8 YOU ARE WATCHING SYSTEM (YAW SYSTEM): AN AGENT BASED SECURE REAL-TIME MONITORING, CAPTURING AND ANALYSIS SYSTEM FOR CLIENT ACTIVITIES ON THE EXPERT AGENT’S SCREEN

8.1 Introduction

The use of the Internet for Web-based Intelligent Examination System is an effective solution for mass level of educating and evaluating students in a faster way. However, since it is difficult to verify the identity of the student through a simple user ID and password system on the client side, performance evaluation through test and examination through the Internet is still in its infancy. Further illegal sites such as any emailing sites, tutorials or any other unwanted sites may be used by students to carryout giving answers and difficult to evaluate as distant learning examination.

To overcome these main hurdles, a You Are Watching System (YAW) has been designed and developed where a You Are Watching Client Agent (YAWCA) at the client node is used to capture the student’s screen and posting at particular intervals during the test. The interval for screen capturing can be either automatic or can be set manually by the You Are Watching Server Agent (YAWSA) for each student/learner.

The captured images are stored at the You Are Watching Server (YAWS) and can be used to verify the identity (using Student’s ID and combination of Client node IP Address+ MAC Address) of the student taking the test if the need arises.

The screen captured at the client node is said to be target image (Ti) and the expected compared legal image available in the YAWS is said to be
source image (Si). The Si and Ti image are compared using a well-known general purpose optimization Kernel Direct Discriminant Analysis (KDDA) algorithm. If a particular student is found as illegal screen, then the student will be warned thrice by the YAWSA and later on again illegal screen is encountered, then YASWA can block that student’s login and notify to student along with teacher/Expert Agent.

The system developed has been successfully used in a Human Knowledge Discovery Agent (HKDA) System. From the beginning to end of the examination, monitoring the student/trainer’ conditions on the teacher’s/Expert Agent’s screen. If there is a change of power failure or Internet disconnect in the client node then it is also tracked by the YAW Agent.

8.2 How You Are Watching Works

YAW System consists of a Client as YAW Client Agent (YAWCA) and a Server as YAW Server Agent (YAWSA). The YAW Client is installed in every managed computer and is responsible for collecting the information about all assets (applications, and other computing resources) in that machine. The YAW Server receives the information in real time and stores it in the Data Layer as Document Repository.

The administrator on logging in is able to access the information stored in the Document Repository in the form of asset reports. He can then define which of the assets in a particular machine are usable by a specific user. These asset usage rules are communicated to the YAW Client Agent. The YAW Client Agent will hook every process invocation attempt and passes it through a validation mechanism. The validation mechanism uses the rules set by the YAW administrator from the YAW Server. Any unauthorized invocation attempts are denied and details logged.
Figure 8.1: Client Desktop Screen Image (Student Examination Screen)

Converting Figure 8.1: Client Desktop Screen Image (Student Examination Screen) into Grayscale Image

Convert

Source Image (Si) {Color}  Source Image (Si) {Gray}

[Windows XP, 600x800]  [Windows XP, 600x800]

Size: 7k  Size: 4k

Figure 8.2: Convert Client Desktop Screen Colored Image into Gray Scale
8.3 Image Registration Process

Figure 8.3 describes the main steps in the registration process. A YAWSA opens Si and Ti images. He selects the type of transformation to be used in the registration process. Depending on the transformation chosen, the YAWCA asks the YAWSA to select two or three landmarks. Landmarks are recognizable physical features in the image, and the YAWSA selects them.

In Client Desktop Screen Images, typical features which can be selected as landmarks are of Operating System, Web Browser, Resolution, texts, or some other characteristic distinguishable points in the image. The same physical features need to be marked in both the Si and the Ti images. User
selections are marked on the screen by corresponding cross hairs at manually selected image locations.

The cross hairs are gray-coded, that is, cross hairs corresponding to the same landmark have the same gray in both the Si and the Ti images. Once the selection of landmarks is completed, the user initiates image registration process, which generates a registered image.

Under the assumption that the Si and the Ti images represent the same Client Desktop Screen Image, registration is successful if the landmarks in the registered image are “sufficiently” close to the landmarks in the Ti image. Successful registration enables a positive match to be established by the YAWSA. However, researcher had refined this by subjective success metric.

Another aspect that could influence the success of the registration process is the quality of the image being considered. The degree of self-similarity among the Client Desktop Screen Images is very high. Therefore, blurred images might cause the alignment optimization algorithm to end up stuck in some local optimum. The probability of this type of failure should decrease in sharp images that clearly depict details.

8.4 Scalability and Deployment

The YAW System must be able to accommodate all class sizes, even the largest. If class size is large, factors such as CPU load, network congestion, and storage limits may affect system performance. Researcher focuses on performance degradation and deployment overhead in large classrooms. Therefore, researcher identified the following four scalability requirements that guided the design of YAW System:
• Distributed agents as YAWCA should be used so that each agent is responsible for only one student computer and CPU load is distributed across multiple computers. Then the system can extend to a large class without worrying about the CPU load on any one computer.

• The YAW System should use the network at off-peak times, rather than during class, to minimize the impact of large log file transfer.

• To handle the large volume of collected data, YAW System should also have sufficient data-handling capabilities that allow the data to be stored, retrieved, and analyzed efficiently.

• To avoid repeated configuration work on each computer, distributed agents YAWCA should report their collected data to a YAW Sever. Each YAWCA should read a configuration file from a central directory and adjust its settings accordingly.

8.5 Image Capturing Steps
Step 1: Repeat the process for each Student/Learner Agent who gives the Examination.

Step 2: Capture Image [Ti] from Client Desktop Screen of Students/Learner Agents.

Step 3: Crop the Ti from Bottom Left corner of the Client Desktop Screen which include Start Button and Web Browser [Start Button reflects the type of the OS used and Web Browser reflects the type of Web Browser used by the Student/Learner Agent]

Step 4: Convert the Ti into Gray Scale Image. [This conversion process is convert large colored image into near about 4k Gray Scale Image]
Step 5: After that YAWCA sends the Ti Gray Scale Image to YAW Server with proper time stamp. [Time Stamp is based on User Id-Institute Id-Course Id-Subject Id-Exam id-OS Type-Window Resolution Type-Browser Type-Date-Time]

Step 6: YAWSA compares the Gray Image of the Ti with the Si which is already available in YAW Server. [Kernel Direct Discriminant Analysis (KDDA) Algorithm]

Step 7: If the match found correct then YAWSA allows Student(s)/Learner Agent(s) to continue in examination.

Step 8: If the match found incorrect then YASWA provides warning thrice to Student(s)/Learner Agent(s) and stops to continue examination.

Step 9: Reports of legal/illega activities carried by Students are sent to Communication Agent, Advising Agent and Expert Agent.

8.6 Kernel Direct Discriminant Analysis Recognizer

Client Desktop Screen Image recognition using Kernel Direct Discriminant Analysis (KDDA), is based on computation of feature space \( F \) (from training set) and projection of input pattern into the feature space to calculate significant discriminant features. The feature space \( F \) is calculated using a set of \( N \) training Client Desktop Screen Image images where each image \( N_i \) is defined as a vector of length \( L (= I_w \times I_h) \), i.e. \( N_i \in F \), where \( I_w \times I_h \) is the

Client Desktop Screen Image size. Then the average Client Desktop Screen Image of training set \( N \) is calculated by
$$\psi = \sum_{i=1}^{N} N_i$$

Each Client Desktop Screen Image difference from average Client Desktop Screen Image is calculated as

$$\phi_i = \psi - N_i$$, and matrix $\Phi_b$ is formed as

$$\Phi_b = [\phi_1, \ldots, \phi_N]$$. Eigenvectors and Eigenvalues are calculated using the kernel matrix $K$

$$K = \Phi_b^T \Phi_b$$

In order to remove null values in feature space only eigenvectors whose corresponding eigenvalues are greater than zero are used to form the low dimensional representation ($\Theta$) For creating the feature space $F$, the following algorithm is applied on a set of training Client Desktop Screen Image images.

**Algorithm KDDA (Si: YAWS)**

begin
for each input/original image in YAWS Si

Step 1: Make a set of training Client Desktop Screen Image images where each Client Desktop Screen Image is represented as n – dimensional vector.

Step 2: Calculate the Kernel matrix ($K$), eigenfaces, and eigenvectors having positive eigenvalues.

Step 3: Calculate the matrix $\Theta$ which causes the low dimensional representation of the feature space.

Step 4: Calculate a vector $y$ which contains optimal discriminant features as
\[ y = \Theta \ast \gamma(\phi(z)) \]

where \( z \) is the input pattern.

Step 5: Store the feature vector \( y \) in the YAWS Si end

Similarly for each Client Desktop image optimal discriminant features can be calculated by projection into the KDDA feature space. Then the matching score is calculated using the following algorithm

**Algorithm MSKDDA (Si: YAWS, Ti: Client Desktop)**

begin
Let there be \( m \) features of the given image in the YAWS and \( n \) features in the Client Desktop image \( Ti \). For each of the \( m \) features in the YAWS and \( n \) features in the Client Desktop image, reference features are chosen depending on the distance and rotation between the positions of features in the feature space.

Step 1: Translate the feature vectors of the YAWS and the Client Desktop images with respect to the reference feature chosen and then convert into polar coordinates.

Step 2: Import the relevant bounding box and for each of the \( m \) features in the YAWS

Step 3: Find Client Desktop image features \( n \) that lie within the bounding box. Increment the matching score accordingly.

Step 4: Calculate matching score for each transformation of YAWS and Client Desktop feature vectors with respect to reference feature chosen
Step 5: MSKDDA is defined by the maximum of all matching scores divided by the maximum number of features (among the Client Desktop and the YAWS).

Given training images for input, these recognizers extract significant features and store it in their respective YAW System. KDDA the optimal discriminant features are calculated and stored in the YAW System.

8.7 Process Hooking

The core of the YAW System is the ability to control application execution on a given machine by a given user. This provides addition protection against illegal users who act as the learner agent for test/examination and all malicious code in general that can impact the integrity of the network as a whole.

The product utilizes a proprietary HKDA System to hook processes at creation itself. Processes are hooked at every mode and passed through a validation mechanism. The YAW administrator can configure the validation rules. The process creation is allowed to continue only if the User invoking the application IS ALLOWED to do so in that specific machine. Any illegal attempts to run the application are logged and the administrator is notified. The advantages of this technology are:

1. The protection mechanism is fully transparent to general users

2. Being in the every mode, it is next to impossible to override it

3. There is no degradation in machine performance.
8.8 Conclusion

With the help of YAW System, Institution is identifying all the test/examination in a good manner. This increases security within the Institution by identifying unauthorized users as well as preventing their use on protected machines thereby mitigating the full assurance of test/examination over the network.

This system is creating audit trails of infractions in the whole network. This system automatically scans the network for potential vulnerabilities in configuration and missed patches. The YAWSA is alerted in such cases. This system also prevents Legal and IP issues by giving advance warning to the use of unauthorized applications within the organization.