following manner. A sliding window $\mathbf{X}_t$ consisting of C input vectors $\{\mathbf{x}_i\}_{i=t-C+1}^{t}$ is maintained. KPCA is then run on this block to obtain a vector $\{\theta_i\}_{i=t-C+1}^{t}$ of the magnitude of the projection onto the residual, anomalous subspace pertaining to timesteps $\{t-C+1:t\}$. The value of this residual magnitude pertaining to the current timestep, θ_t, is then compared with a detection threshold, and a decision is made immediately regarding the presence or absence of an anomaly. In the next instance, the sliding window $\mathbf{X}_t$ of stored input vectors moves forward one step, KPCA is run on the new data block, and the process is repeated.

It must be mentioned here that for online applications, early proponents of PCA-based methods of anomaly detection had suggested projecting arriving input vectors onto eigenvectors calculated from a previous block of data.[44] However, it has since been shown that PCA is extremely sensitive to calibration settings,[51,52] and we have found here that unstable results are obtained if a current data vectors are projected onto eigenvectors calculated from a previous block of timesteps.

5. Experimental setup

5.1 Data

We collected real footage from a set of four cameras from the CCTV surveillance system deployed at BRAC University, Dhaka, Bangladesh. The raw data comprised of a video sequence in the AVI format. From the videos, we first extracted still images in JPEG format at two-second intervals. The total data set consisted of 500 timesteps, of which 62 were identified as potential anomalies after performing an exhaustive, manual inspection of the data set. We isolated the first 50 timesteps as the training period for our experiments.

Figure 1 shows pictures corresponding to six example timesteps. Figure 1(a) and (b) show regular (normal) scenarios; Figure 1(c) and (d) show obvious cases of human forms appearing on the scene (top-left and top-right cameras in Figure 1(c), top-right and bottom-right cameras in Figure 1(d)); Figure 1(e) presents a subtle case where a small portion of a person's foot is visible (top-right camera); Figure 1(f) presents another subtle case where alien lights appear (bottom-right camera). The actual timestamps have been removed from the images for the sake of privacy.

5.2 Monitoring architecture

We propose two monitoring architectures: a distributed topology and a centralized topology. In the distributed topology, the agents are run locally at each node. After each timestep, each individual node makes a decision on whether an intruder has been detected or not, and then communicates a binary result to the central monitoring unit (CMU). The CMU then draws the attention of the operator if at least p out of the n nodes has detected an anomaly in the same timestep. The idea behind this p-out-of-n detection scheme[53] is that bona fide intruders are likely to show up on multiple cameras, while isolated flags are more likely to result from false alarms such as camera malfunctions and atmospheric/weather elements affecting

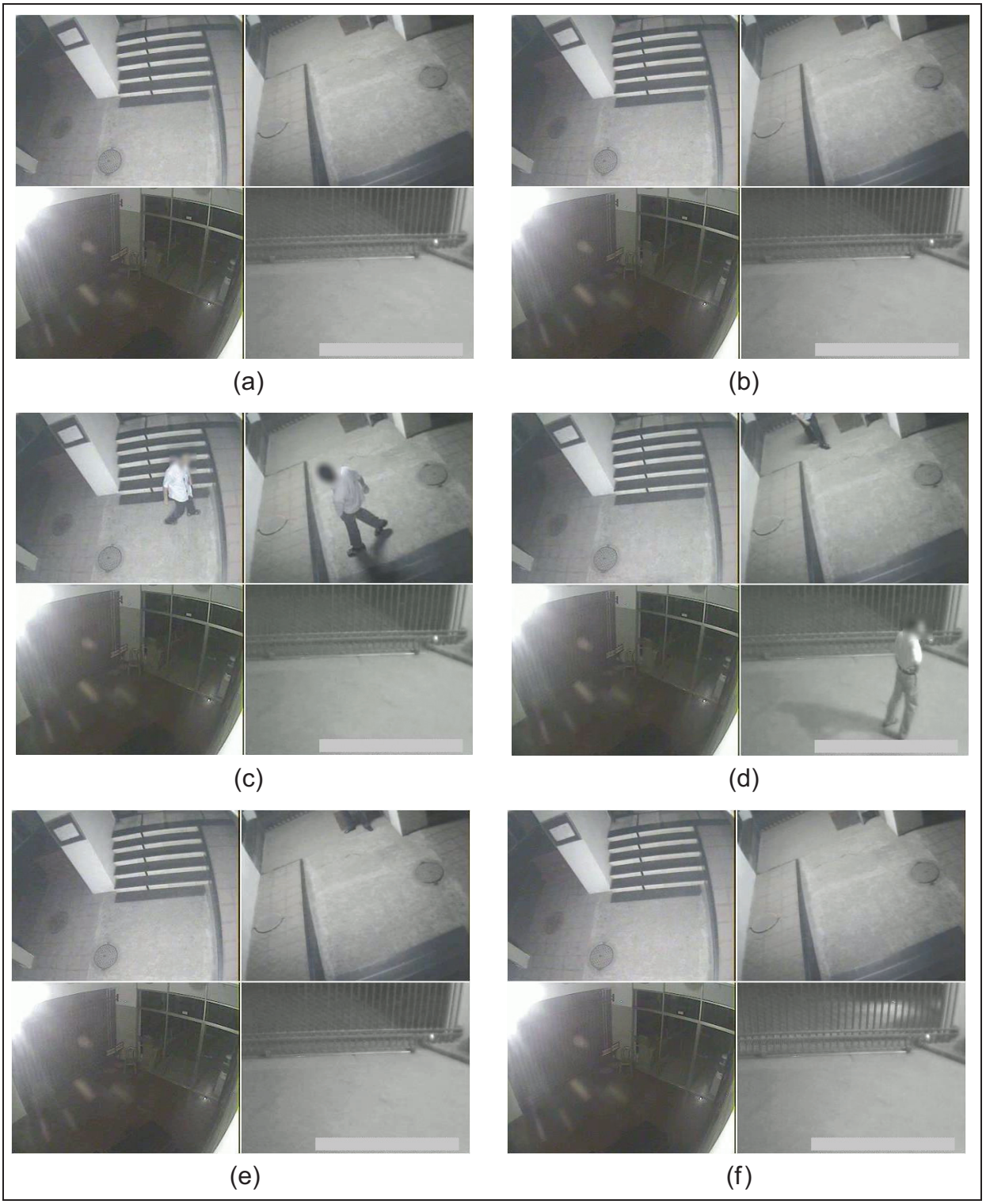

Figure 1. Set of images obtained from four cameras in the BRAC University CCTV surveillance system, corresponding to six different timesteps. Usual images are observed in two of the timesteps, two timesteps show situations where human forms are easily visible, and two show subtle cases where the foot of a person is visible in one and some alien light beams are observed in another. Actual time stamps on the images have been concealed because for privacy. (a) Normal image set at timestep $t = 200$. (b) Normal image set at timestep $t = 350$. (c) Pronounced anomaly at timestep $t = 106$. (d) Pronounced anomaly at timestep $t = 112$. (e) Subtle anomaly at timestep $t = 297$. (f) Subtle anomaly at timestep $t = 400$.

the image. The distributed architecture is intended for application in a large surveillance network of sensors where the computing power is distributed.

In the centralized topology, the entire image data must be transmitted to the CMU. The CMU then runs the detection agents at each timestep on the globally obtained (concatenated) image set. This is the more likely architecture for most common visual surveillance systems, where all images are transmitted to a control room.

5.3 Data preprocessing and feature extraction

After extracting JPEG images at two-second intervals from the AVI videos, we performed standard two-dimensional Haar wavelet decompositions to obtain a $120 \times 160 \times 3$ tensor representation for each image. Working in the frequency domain is preferable to working in the space domain in order to account for differences between specific pixels in different images arising as a result of minor camera movements, and also to consider and compare each image as a whole. Wavelets provide a convenient technique for representing image details in the frequency domain. The wavelet decomposition represents an image in a manner that reflects variation in neighbouring pixel intensities, and also performs image compression. Because this representation relates neighbouring pixel intensities, it is also suitable to be fed to agents which look to find patterns between higher-order statistics of the pixels.

We finally performed 10% bilinear interpolation to rescale and reduce the size of each dimension. The output of the four cameras was then concatenated to obtain one $12 \times 16 \times 3 \times 4 = 2304$-dimensional row vector of input data corresponding to each timestep.

5.4 The kernel choice

The postulate behind applying a kernel approach is that similar images are expected to cluster better in the higher-dimensional feature space, and mapping onto the feature space will enable the comparison of additional, higher-order correlations between the pixels. The Gaussian kernel defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\{-\frac{\| \mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}\}, \quad (10)$$

where σ is the standard deviation parameter, and the polynomial kernel of degree q defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) = (\alpha\langle \mathbf{x}_1 \mathbf{x}_2\rangle + \beta)^q \quad (11)$$

where α and β are constants, are conventional choices for image comparison applications. Note that a special case of the polynomial kernel with $\alpha = \beta = q = 1$, is the linear kernel:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \langle \mathbf{x}_1, \mathbf{x}_2\rangle. \quad (12)$$

We have observed that choosing the Gaussian kernel provides the best results. In order to magnify the distances between the clusters formed in the feature space, the standard deviation parameter for the Gaussian kernel (also known in this context as the kernel bandwidth), σ, should be chosen so as to produce the maximum variation in the kernel density estimates. We determined the value to set for σ during the training period, by cycling through a number of possible choices for σ, chosen at logarithmically spaced intervals between the minimum and maximum values of the component dimensions of the training data points, and then determining the value that produced the maximum variance in the kernel density estimates. The objective was to enhance the variation in the kernel density estimates between normal points and outliers. Figure 2 shows the variance in the kernel density estimates as a function of the kernel bandwidth, σ, during the training period.

6. Experimental results

6.1 Parameter selections

The detection performance of KOAD is primarily a function of the thresholds ν_1 and ν_2. Threshold ν_1 has the most direct effect on the detection performance, while threshold ν_2 determines the instant flagging of an anomaly. Our experiments have shown that while optimal settings for ν_1 and ν_2 vary between different applications, the performance of a setting remains approximately the same across widely separated time periods within the same application. Optimum values may be set after running the agent over a training set of labeled data with known anomalies. Our experiments have indicated that the detection performance is not particularly sensitive to the choice of the KOAD 'Orange' alarm resolution parameters, or the parameters

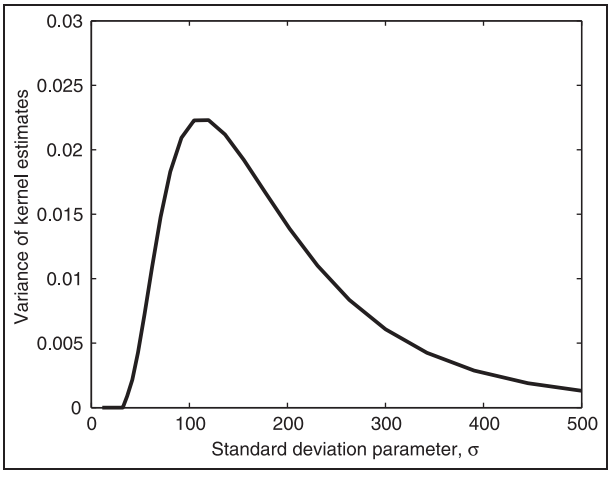

Figure 2. Variance of kernel estimates as a function of the standard deviation (bandwidth) of the Gaussian kernel function, σ.

governing the dropping of dictionary members. They may thus be suited to taste depending on how much of a time-lag is allowable for the 'Orange' alarm resolution, and the storage resources available to the system. The parameter sensitivities of KOAD when applied to image sequences have been discussed previously by Ahmed et al.,[14] so we concentrate on the other algorithms in this paper.

We ran KEAD with a range of values for its sparsification parameter ν and window length L, setting $\ell = 1$ to resolve orange alarms immediately. Figure 3 shows the variation in the average size of the dictionary as a result of varying the sparsification parameter ν. It is clear that no significant improvement is sparsity is obtained for $\nu > 0.20$. Thus, we use $\nu = 0.20$ as the default value in our experiments involving KEAD. Figure 4 shows the variation in the probability of detection (P_D) and the probability of false alarms (P_{FA}) with window length L, for different values of the KEAD detection threshold η. We have observed in our experiments that the detection and false-alarm rates usually stabilize after $L > 3$, over a range of values of η. We have set $L = 6$ as the default in our experiments involving KEAD.

The KEAD detection statistic $\hat{\tau}_t$ is an estimate of the true kernel estimate τ_t, with approximation errors introduced on two major counts: the use of the dictionary, and a past window of length L, instead of the entire data sequence $\{\mathbf{x}_i\}_{i=t-L}^{i=t+L}$.[15] We wish to investigate the effects of sparsification parameter ν (and, hence, the size of the dictionary) and the window length L, on the closeness of KEAD estimate $\hat{\tau}$ to the true value of τ. As our metric for the approximation error, we use the mean absolute relative error over the data set defined as

$$V = \frac{1}{T}\sum_{t=1}\left|\frac{\tau_t - \hat{\tau}_t}{\tau_t}\right| \tag{13}$$

where T is the length of the test data set. Figure 5 shows the variation in this error metric V with sparsification parameter ν (top panel) and window length L (bottom panel). The window length L is set to its default value of 6 in the experiment varying sparsification parameter ν, while ν is set to its default value of 0.20 in the experiment varying L. We see from Figure 5(top panel) that this metric representing the average, relative error increases as ν is increased and the dictionary size falls (Figure 3). This is expected, as a sparser dictionary corresponding to a higher ν will result in a worse approximation. Figure 5(bottom panel) shows that $\hat{\tau}_t$ becomes a better estimate of τ_t with increasing window length L. This is also expected, assuming that our data is drawn from a distribution that is stationary in small intervals, as sequence $\{\mathbf{x}_i\}_{i=t-L}^{i=t}$ will tend to a better representation of $\{\mathbf{x}_i\}_{i=t-L}^{i=t+L}$ with increasing L. Moreover, we also observe that the improvement in the error metric is very small in absolute terms, thereby justifying our assumption of short-term stationarity for this data set.

We ran KPCA with the sliding window implementation described in Section 4.3. We set the window size to 30 timesteps, used a Gaussian kernel function with the same value for the standard deviation parameter as used for KEAD, and assigned two principal components to the normal subspace.

As our first benchmark scheme, we ran standard PCA using the sliding window implementation and the same window size KPCA (30 timesteps) and with two principal components assigned to the normal subspace. As our second benchmark scheme, Au et al.'s NCD-based system was implemented from the pseudocode provided in their work,[54] with the authors' recommended value of 0.055 as the similarity threshold.

6.2 Detection performances

Figure 6 compares the performances of the various schemes studied in this paper, through ROC curves[43]

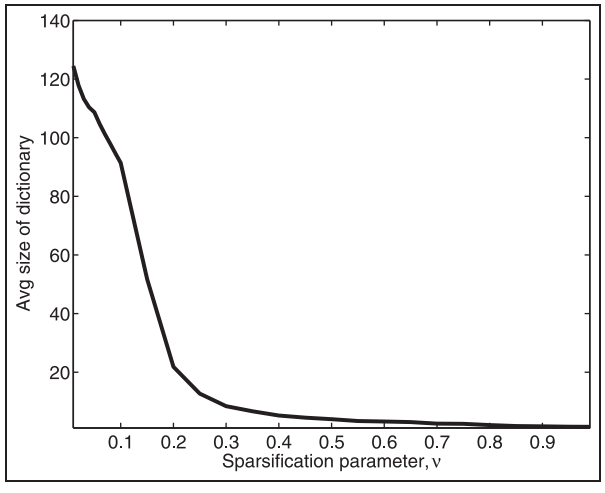

Figure 3. Variation in the average size of KEAD dictionary versus sparsification parameter, ν.

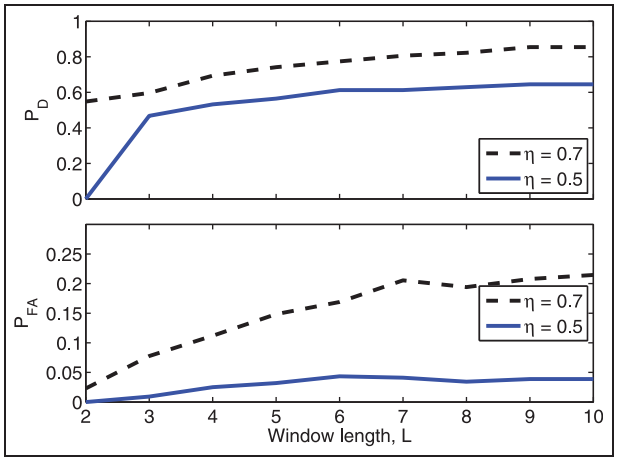

Figure 4. Variation in the detection performance of KEAD as a function of the window length, *L*.

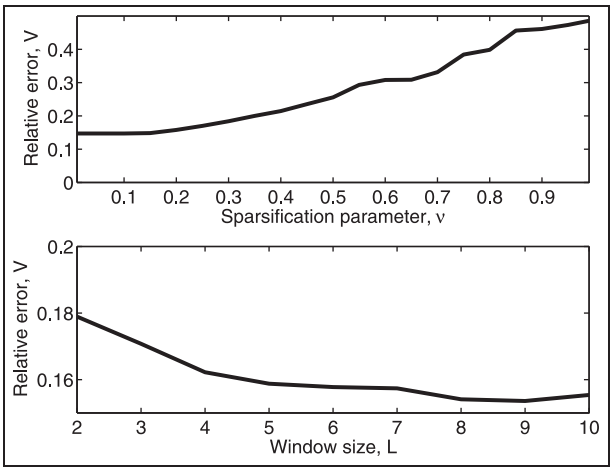

Figure 5. Variation in the average, relative error between true τ and KEAD estimate $\hat{\tau}$, as a function of sparsification parameter ν with default window length (top panel), and as a function of window length L with default sparsification parameter $\nu = 0.20$ (bottom panel).

demonstrating the tradeoff between the probability of detection (P_D) and the probability of false alarms (P_{FA}).

It must be mentioned here that KOAD was run with the standard deviation parameter of the Gaussian kernel set to twice the value recommended by the process described in the beginning of Section 5.4, where we cycle through a number of possible choices between the minimum and maximum values of the component dimensions of the training data points, and then determine the value that produces the greatest variance in the kernel density estimates. It was observed that setting σ to twice this value even further enhanced the difference between normal points and outliers, and subsequently yielded the best results for KOAD. The curve presented for KOAD in Figure 6 is a representative one, obtained by setting $\nu_1 = 0.60$ and varying ν_2.[14]

The curves for the other agents were obtained by varying their respective anomaly detection thresholds. Default settings, as discussed earlier, were used for all parameters. It is evident that the performances of all of KPCA, KOAD and KEAD are significantly superior to those of PCA and Au et al.'s NCD-based schemes. KPCA perform the best, as this is intrinsically a block-based agent, implemented here using a sliding window, while KOAD and KEAD are intrinsically incremental algorithms. The performances of KOAD and KEAD are seen to be comparable, with KOAD faring marginally better. It must be remembered here that the thresholds ν_1 and ν_2 must be manually set in KOAD, while KEAD incorporates autonomous setting for all important parameters.[15] The strikingly low performance of the NCD-based scheme may be attributed to the fact that this scheme requires a significantly longer training period, and needs to maintain a significantly larger database of images to compare new arrivals against.[54]

Figure 7 presents the detection statistics for each agent as a function of time. The location of the 'true' anomalies, as were manually labelled, are indicated as red stems with filled circles. It is clear from Figure 7(a) that KPCA (top panel) and KOAD (middle panel) do the best jobs of isolating the identified anomalies from the normal points, followed by KEAD (bottom panel). The example run of KOAD presented here was obtained using $\nu_1 = 0.10$ and $\nu_2 = 0.60$. It can be further observed from Figure 7(b) that PCA misses a lot of the identified anomalies (top panel), while the performance of Au et al.'s NCD-based similarity metric is clearly the worst (bottom panel), being quite unable to differentiate between normal points and outliers here. These results are in agreement with the ROC curves for the agents presented in Figure 6.

It can be further observed from Figure 7(a) that the proposed agents seem to produce the same missed-detections (black stems of relatively significant height with open circles). Investigating this will form part of our future work.

6.3 Complexity analysis

Storage and complexity issues are paramount to online applications. Assuming m elements in the dictionary, the computational complexity of KOAD is $O(m^2)$ for every standard timestep, and $O(m^3)$ on the rare occasions when an element removal occurs.[12] Assuming that the data points are F-dimensional, the computational complexity of KEAD is $O(mF^2 + m^2)$ for all timestep where an element removal does not occur, in which case it is $O(mF^2 + m^3)$.[15] The complexities are thus independent of time, making the algorithms naturally suited for online use. Our experiments have shown that high sparsity levels

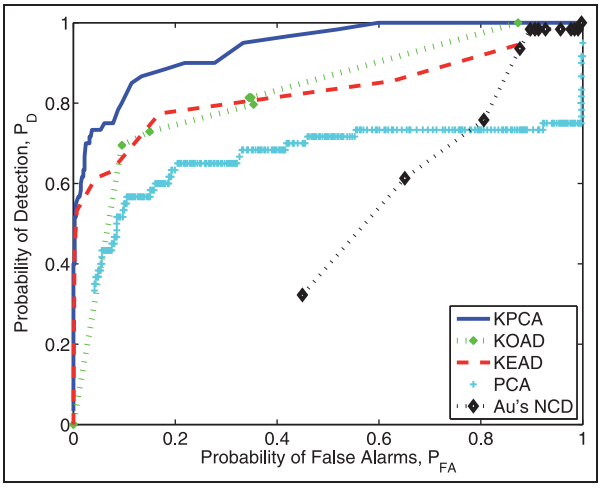

Figure 6. ROC curves showing performances of proposed KPCA, KOAD and KEAD versus existing PCA and Au et al.'s NCD-based algorithms. All three of KPCA, KOAD and KEAD are seen to substantially outperform PCA and NCD.

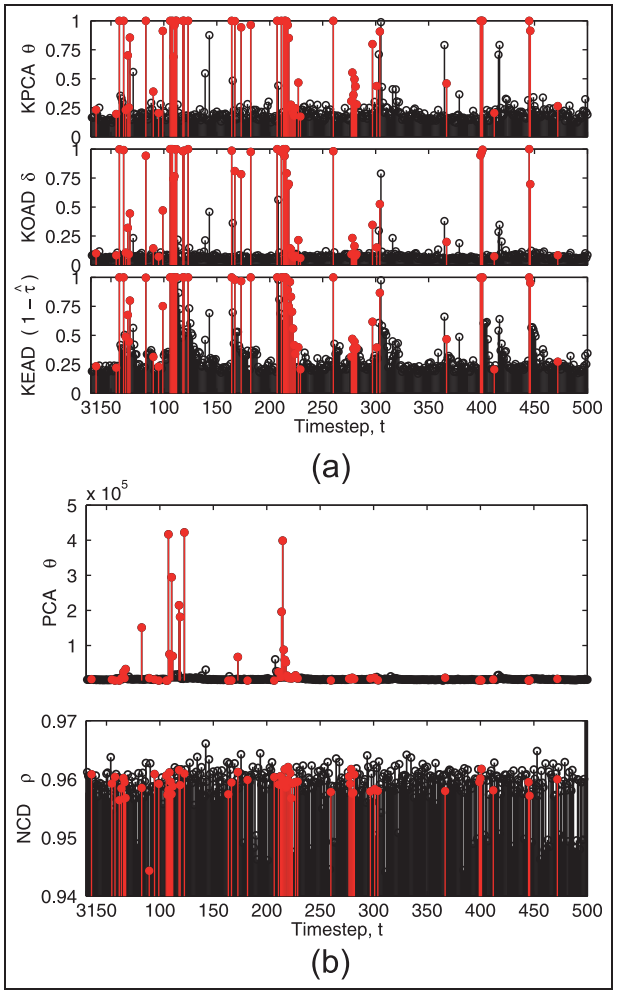

Figure 7. Progression in the anomaly detection statistics for each algorithm for an example setting of the relevant detection thresholds. The true anomalies are indicated as red stems with filled circles. (a) Top panel: Magnitude of projection onto the residual subspace, θ_t, for KPCA. Middle panel: KOAD sparsification statistic δ_t. Bottom panel: $1 - \hat{\tau}_t$, where $\hat{\tau}_t$ is the KEAD online detection statistic. (b) Top panel: Magnitude of projection onto the residual subspace, θ_t, for PCA. Bottom panel: $1 - \rho_t$, where ρ_t is Au et al.'s NCD-based similarity metric.

are achieved in practice, and the dictionary size does not grow indefinitely.

When run over a window consisting of L, F-dimensional data points, KPCA begins by evaluating the full $L \times L$ Gram matrix of kernel values for the window. Assuming a Gaussian kernel function, this involves a computational complexity of $O(L^2F^2)$. Obtaining the eigenvectors of the Gram matrix involves a complexity of $O(L^3)$. Once the number of principal components to be assigned to the normal subspace is decided upon, say R, obtaining the projections involves a complexity of $O(RL^2)$. Thus, the overall computational complexity of running the KPCA algorithm on a window of L, F-dimensional points is $O(L^2F^2 + L^3 + RL^2)$, which may be simplified into $O(F^2)$ in instances where $F \gg L$.

In terms of actual run-times, KPCA took less than 0.04 seconds when run over a window of 30, 2304-dimensional timesteps. KOAD and KEAD took less than 0.0025 seconds to process an arriving 2304-dimensional input vector. We used a Dell personal computer with an i3™ processor and standard configuration. Note that the raw image sequence used in our experiments arrived at 2-second intervals.

Performing PCA over a window of L, F-dimensional points with R principal components assigned to the normal subspace involves an overall computational complexity of $O(LF^2 + F^3 + RLF)$,[15] which may be simplified into $O(F^3)$ where $F \gg L$. Note that the complexity of PCA is an order of F times that of KPCA. This is because KPCA operates on the $L \times L$ Gram matrix of kernel values for the data points, while PCA operates on the $F \times F$ covariance matrix of the points.

For a discussion of the complexities involved in Au et al.'s NCD-based scheme, the reader is referred to Au.[54]

7. Conclusions and future work

In this paper, we have presented three learning agents to develop automated surveillance systems. We have applied the KOAD, KEAD and KPCA algorithms to real footage captured from a simple, closed-circuit television surveillance system.

By virtue of being kernel-based agents, KOAD, KEAD and KPCA look for patterns in a richer feature space, and are able to exploit higher-order correlations between pixels in the sequence of images. The three agents each exhibit different characteristics. While KOAD and KEAD are recursive and inherently provide real-time decisions, KPCA must be implemented using a sliding window approach. KPCA performs best, while KOAD and KEAD are faster. KEAD provides the added benefit of not requiring manual setting of important algorithm parameters, as KOAD must first do during a training period. All three agents are efficient and lightweight, with runtimes of the order of hundredths of a second, making them suitable for high data rate applications.

Our proposed agents do not require any custom equipment or infrastructure, unlike many existing methods. Instead, they may be easily incorporated in already extant and deployed simple surveillance systems. By direct quantitative comparisons with the standard PCA technique and the NCD measure between images first proposed by Au et al.,[23] two schemes on which many existing autonomous IDSs have been based, we have shown that our proposed agents achieve significantly superior performance, along with requiring lower time-to-detection and involving lower complexity. Furthermore, even when run over the simplest possible CCTV system, our agents have been able to easily

achieve near-perfect detection rates. Hence, we advocate our kernel-based agents as inexpensive, efficient and lightweight solutions to the problem of automated surveillance.

This paper is concerned primarily with environments that are typically sparsely populated and mostly static, such as the lobby of a bank or a museum at night. The objective is to enable the officer monitoring the security system to relax, and to draw his attention, in real-time, when the cameras obtain any 'abnormal' image. The learning algorithms applied train to the normal images in the given application. By varying the detection threshold, the operator may control the level of 'abnormality' that he wishes to be alerted to. He may thus decide himself if he wants to be alerted only when a large physical form such as a human intruder/burglar appears, or even when a small change occurs such as light bulb blowing or a cat appearing. Our future work will extend the domain of application into more crowded environments involving data collected from sophisticated multimodal sensors, and on video streams arriving in the order of tens of frames per second. In addition, we wish to integrate the proposed agents with face-detection algorithms to learn the characteristics of the regular visitors to the applicable premises.[55] Furthermore, other methods of feature extraction from the field of image processing will be investigated, for the preprocessing part of our scheme.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments, which greatly helped improve the overall quality and readability of this paper.

Funding

This work was partially supported by the Research Endowment Grant (Type B), Project Title: 'Automated Intruder Detection in Surveillance Networks using Machine Learning Algorithms' (grant number EDW B11-167-0645) at the International Islamic University Malaysia, Kuala Lumpur, Malaysia, for which Dr Al-Sakib Khan Pathan is the Principal Researcher.

References

1. Valera M and Velastin S. A review of the state-of-the-art in distributed surveillance systems. In Velastin S and Remagnino P (eds), *Intelligent Distributed Video Surveillance Systems*. London: Institution of Electrical Engineers, 2008, pp. 1–30.
2. Donald C. Vigilance. In Noyes J and Bransby M (eds), *People in Control: Human Factors in Control Room Design*. Stevenage: Institution of Engineering and Technology, 2002, pp. 35–50.
3. Ko T. A survey on behavior analysis in video surveillance for homeland security applications. In *Proceedings IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, Washington, DC, October 2008.
4. Erasmus mundus joint doctorate in interactive and cognitive environments. Project homepage, http://www.icephd.org/. Accessed 1 August 2012.
5. Remagnino P, Jones G, Paragios N and Regazzoni C. *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*. Boston, MA: Kluwer, 2002.
6. Velastin S and Remagnino P. *Intelligent Distributed Video Surveillance Systems*. London: Institution of Electrical Engineers, 2008.
7. Monari E, Voth S and Kroschel K. An object- and task-oriented architecture for automated video surveillance in distributed sensor networks. In *Proceedings IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Santa Fe, NM, September 2008.
8. Chao W and Jun X. Multi-agent based distributed video surveillance system over IP. In *Proceedings International Symposium on Computer Science and Computational Technology*, Shanghai, China, December 2008.
9. Aguilar-Ponce R, Kumar A, Tecpanecatl-Xihuitl J and Bayoumi M. A network of sensor-based framework for automated visual surveillance. *J Network Comput Appl* 2007; 30: 1244–1271.
10. Shimshoni I, Adam A, Rivlin E and Reinitz D. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Trans Pattern Anal Machine Intell* 2008; 30: 555–560.
11. Schölkopf B and Smola A. *Learning with Kernels*. Cambridge, MA: MIT Press, 2001.
12. Ahmed T, Coates M and Lakhina A. Multivariate online anomaly detection using kernel recursive least squares. In *Proceedings IEEE INFOCOM*, Anchorage, AK, May 2007.
13. Engel Y, Mannor S and Meir R. The kernel recursive least squares algorithm. *IEEE Trans Signal Process* 2004; 52: 2275–2285.
14. Ahmed T, Ahmed S, Ahmed S and Motiwala M. Real-time intruder detection in surveillance systems using adaptive kernel methods. In *Proceedings IEEE International Conference on Communications (ICC)*, Cape Town, South Africa, May 2010.
15. Ahmed T. Online anomaly detection using KDE. In *Proceedings IEEE Global Communications Conference (GLOBECOM)*, Honolulu, HI, November 2009.
16. Parzen E. On estimation of a probability density function and mode. *Ann Math Stat* 1962; 33: 1065–1076.
17. Rosenblatt M. Remarks on some nonparametric estimates of a density function. *Ann Math Stat* 1956; 27: 832–837.
18. Cadre B. Kernel estimation of density level sets. *J Multivariate Anal* 2006; 97: 999–1023.
19. Ahmed T, Wei X, Ahmed S and Pathan A-S. Intruder detection in camera networks using the one-class neighbor machine. In *Proceedings American Telecommunications Systems Management Association (ATSMA) Networking and Electronic Commerce Research Conference (NAEC)*, Riva del Garda, Italy, October 2011.
20. Ahmed T, Wei X, Ahmed S and Pathan A-S. Automated intruder detection from image sequences using minimum volume sets. *Int J Commun Networks Inform Security* 2012; 4: 11–17.
21. Schölkopf B, Smola A and Müller K-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computat* 1998; 10: 1299–1319.
22. Jolliffe I. *Principal Component Analysis*, 2nd edition. New York: Springer-Verlag, 2002.

23. Au C, Skaff S and Clark J. Anomaly detection for video surveillance applications. In *Proceedings IEEE International Conference on Pattern Recognition (ICPR)*, Hong Kong, China, May 2006.
24. Aguilar-Ponce R, Kumar A, Tecpanecatl-Xihuitl J, Bayoumi M and Radle M. Automated object detection and tracking for intelligent visual surveillance based on sensor network. In Zha X (ed.), *Artificial Intelligence and Integrated Intelligent Information Systems: Emerging Technologies and Applications*. IGI Global, 2006, pp. 206–228.
25. Singh R, Vishwakarma S, Agrawal A and Tiwari M. Unusual activity detection for video surveillance. In *Proceedings International Conference on Intelligent Interactive Technologies and Multimedia*, Allahabad, India, December 2010.
26. Yahalom S. *URI Anomaly Detection Using Similarity Metrics*. Master's thesis, Tel-Aviv University, Tel Aviv, Israel, May 2008.
27. Verbeke N and Vincent N. A PCA-based technique to detect moving objects. In Ersbøll B and Pedersen K (eds), *SCIA'07 Proceedings of the 15th Scandinavian Conference on Image Analysis*. Berlin: Springer, 2007, pp. 641–650.
28. Wang T, Gu I and Shi P. Object tracking using incremental 2D-PCA learning and ML estimation. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Process. (ICASSP)*, Honolulu, HI, April 2007.
29. Cohen A, Bjornsson C, Temple S, Banker G and Roysam B. Automatic summarization of changes in biological image sequences using algorithmic information theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2009; 31: 1386–1403.
30. Nowak E and Jurie F. Learning visual similarity measures for comparing never seen objects. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, MN, June 2007.
31. Breitenstein M, Grabner H and Van Gool L. Hunting Nessie: Real time abnormality detection from webcams. In *Proceedings IEEE International Conference on Computer Vision (ICCV) Workshops*, Kyoto, Japan, September 2009.
32. Schuster R, Mörzinger R, Haas W, Grabner H and Van Gool L. Real-time detection of unusual regions in image streams. In *Proceedings ACM International Conference on Multimedia*, Florence, Italy, October 2010.
33. Kuklyte J, Kelly P, Conaire CÓ, O'Connor N and Xu L-Q. Anti-social behavior detection in audio-visual surveillance systems. In *Proceedings Workshop on Pattern Recognition and Artificial Intelligence for Human Behaviour Analysis (PRAI*HBA)*, Reggio Emilia, Italy, December 2009.
34. Kuklyte J, Kelly P and O'Connor N. PhD forum: Investigating the performance of a multi-modal approach to unusual event detection. In *Proceedings ACM/IEEE International Conference on Distributed Smart Camera (ICDSC)*, Ghent, Belgium, August 2011.
35. Gowsikhaa D, Manjunath and Abirami S. Suspicious human activity detection from surveillance videos. *Int J Internet Distrib Comput Syst* 2012; 2: 141–148.
36. Omerčevic′ D, Drbohlav O and Leonardis A. High-dimensional feature matching: Employing the concept of meaningful nearest neighbors. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, October 2007.
37. Dalal N and Triggs B. Histograms of oriented gradients for human detection,. In *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, June 2005.
38. Honeywell international incorporated. Company homepage, http://www.honeywell.com/. Accessed 1 August 2012.
39. V. Morellas, I. Pavlidis and P. Tsiamyrtzis. DETER: Detection of events for threat evaluation and recognition. *Machine Vision and Applications*, 15(1): 29–45, Oct. 2003.
40. ObjectVideo. Company homepage, http://www.objectvideo.com/. Accessed 1 August 2012.
41. NICE Systems. Company homepage, http://www.nice.com/. Accessed 1 August 2012.
42. Ipsotek Ltd. Company homepage, http://www.ipsotek.com/. Accessed 1 August 2012.
43. Proakis J and Salehi M. *Digital Communications*, 5th edition. New York: McGraw-Hill, 2007.
44. Lakhina A, Papagiannaki K, Crovella M, Diot C, Kolaczyk E and Taft N. Structural analysis of network traffic flows. In *Proceedings ACM SIGMETRICS*, New York, NY, June 2004.
45. Sudo K, Osawa T, Wu X, Wakabayashi K and Yasuno T. Detecting the degree of anomal in security video. In *Proceedings IAPR Conference on Machine Vision Applications*, Tokyo, Japan, May 2007.
46. Li M, Chen X, Li X, Ma B and Vitanyi P. The similarity metric. *IEEE Trans Inform Theory* 2004; 50: 3250–3264.
47. Keogh E, Lonardi S and Ratanamahatana C. Towards parameter-free data mining. In *Proceedings ACM SIGKDD*, Seattle, WA, August 2004.
48. Cebrian M, Alfonseca M and Ortega A. The normalized compression distance is resistant to noise. *IEEE Trans Inform Theory* 2007; 53: 1895–1900.
49. Cilibrasi R and Vitanyi P. Clustering by compression. *IEEE Trans Inform Theory* 2005; 51: 1523–1545.
50. Lanckriet G, Cristianini N, Bartlett P, Ghaoui L and Jordan M. Learning the kernel matrix with semidefinite programming. *J Machine Learn Res* 2004; 5: 27–72.
51. Brauckhoff D, Salamatian K and May M. Applying PCA for traffic anomaly detection: problems and solutions. In *Proceedings IEEE INFOCOM Miniconference*, Rio de Janeiro, Brazil, May 2009.
52. Ringberg H. *Privacy-Preserving Collaborative Anomaly Detection*. PhD thesis, Princeton University, Princeton, NJ, September 2009.
53. Skolnik J. *Radar Handbook*, 2nd edition. New York: McGraw-Hill, 1990.
54. Au C. *Compression-based Anomaly Detection for Video Surveillance Applications*. Master's thesis, McGill University, Montreal, QC, Canada, February 2006.

55. Faruqe M and Hasan M. Face recognition using PCA and SVM. In *Proceedings IEEE International Conference on Anti-counterfeiting, Security, and Identification in Communication (ASID)*, Hong Kong, China, August 2009.

Author biographies

Tarem Ahmed was born in Dhaka, Bangladesh. He received a Bachelors degree with a double major in physics and economics from Middlebury College, Middlebury, VT, USA in 1999 and a Masters degree in electrical engineering from the University of Pennsylvania, Philadelphia, PA, USA in 2000. After serving in industry as an ASIC design engineer in the Silicon Valley area of CA, USA, he has held research positions at the department of electrical and computer engineering at McGill University, Montreal, QC, Canada and the computer engineering and networks laboratory at the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. He is presently affiliated with the department of computer science at the International Islamic University Malaysia, Kuala Lumpur, Malaysia and the department of electrical and electronic engineering at BRAC University in his native city of Dhaka, Bangladesh.

Xianglin Wei was born in Anhui province, China. He received a Bachelors degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China in 2007, and his PhD degree in PLA University of Science and Technology, Nanjing, China. He is now working as a researcher at the Nanjing Telecommunication Technology Institute, Nanjing, China. His research interests include cloud computing, peer-to-peer network, network anomaly detection, network measurement, and distributed system design and optimization.

Supriyo Ahmed was born in Dhaka, Bangladesh. He received a Bachelors degree in electrical and communications engineering in 2008, and a Masters degree in electrical engineering in 2012, both from BRAC University, Dhaka, Bangladesh. He is presently serving the same university as a lecturer in the department of electrical and electronic engineering, and in a number of other administrative roles.

Al-Sakib Khan Pathan received a PhD degree in computer engineering in 2009 from Kyung Hee University, South Korea. He received a BSc degree in computer science and information technology from Islamic University of Technology (IUT), Bangladesh in 2003. He is currently an assistant professor at computer science department in International Islamic University Malaysia (IIUM), Malaysia. Until June 2010, he served as an assistant professor at the computer science and engineering department in BRAC University, Bangladesh. Prior to holding this position, he worked as a researcher at networking lab, Kyung Hee University, South Korea until August 2009. His research interest includes wireless sensor networks, network security, and e-services technologies. He is a recipient of several awards/best paper awards and has several publications in these areas. He has served as a chair, organizing committee member and technical program committee member in numerous international conferences/workshops such as HPCS, ICA3PP, IWCMC, VTC, HPCC, IDCS, etc. He is currently serving as the editor-in-chief of IJIDCS, an area editor of IJCNIS, editor of IJCSE, Inderscience, associate editor of IASTED/ACTA Press IJCA and CCS, guest editor of some special issues of top-ranked journals, and editor/author of five published books. He also serves as a referee for some renowned journals. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), USA; IEEE Communications Society (IEEE ComSoc), USA, and IEEE ComSoc Bangladesh Chapter, and several other international organizations.