SURVEILLANCE STRATEGY USING A VARIANCE OF RADIAL BASED FUNCTIONS FOR CELLULAR BASED NETWORKS

Prepared By: KANGETHE MICHAEL MAIGWA MARTIN

Supervised by: DR. ROBERT OBOKO

MSC COMPUTER SCIENCE

UNIVERSITY OF NAIROBI

2012

This report is submitted in partial fulfillment of the requirements for the degree of Master of Science Computer Science of the University of Nairobi. This report is the product of my own work except where indicated in the text. The report may be copied and distributed freely in part or whole as long as the source is acknowledged.

DECLARATION

This research and thesis is my original work and has not been presented for a degree in any other university or for any other award.

Signature__________________ Date 9th Nov 2012

Michael Maigwa Martin Kangethe
University of Nairobi
Reg: P58/61780/2010

This thesis has been submitted with my approval as the university supervisor

Signature__________________ Date 12/11/2012

Dr Robert Oboko
School of Computing and Informatics
University of Nairobi

Acknowledgement

This work is first and foremost dedicated to God for His gracious mercies and blessings in enabling me to not only conceptualize this Idea, but to develop this research and complete it.

Its better – much better- to have wisdom and knowledge than gold and silver (*Proverbs 16:16*)

I would like to take this special opportunity to express my most and heartfelt gratitude to my supervisor Dr Robert Oboko firstly for agreeing to supervise me, and secondly for his patience and devoted effort in guiding me during the project duration. This work has come to fruition due to his thorough repeated reading and guidance on the best way to develop it. I heartily acknowledge and value your professional guidance and sincere friendship Thank you *Mwalimu*.

I would also like to recognize the invaluable role and advice the panelists Mr. Stephen Mburu , MS Christine Ronge, Prof Okelo Odongo for their constructive criticism and keen interest in my project, their guidance and direction has greatly contributed to the quality of this research.

I may also like to take this opportunity to thank my friend and colleague Mr. Kiprotich Chemweno for his interest and valuable contribution and encouragement during the project development phase. Very few people may be willing to understand encourage unconditionally and selflessly assist a fellow researcher with his/her research while working on his own project at the same time. You happen to be the very few. May God bless you *asante sana.*

Finally I would like to offer my special thanks and express my gratitude to my Family. My mother Dr Rachel Wanjitu Kamau, My Step-Father Mr. Peter K. Njuguna. Only God can repay you for your contribution and all that you have done for me and believing in me all my life. I am truly blessed and I shall always be the example I we need, my sisters Stephanie Joan Wacu, Patience Beatrice Nyokabi. God couldn't have given me better Sisters thanks for your encouragement and support and always remember my success will and has always been our success I love you dearly.

Reduced privacy is a small price to pay for enhanced National Security and personal safety.

Abstract

This is the study and implementation of a mass population interaction surveillance technology framework, using the Radial Based Functions Algorithm using the Radio based (Cellular to be specific) network and to develop surveillance systems strategy. Which enables the efficient and automated identification of entity interactions and potential relationships between several entities and events based on a hierarchy of interactions? The scientific approach to this problem is to combine a modified variation of the Radial Basis Functions algorithm computing theory to develop a computing system that enables the tracking of individual entity's relationship with others based on their interaction judged by their proximity to the entity of interest as the future of automated surveillance will not just include the collection of geographic and visual data but also intelligence on the particular entity's interaction log information from activity patterns which can be mapped in an easy to present format to the interested parties.

Table of Content

List of Figures

List of Tables

Acronyms:

MS	Mobile Station /Mobile Phone
RFID	Radio Frequency Identification (RFID)
BTS	Base Transceiver Station/System
GPS	Global Positioning System
IMEI	International Mobile Equipment Identifier
ID	Identification Number (National)
ANN	Artificial Neural Networks
GSM	Global System for Mobile Communications
POI	Person of Interest
Shadow MS	These are all the active Mobile phones that were detected near the Person of Interest at any particular time.
Shadow ID	This is the Identity of the individual(s) who owns the Shadow Mobile Station or Shadow Mobile Phone.

Definitions:

Mobile Station: This is an active Mobile Phone under observation.

Shadow MS: These are all the active Mobile Numbers that are within an area of proximity from the Target Mobile station.

Global Positioning System: is a space-based satellite navigation system that provides location and time information in all weather, anywhere on or near the Earth, where there is an unobstructed line of sight to four or more GPS satellites. It is maintained by the United States government and is freely accessible by anyone with a GPS receiver.

International Mobile Equipment Identity: s a number, usually unique, to identify mobile phones, as well as some satellite phones. It is usually found printed inside the battery compartment of the phone. It can also be displayed on the screen of the phone by entering ***#06#** into the keypad on most phones.

CHAPTER 1: INTRODUCTION

Security and surveillance has always been intertwined and have resulted in the advancement of individual and public safety in general. Surveillance strategies have been developed to target or observe specific phenomena related to an investigated occurrence involving multiple parties. The observation of individual Persons of Interest for intelligence gathering on the subjects suspected activities or association of the subject to a particular group or event of interest has been the most practiced form of surveillance. But as the number of individuals being investigated increases and the communication technology medium improves so does the difficulty in surveillance.

The main purpose of surveillance has been to provide intelligence that will be used as a guiding factor for strategic and tactical decisions concerning the particular subject of interest and the possible and /or affected stakeholders (weather direct or indirect depending on either an event or an individual).

This has resulted into better managed civilizations that are managed by structures and authorities that have been put in place to monitor and control the activities and operations that have and are being performed by their subjects based on the predefined set of laws and policies.

In human activities and events, once is considered an isolated occurrence, twice is a coincidence and thrice is a pattern. Question is how to identify the patterns, with limited resources and no knowledge of the parties involved using Artificial Intelligence, Data mining techniques for the currently available mobile telecommunications technology.

1.2 Problem Statement

The growth of populations, the adoption and advancement of technology, has drastically changed people's behavioral patterns, motivations and approach to known and unknown issues. This has made it difficult for the authorities to monitor and maintain its population's safety. This has been influenced by several factors which include but are not limited to the below:

1. Number of observed parties
2. Distance between the observed parries
3. The interaction and communication medium and techniques.
4. Relationships between the parties in relation to the case being investigated.
5. Time of observation.
6. Purpose of the surveillance.

This problem is greatly increased when observing a population of over 1 million subjects daily to identify previously unknown patterns that might create justifiable curiosity and follow-up that will lead to either the prevention or containment of undesirable circumstances.

The questions raised and have been considered during the research of the research study include:

1. How can we be able to observe and collect quantifiably qualitative and reproducible results based on the observation of a mass population with no particular specified single objective in mind?

2. How can we be able to positively and with a desirable degree of proof associate observed subjects to events of interest?
3. How can we quantify the degree of association between two or more parties based on their documented interactions and proximity (Both physical and social) to each other?
4. How can we identify unknown patterns based on the time of observation of populations in relation to events?

1.3 Hypothetical Scenarios

This observation is partly justifiable by the following hypothetical scenarios:

- A group of people who are involved in criminal acts that are being investigated. One of them is known by the authorities but the rest remain unknown. In order to arrest all the members of that group they are first supposed to identify them all. One way is to follow the person of interest and try to identify his or her accomplices when they meet. The other way is to arrest and interrogate the subject to reveal the others. Of-course this methods have their own limitations as the first one might expose the investigating officers to the scouting criminals who act as remote scouts observing the environment.
- A crime committed by a member of a criminal outlawed group at a specific location and the assailant number telephone number is not known. The target is to find out the possible assailants and their accomplices. One way to find out the possible suspects is to try to identify numbers that were active at that location at that particular time at that date and to use statistical models to narrow down the suspects list to possible assailants.
- An outbreak of a contagious disease is recorded by the identification of a patient at a hospital. In-order to contain the outbreak the center of disease control will need to track the previous, current and future movement of the patient and his/her close encounters with the unsuspecting public. Since it will be difficult and almost impossible to know the patients whereabouts from the tine of infection incubation to the first symptoms. Tracking all possible can be done better by identification of active numbers near a certain proximity to the infected party this will reduce the cost of containment considerably.

These are three of the cases of focus that have guided the research. They are hypothetical cases gotten from the local Newspapers, The local television station news broadcast and from individual elaborations of past events and experiences. They represent events that happen in real life daily around the world that need to be avoided, controlled or arrested. This main focus of this research has been on the development of an automated system framework and demonstrated by the use of a developed prototype and simulated data that reflect the above scenarios that the system is supposed to monitor and identify the possible individual associations. It has focused on the use of mobile telecommunications technology and Machine Learning algorithms for Data Mining of the relevant information detailed earlier.

The reasons for the inclusion of this hypothetical situations is because when an event of interest happens there are people involved, and due to the widespread use of mobile technologies among the populations, tracking of the individuals of interest becomes easier thus the need to identify activity and association patterns between the individuals that will, with a high degree of validity indicate the likelihood of the involvement of specific people to events. This is a surveillance strategy that mainly focuses on the improvement of national security via monitoring the activity patterns that involve the use of mobile technology tracking and communication information.

1.4 Objective

1.4.1 Main Objective

The main objective of this research is to formulate a strategy for identifying patterns and level of associations between parties involved in an event or associated to a particular subject of interest. Using the Mobile Telecommunications Technology (Cellular Networks to be specific) and to develop a prototype to simulate the framework model developed by this research.

1.4.2 Specific Objectives

Due to the varying level of associations between different entities, the research is supposed to formulate a framework Model that can be used to compute the two types of Cases.

1. The likelihood of different Entities being related to each other from an Entity based Computation model.
2. The likelihood of associations between entities or people to specific events from an Event Based Computation Model.

1.5 Significance of Research

Due to the difficulty in tracking human group activity patterns and associations due to the size of a local population and the use of telecommunications technology, The impact of this project if implemented successfully will result in the better faster and cheaper automated process of surveillance for the sake of both national security and social behavioral study by the use of the currently existing (GSM) technology. This will enable the relevant authority not only to monitor its governments populations better but will result to the increased safety of the people by the provision of better intelligence concerning not only the people and associations to each other but also events of interest.

CHAPTER 2: LITERATURE REVIEW

2.1 Surveillance

Surveillance is the monitoring of the behavior, activities, or other changing information, usually of people for the purpose of influencing, managing, directing, or protecting. It most usually refers to observation of individuals or groups by government organizations, but disease surveillance, for example, is monitoring the progress of a disease in a community.

The word *surveillance* may be applied to observation from a distance by means of electronic equipment (such as CCTV cameras), or interception of electronically transmitted information (such as Internet traffic or phone calls). It may also refer to simple, relatively no-or low-technology methods such as human intelligence agents and postal interception. Surveillance is very useful to governments and law enforcement to maintain social control, recognize and monitor threats, and prevent/investigate criminal activity. (*www.wikipedia.org [online]. (3 June 2012). Available from: <http://en.wikipedia.org/wiki/Surveillance>. [Accessed 14 June 2012].)*

Surveillance can be conducted in various forms depending on the particular reasons or purpose. This may range from:

- Individual to group surveillance targets
- Fixed or roaming mobile location surveillance

The above factors can be used to determine the type of surveillance and strategies to be observed and followed when surveying the target(s) as listed below.

The Types of surveillance include:

a) **Computer surveillance:** this generally involves monitoring the data and traffic over the internet.
b) **Telephones:** This generally involves the real-time tapping and monitoring of the targets call logs messages and also the remote activation of the targets phone's microphone to listen in on the target unknowingly.
c) **Surveillance cameras:** this is the use of video cameras for observing an area.
d) **Social network analysis:** this is the creation of maps of social networks based on data from social networking sites to extract useful information.
e) **Biometric surveillance:** This refers to technologies that measure and analyze human physical and/or behavioral characteristics for authentication, identification, or screening purposes.
f) **Aerial surveillance:** is the gathering of surveillance, usually visual imagery or video, from an airborne vehicle—such as an unmanned aerial vehicle, helicopter, or spy plane. Military surveillance aircraft use a range of sensors (e.g. radar) to monitor the battlefield.
g) **Data mining and profiling:** is the application of statistical techniques and programmatic algorithms to discover previously unnoticed relationships within the data. Data profiling in this context is the process of assembling information about a particular individual or group in order to generate a profile— that is, a picture of their patterns and behavior. (Which is vital to this research)?
h) **Corporate surveillance:** Corporate surveillance is the monitoring of a person or group's behavior by a corporation.

The data collected is most often used for marketing purposes or sold to other corporations, but is also regularly shared with government agencies.

i) **Human operatives:** This is the infiltration of Organizations that have enemies who wish to gather information about the groups' members or activities by individuals or groups (mostly know as spies).

j) **Satellite imagery:** The satellites and aircraft sensors will be able to penetrate cloud cover, detect chemical traces, and identify objects in buildings and "underground bunkers", and will provide real-time video at much higher resolutions than the still-images produced by programs such as Google Earth

k) **Identification and credentials:** One of the simplest forms of identification is the carrying of credentials. Some nations have an identity card system to aid identification, whilst many, such as Britain, are considering it but face public opposition. Other documents, such as passports, driver's licenses, library cards, banking or credit cards are also used to verify identity.

l) **RFID and Geo-location devices:** Radio Frequency Identification (RFID) tagging is the use of very small electronic devices (called 'RFID tags') which are applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. The tags can be read from several meters away. They are extremely inexpensive, costing a few cents per piece, so they can be inserted into many types of everyday products without significantly increasing the price, and can be used to track and identify these objects for a variety of purposes.

- 1.12.1 RFID tagging
- 1.12.2 Global Positioning System
- 1.12.3 Mobile phones

The subject, distance, size of the population event, location, accessible technology (to both the target and the observer), mobility (of both the target and the observer) will determine both the technology and strategies used during the surveillance.

2.2 Machine Learning

2.2.1 Introduction

Machine Learning is a branch of computer science specifically Artificial Intelligence, that deals with the research and development of computer systems (Mainly Software) that automatically improve its performance in a given task with experience. This improvement is either in efficiency (such as a Robot navigation system finding the shortest route to its destination while driving within a set path), or effectiveness (such as learning from medical records which treatments are most effective for new diseases). This is achieved by the use of special algorithms invented specifically for learning tasks given relevant single or multivariate training examples. There are several algorithms that are specific to machine learning and data mining that have been developed for various purposes from the prediction of weather patterns to voice, handprint and image recognition in security biometric systems.

In artificial intelligence specifically machine learning and data mining, algorithms have been designed using mathematical principles to identify patterns and associations in voluminous data for the purpose of making predictions or explaining phenomena that is otherwise unknown and there is no known model of computing the prediction. There are several classifications of algorithms that are designed for that task which include but are not limited to:

2.2.2 Decision Tree Learning

This is a method for approximating discrete-valued target functions. The learned function is represented by a decision tree or as if-then rules for <value attribute>pair instances which are easier for humans to read. It is best used in classification of medical patients by their disease, equipment malfunctions by their cause and other similar classification (discreet) function problems. Such class of algorithms include the ID3 Tree Learning algorithm which uses A top-down greedy search through the hypothesis space of possible decision trees where in each step, the "best" attribute for testing is selected using some measure (statistical measure-how well alone it classifies the training examples), and branching occurs along its values, continuing the process for examples sorted through lower nodes and Ends when all attributes have been used, or all examples in this node are either positive or negative. This class of algorithms is limited to the classification of instances to specific discrete output values target functions and not patterns and entity-event association likelihood based on independent and changing data within the collected instance entity records, (every instance of computation does not affect all the target values of other instances which is required in this research). *(Tom M. Mitchell (1997). Decision Tree Learning. In: McGraw-Hill, (ed). Machine Learning, Canada: McGraw-Hill, pp52-56)*

2.2.3 Concept Learning

Concept learning can be formulated as a problem of searching through a predefined search space of potential hypothesis for the hypothesis that best fits the training examples. It is a form of prediction algorithm class that gives the definition of a general category given a sample of positive and negative training examples of the category. Concept learning can be formulated as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples. In many cases this search can be efficiently organized by taking advantage of a naturally occurring structure over the hypothesis space-a general to-a specific ordering of hypotheses.

In surveillance of large populations the main objective is to identify patterns caused by either recurrences or multiple interactions of several persons to a single entity or event. An example of this class of algorithms include the Find-S which tries to find a single more general Hypothesis that is consistent with the training examples in the hypothesis space by beginning with the most specific possible hypothesis in H, then generalize this hypothesis each time it fails to cover an observed positive training example till it fits all the positive training examples in the Hypothesis space. And Candidate Elimination Algorithms which tries to find one or more hypotheses that are consistent to the hypothesis space using version spaces based on the more general and specific boundary described as

$$VS_{H,D} = \{h \in H \mid (\exists\, s \in S)(\exists\, g \in G)(g \geq h \geq s)$$

Equation 1

Where $x \geq y$ means x is more general than or equal to y.

These category of algorithms are good in describing a general hypothesis from a collected set of observed data that produces a desired result. But their limitation in this research is that in surveillance and specifically person-to-person and event-to-person association the likelihood value of association cannot be computed as it doesn't rank the association of one entity at any instance to an event or to another entity at any instance which is a fundamental flaw if applied. *(Tom M. Mitchell (1997). Bayes Theorem and Concept Learning. In: McGraw-Hill, (ed). Machine Learning, Canada: McGraw-Hill, pp159-164)*

2.2.4 Bayesian Learning

This is a class of machine learning algorithms which provides a probabilistic approach to inference. It is based on the assumption that the quantities of interest are governed by probability distributions and that optimal decisions can be made by reasoning about these probabilities together with observed data.

In mapping of trigger functions to symbolic inputs that are hard to quantify has been a major improvement to identifying patterns and relationships between two or more symbols. It is also useful as it provides a framework within which many non-Bayesian classifiers can be studied.

Features of Bayesian learning methods:

a) Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct, unlike algorithms which completely eliminate a hypothesis if it is inconsistent with any single example

b) Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. Prior probability is provided by asserting a prior probability for each candidate hypothesis and also a probability distribution over observed data for each possible hypothesis

c) Bayesian learning can accommodate hypotheses which make probabilistic predictions e.g. this patient has a 93 % chance of recovery

d) New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities

Its main goal is to determine the most probable hypothesis, given the data D plus any initial knowledge about the prior probabilities of the various hypotheses in H.

The Bayesian Learning class of algorithms is more desirable in its application surveillance than the above two classes due to its involvement of probabilities and belief networks and can integrate both the Concept learning and decision tree learning algorithms better by using statistical methods which in some applications out perform their generic algorithms. Limitations of this Class of algorithms is that it is not an instance based algorithm as the computation is done only once and it requires initial probabilities to compute the overall probability of a hypothesis H. Also, the model for the Bayes optimal classifier,

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Equation 2

tries to classify a hypothesis based on a collection of the probabilities of other Hypothesis and-ranks the new instance hypothesis based on the majority classification that is either a positive or negative represented as V={+,-} which is not a desirable computation of this probability based likelihood value Research. This research is not supposed to classify a collected Shadow ID as either a positive identification or Negative identification but in a varying likelihood value of any particular Shadow ID at any particular instance of time. Finally the new value computed for all the new and existing Shadow IDs should vary at any particular instance while the Bayes Optimal Classifier only classifies new instances based on the classification of the already existing collected Shadow IDs which will lead to an erroneous result that does not reflect the desired hypothesis. *(Tom M. Mitchell (1997). Bayes Theorem and Concept Learning. In: McGraw-Hill, (ed). Machine Learning, Canada: McGraw-Hill, pp159-164)*

2.2.5 Artificial Neural Networks

This is the most common class of Machine Learning algorithms available. Artificial neural networks (ANNs) provide a general, practical method for learning real-valued, discrete-valued, and vector-valued functions from examples. It was developed as a representation of the way the representation of the real Brain, which is made up of a complex web of interconnected of neurons. In rough analogy, artificial neural networks are built out of a densely interconnected set of simple units, where each unit takes a number of real-valued inputs (possibly the outputs of other units) and produces a single real-valued output (which may become the input to many other units).

ANN learning is robust to errors in the training data and has been successfully applied to problems such as face recognition/detection, speech recognition, and learning robot control strategies. They are used to provide a robust approach to approximating real-valued, discreet-valued and vector-valued target functions, for certain types of problems such as learning to interpret complex real-world sensor data. Due to its effectiveness the ANN has become the most popular classifications of learning methods.

The development of ANN has been inspired in part by the observation that biological learning systems are built of very complex webs of interconnected set of simple units, where each unit takes a number of real-valued inputs (possibly the outputs of other units) and produces a single real-valued output(which may become the input to many other units). The BACKPROPAGATION algorithm is based on the combination of several perceptions which act as the unit nerve which takes a vector on inputs and produces a single output based on its threshold sigmoid function which is either a fire or miss.

2.2.5.1 **Instance Based Learning:** This is the approximation of the target function from the training examples, as the approximation process is repeated with each and every query. Each time a new instance is encountered, its relationship to the previously stored examples is examined to assign a target function value for the new instance. There are several algorithms which include the Locally Weighted Regression, Case Based Reasoning, and the one to observe, the K- Nearest Neighbor and the Radial Basis Functions.

2.2.5.1.1 **Radial basis functions** have over time been the most desirable class of algorithms used in the development of intelligent surveillance systems such as the Object tracking using Radial basis function networks surveillance system which by principle is the design of video surveillance system to tackle terrorism by tracking of moving objects of interest using video surveillance through inexpensive CCTV cameras. Due to its hybrid nature of the combination of several algorithms which reflect a similarity to the Bayes Optimal Classifier, K-NN, Back-Propagation it provides an adequate easy to adjust model that can use a combination of different sub models combined to compute the final output value which is in fact not on a fixed scale of valid outputs but a flexible range that can be used recursively to compute the next instance of the same data based on newly acquired input data

The Radial Basis Function is related to distance-weighted regression and ANNs and is used for function approximation which is more efficient than ANNs using Backpropagation.

In this approach, the learned hypothesis is a function of the form

$$\hat{f}(x) = w_0 + \sum_{u=1}^{k} w_u K_u(d(x_u, x))$$

Equation 3: Radial Basis Function Mathematical Model

- Where each x_u is an instance from x and where the kernel function *K,(d(x,, x))* is defined so that it decreases as the distance d *(x , , x)* increases.
- *K* is a user-provided constant that specifies the number of kernel functions to be included.
- *f (x)* is a global approximation to *f (x)* , the contribution from each of the $K_u(d(x_u, x))$ terms is localized to a region nearby the point x_u
- It is common to choose each function *K, (d* $(x_u, x))$ to be a Gaussian function centered at the point x_u with some variance σ_u^2 that has been shown to produce a small error.

$$K_u(d(x_u, x)) = e^{\frac{1}{2\sigma_u^2} d^2(x_u, x)}$$

Equation 4 Radial Basis function Error function

The functional form of Equation can approximate any function with arbitrarily small error, provided a sufficiently large number k of such Gaussian kernels and provided the width σ^2 of each kernel can be separately specified.

The function given by Equation (Equation 4 :) can be viewed as describing a two layer network where the first layer of units computes the values of the various *K,(d*$(x_u, x))$ and where the second layer computes a linear combination of these first-layer unit values. An example radial basis function (RBF) network is illustrated in the figure below.

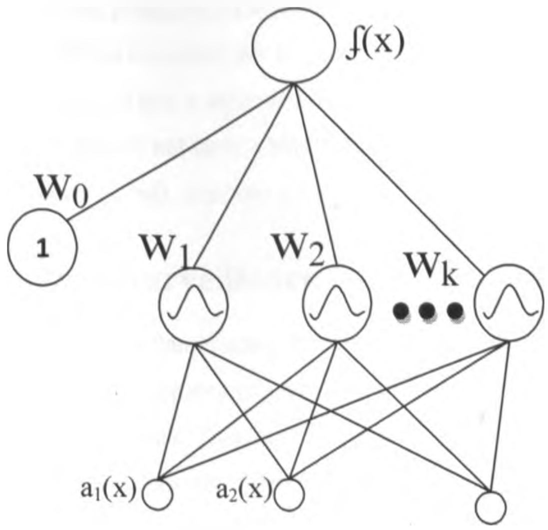


Figure 2: Radial Basis Function Network representation from the book Machine Learning by T Mitchel

Given a set of training examples of the target function, RBF networks are typically trained in a two-stage process. First, the number k of hidden units is determined and each hidden unit u is defined by choosing the values of x_u and σ_u^2 that define its kernel function $K(d(x_u, x))$. Second, the weights w_u are trained to maximize the fit of the network to the training data, using the global error criterion given by Equation (Equation 4). Because the kernel functions are held fixed during this second stage, the linear weight values w_u can be trained very efficiently. This principle has proved to be very useful in calculating the likelihood associations of entity-event based patterns as it is a combination of several different functions which are collectively summed to provide a single likelihood value of an entity based on not a rigid and fixed rule of computation (which is the limitations of other Classes of Machine learning algorithms such as the Concept Learning and the Decision Tree Learning). As it describes the general rule to which different models of distance computation $K(d(x_u, x))$ are summed up (represented by $\sum$) For each and every instance x_i based on multiple local approximations on an n-dimensional space where each n represents an individual attribute about the person or entity that is relevant to the event of interest. The Radial basis network diagram (Figure 3) shows that the computation of the likelihood value can be applied to any instance based on all the training examples at any particular instance while adjusting each attribute-entity-event likelihood value at any particular instance based on new data. This is important when trying to map the entity to an event of interest as the main goal will to use independent models for calculating the distance between entities based on a particular attribute at any instance and finally combining all the values gotten from all the models at that particular instance.

Due to the widespread use of mobile technologies, it has become easier to individuals and communicates over long distances changing the way people interact and work together. This has also complicated and made it hard to identify behavioral patterns and interactions between people as they have adjusted their activity patterns in performing various team-work related tasks by adopting technology to their day-to-day and strategic activities.

By the use of mobile technology and Artificial intelligence, voluminous data that consist of both tracking or geo-location information and call logs can be analyzed to generate valuable information that hasn't previously been identified by other forms of surveillance technology such as CCTV or described before. The future of surveillance should not depend only on the physical technology alone but also in the processes of information gathering and analysis. Frameworks define how operations will be conducted using the current resources for better results, frameworks not only involve the physical procedures but also computation processes and this is achieved better by the use of software technologies developed using AI principles to solve complex problems that do not have a single procedure of solving them. *(Tom M. Mitchell (1997). Artificial Neural Networks. In: McGraw-Hill, (ed). Machine Learning, Canada: McGraw-Hill, pp230-240)*

2.3Recent Developments in Surveillance using Machine Learning

Several developments have been made on surveillance using machine learning algorithms which include object tracking for video based surveillance technology; this is the implementation of an intelligent system that can track moving objects and people within an area as opposed to motion sensors. This technology identifies the object of interest and tries to trace its movements by following the target object in successive video frames. The object is selected, and a white rectangle then marks the object domain. Another box is marked around the first one with surrounding region has equal number of pixels, which is used as the object background. The object and background are separated from each other. The R-G-B based joint probability density function (pdf) of the object region and that of the background region is obtained. The region within the marked region is used to obtain the object pdf and using the marked background region the background pdf is obtained. The Log-likelihood of a pixel considered in the object and background region is obtained as

$$L_i = \log(\max\{h_o(i), \varepsilon\} / \max\{h_b(i), \varepsilon\})$$

Equation 5

Where h_o and h_b are the probabilities of the ith pixel belonging to the object or the background respectively, and ε is small non-zero value to avoid numerical instability. A binary image is then constructed by giving a threshold for which a particular pixel is considered to be on object or in the background.

Though this technology serves as a good tool for movement tracking, its only limited to a minimal specific area of coverage that has the CCTV cameras installed. This limits the surveillance coverage area to a small distance and is both expensive to install and isn't effective on blind spots (Areas where the CCTV cameras aren't viewing). Also it is only used for the tracking of a single object movement.

Qi Zang and Reinhard Klette have designed a video surveillance system is directed on automatic identification of events of interest, especially on tracking and classification of moving vehicles or pedestrians using background subtraction to segment the moving objects. Each background pixel is modeled using a mixture of Gaussian distributions.

The Gaussians are evaluated using a simple heuristic to hypothesize which are most likely to be part of the "background process". Each pixel is modeled by a mixture of K Gaussians as stated in formula

$$P(X_t) = \sum_{i=1}^{k} w_{i,t}\eta(X_t; \mu_{i,t}\Sigma_{i,t})$$

Equation 6

Where:

- Xt is the variable, which represents the pixel, and t represents time.

- K is the number of distributions: normally we choose K between 3 to 5.
- $W_{i,t}$ is an estimate of the weight of the *i*th Gaussian in the mixture at time t, $\mu_{i,t}$ is the mean value of the *i*th Gaussian in the mixture at time t. $\sum_{i,t}$ is the covariance matrix of the ith Gaussian in the mixture at time t.
- X_t is checked against the existing K Gaussian distributions until a match is found. Based on the matching results, the background is updated as follows:
- X_t matches component *i*, that is X_t decreases by 2.5 standard deviations of the distribution, then the parameters of the *i*th component are updated as follows:

- $W_{i,t}=(1=\alpha)\ W_{i,t-1}+\alpha$

Equation 7

- $\mu_{i,t}=(1-\rho)\ \mu_{i,t-1}+\rho I_t$

Equation 8

- $\sigma_{i,t}^{2}=(1-\rho)\ \sigma_{i,t-1}^{2}+\rho(I_{t-}\mu_{i,t})^{T}(I_{t-}\mu_{i,t})$

Equation 9

Where $\rho = \alpha \Pr(I_t| \mu_{i,t-1},\sum_{i,t-1})$ α is the predefined learning parameter, μ_t is the mean value of the pixel at time t, and I_t is the recent pixel at time t.

It provides a similar framework of computation but instead focuses on images especially the separation of changing pixels of a moving object from its background and this is purely a video based object tracking algorithm implementation of a Radial Basis Learning Rule principle for video based object tracking surveillance.

Neill and Cooper (2010) extend a univariate Bayesian detection framework into a principled multivariate Bayesian approach that integrates prior domain knowledge for a highly powerful detector of emerging patterns. Their multivariate Bayesian scan statistic (MBSS) approach stands out because of its flexibility and applicability to a wide range of multivariate detection problems. The approach has yet to be implemented to the identification of entities associated to the event as it already assumes all the entities have been identified. Furthermore it hasn't yet been applied to the already existing radio based technologies such as mobile telecommunication networks for surveillance.

Singliar and Hauskrecht (2010) defined detection requirements, developed, and analyzed over different parameter ranges, a series of detectors for traffic incidents. They specifically address two of the most important aspects of handling of streaming data that is collected from sensors: data affected by noise and data that is not aligned. Their learning of the Tree-Augmented Naïve Bayes (TAN) approach addresses the alignment issue in a very elegant manner, and clearly improves detection. This approach only focuses on traffic and the observation of change in object movement ratios.

In Anders Jonsson, et al paper they present a novel application for interactive browsing of (recorded) surveillance content. The application is based on user feedback and enables an operator to switch between camera views that are likely to contain the same activity.

Their system relies on off-the-shelf background-subtraction activity detection mechanisms. They use two techniques from machine learning to automatically learn the topology of surveillance camera networks.

The first technique identifies connections between camera views for which objects are temporarily out of view, while the second technique identifies overlap between views. To learn the camera network topology, they use techniques from machine learning. For camera views such that objects are temporarily out of view, they establish temporal correlations between activities of objects transiting between views using clustering and cross-correlation.

For views such that objects appear simultaneously, they use a technique based on exclusion count to detect overlap. Neither approaches rely on established correspondences between trajectories, and are thus completely unsupervised. They show how to adapt both techniques to the surveillance scenario, in which the amount of detected activity is usually very small.

The algorithm they employed consists of two components: video segmentation and tracking. The first component, video segmentation, detects the image areas in which moving objects are located. The component maintains and updates dynamic background estimation in the form of a mixture of Gaussians for each pixel luminance. This approach makes it possible to represent regularly oscillating or blinking objects as part of the background estimation. In each new frame, image areas that differ from the background are tagged as moving objects. They use rectangular bounding boxes to represent objects in each frame.

The second component, tracking, constructs coherent trajectories from the video segmentation. Tracks are simply sequences of bounding boxes across image frames. For each detected object, they compare its position with the last position of all active tracks. In case the object fits one track, that track is updated with the object description. In case no track fits the object position, a new track is initiated. They apply a filtering process to remove tracks whose duration is too short or whose behavior is not smooth. The difference between theirs and the proposed framework is that they have developed a surveillance strategy for Video based surveillance.

An automated methodology for extracting the spatiotemporal activity model of a person using a wireless sensor network deployed inside a home. The main principle of this approach is by sensing the person's movement across different rooms over a period of time e.g. 30 days while observing the time frequency of specific states i.e. awake, asleep, in the toilet over a period of time to map out the persons general baseline activity patterns for better prediction and placement of items such as medicine, food the elderly patient might need at any particular time.

The data considered for this work comes from an ongoing sensor network deployment that monitors an elder living alone. The test bed includes a wide variety of sensors including tracking cameras, door sensors and passive infrared sensors. Only PIR (Passive Infra Red Sensor) measurements are used for this model. Every room in the house contains PIR sensors placed in a pattern that can capture the elder's transitions from room to room. Each time a sensor gets triggered it transmits its ID to a home gateway that timestamps and records the sensor ID and uses the data to compute a room-transition function. Based on our deployment, the network generates a sequence of triplets of the form: {P, T, D} where: P is the phoneme detected by the sensor node (a room identifier for this discussion), T is the actual timestamp at which this phoneme was detected and D is the duration of the phoneme. According to the above definition, the output of sensor node i over time will be a time ordered sequence of triplets Si. Assuming that in a given time frame, sensor node i has generated Ni triplets, its output can be denoted as follows:

$$S_i=<\{P_1^i,T_1^i,D_1^i\},\{P_2^i,T_2^i,D_2^i\},\ldots,\{P_{Ni}^i,T_{Ni}^i,D_{Ni}^i\}>T_1^i<T_2^i<T_{Ni}^i$$

Equation10

Consequently, the output O over time of a sensor network with n nodes becomes a collection of such time ordered sequences of triplets:

$$O = \{S^{1,} S^2,.., S^n\}, \ |S^1|= N^1| S^2|=N_2,\dots\dots\dots, |S^n|=N_n$$

Equation11

Where Si is the time ordered outputs sequence at node i, containing Ni triplets.

The difference with this study is that it only focuses on a small scale environment of observation and a single party of observation, but is quite useful in identifying human Spatiotemporal Activity Patterns using radio based sensor technologies.

Due to the differences in the above research approaches. The main focus of this research is to develop a framework for use in radio based networks specifically mobile Telecommunications networks service providers for the identification of entities that are likely to be associated to a Person of Interest or an Event of Interest by observing the movement activity patterns of the observed populations, physical proximity between the observed parties, contact or communications between the Person of Interest or other identified people and their frequency of appearance and/or contact.

CHAPTER 3: METHODOLOGY

3.0 Introduction

The Surveillance framework is strategy is supposed to solve the problem of computing likelihoods of association of individuals or groups based on events or other individuals. It is also supposed to provide an automation process for this computation using known machine learning and data mining techniques that have successfully been applied to other problems of similar nature such as the prediction of probability of default via credit scoring, Object tracking for security surveillance www.facebooks.com "whom you may know" likelihood algorithms is suspected to use a similar principle in ranking possible friends (cannot be confirmed).

The design of this prototype to reflect the system framework has been done in layers and components which have been designed individually and developed independently to reflect the independent framework sub processes. Each subcomponent takes a collection of different discreet valued, real valued, inputs and produce real valued outputs that are at times be the input values of other or the same components based on the instance of computation.

The several individual components have been designed so as compute the likelihood values of all the collected IDs based on a single formula or model and their relationship or comparison levels between the POI, Shadow ID and other Shadow IDs have been represented by the use of histograms as this is the best presentation format to explicate differences between individual entities based on any real valued model.

Due to the fact that it is nearly impossible to acquire the actual data from the mobile telecommunication companies the data to be used has been generated in the lab that reflect real life situations that involve multiple parties. Though this method, its accuracy in reflecting actual real life situations has been limited to the accuracy and details of the data collected out of the available noted events.

The data generated includes the calls made by/to the party of interest. This has proved important as it acts as a factor for determining the level of association between the Person of interest and the identified caller or receiver. The level of association will be assigned a value using a hierarchy represented by a network tree whereby any caller ID's that has made direct access to the person of interest has a higher ranking value than those who have had an indirect association to the POI based either on the interactions with callers that have had direct access to the POI. The first association computation has been represented as illustrated in the diagram below.

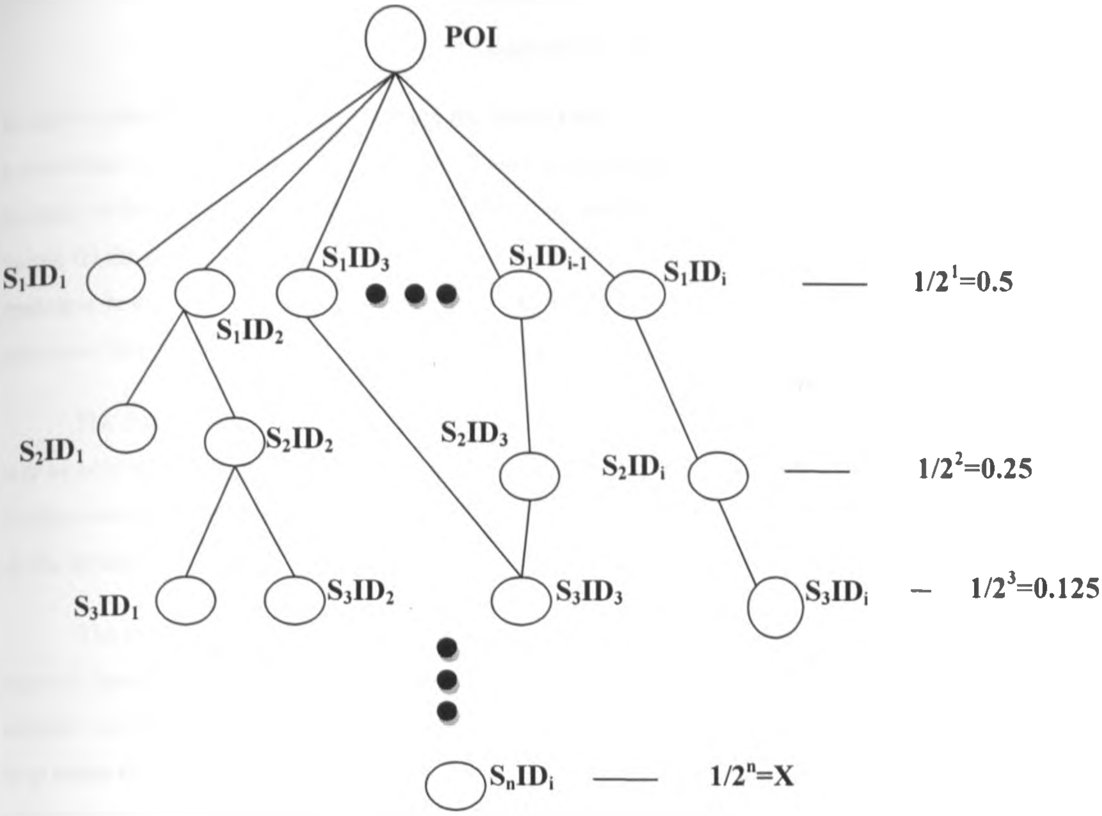


Figure 2 Level of Association Computation Network developed during this research

The above network Diagram shows how each collected Shadow ID Level of Association value from the POI has been computed where if a particular ID e.g. SID_x knows SID_y who in turns knows or has directly called or was near the POI, the level will be 2 and the Value of association of SIDx to the POI at that particular instance is,

POI ----- $(1/2^1) = 0.5$ --→ SID_y ------ $(1/2^2)0.25$--→ SID_x

$1/2^2=0.25$. This is because the Level of association rule is guided by the valuation rule

$$X=1/2^n$$

Equation 12: X value representation

Where X is the value to be computed at that particular instance of that specified dimension attribute and n is the level of association, this means that K,*(d(x,, x))=X* which in this case are both the direct and indirect calls and contact made to and from the POI to the particular ShadowID. And the physical proximity from thePOI at any particular instance. But the overal X value is the summation of all the association values of proximity to the POI since multipe Shadow IDs can be associated to each other where all the different assosiations from the POI (only from parent to child) have been considered forming the overal rule to be

$$f(X_i)=\sum_{i=0}^{\infty}(1/2^n)$$

Equation 13: General Ranking Function

The above rules have been used to measure the direct likelihood of association between any Shadow ID and the POI while the event based associations which include a filtering rule which eliminates all Shadow IDs that have not been in close proximity to the POI at the Events particular time of occurrence. The more frequent the calls from either the POI to a Shadow ID the greater the ranking value of association. This also applies to the rule that the greater the frequency of appearance in close proximity of a Shadow ID to the POI the greater the ranking value of that Shadow ID. These rules have been turned into system modules that compute the ranking value of any shadow ID from the records generated.

The final output is be a list of numbers in descending order where the ID or individual with the combined most likely association value to a particular event is displayed as the first record on the list of entities and the person or individual with the overall least possible likelihood to be associated to the POI or event that is above the threshold of likelihood value is at the bottom of the list.

The validity of this framework is based on the observation of its ranking order of the Shadow IDs that has been given to it based on a particular POI and event Associated with the POI. Some of the Shadow IDs information generated is consistent to a particular hypothetical event or POI. The system prototype computes the levels of association of the Shadow IDs to either the POI or Event (based on the time of the event) on which the systems variability is measured in terms of the total number of Shadow IDs it was able to rank highly to either the POI or the Event out of the actual hypothetical Shadow IDs that were set to be associated to the particular event or POI.

3.1 Framework Validity Computation

In percentage of viability is measured as the accuracy of the system producing the same results and in the same order to the generated case scenarios validation data as described below.

a. (Total SIDs identified by the system/the total actual SIDs that was set to be associated)*100
b. (the total actual SIDs that was set to be associated /the number of SIDs that were not involved within the listing)*100
c. (a+b)/2

The main achievement of this project framework is the separation of the stages of development into two subsections.

1. The development of an association model that is used to compute the value of the possible associations between two or more parties based on the hierarchy of interaction from the POI to the Shadow or Collected ID (from the Shadow ID's call logs) to the frequency of contact between the two based on the observed Time of interest.
2. The development of an association model that is used to compute the likelihood value of the possible associations between two or more possible parties based on the physical proximity from the POI and the Shadow ID(s).

3.1.1 Framework operational Stages

3.1.1.1 Step 1: Collection of Active MS data

3.1.1.1.1 The Identification of the target MS and Relevant Attribute extraction:

This has been achieved by first Identification of the Targets Person of Interest's MS that have been provided to the relevant authorities as they include

- Phone Number
- Phone IMEI
- Time stamp of the data collection
- The MS of the POI location or Co-ordinates

3.1.1.1.2 Collection of the Shadowing MS or POI within a set distance radius from the target MS.

The results collected from this are the range of all Shadow MS that have been collected minus the filtered out Shadow MS, their Locations plus their timestamps.

For the purpose of this research it is assumed that the first of the two main phases of this framework that has been applied in the collection of the relevant MS that have been used in the collection for the final association computation model to be used during the project phase.

It is assumed that this number and the details will have already been collected.

3.1.1.2 Analysis and Entity to Entity to Entity to Event Based Ranking

The second phase is the analysis of the collected data results from the initial stage. Unlike the first phase the results of the second stage final model is based on the previous model results plus the computation of the same model based on previously unused data.

The research framework comes into operation in this stage as the final Rankings computation is done in this stage.

During this phase: the final model framework is achieved by:

1. Identification of all the attributes needed that might be of use to the observer to positively identify the possible devices that the Target Individual might use in his/her operations.
2. Identification of all the possible factors needed to associate the entity to an event or his/her relationship between two or more entities.
3. Determination of the relationship between the attributes and the possible likelihood of association between the observed MS and the subject of interest.
4. Determination of the relationships between the attributes and other attributes.
5. Quantification of the attributes for the development of the final model as variables of a mathematical function.
6. Testing of the final model for its viability towards the stated objective.

The system framework prototype reflects the diagram (Figure 2).

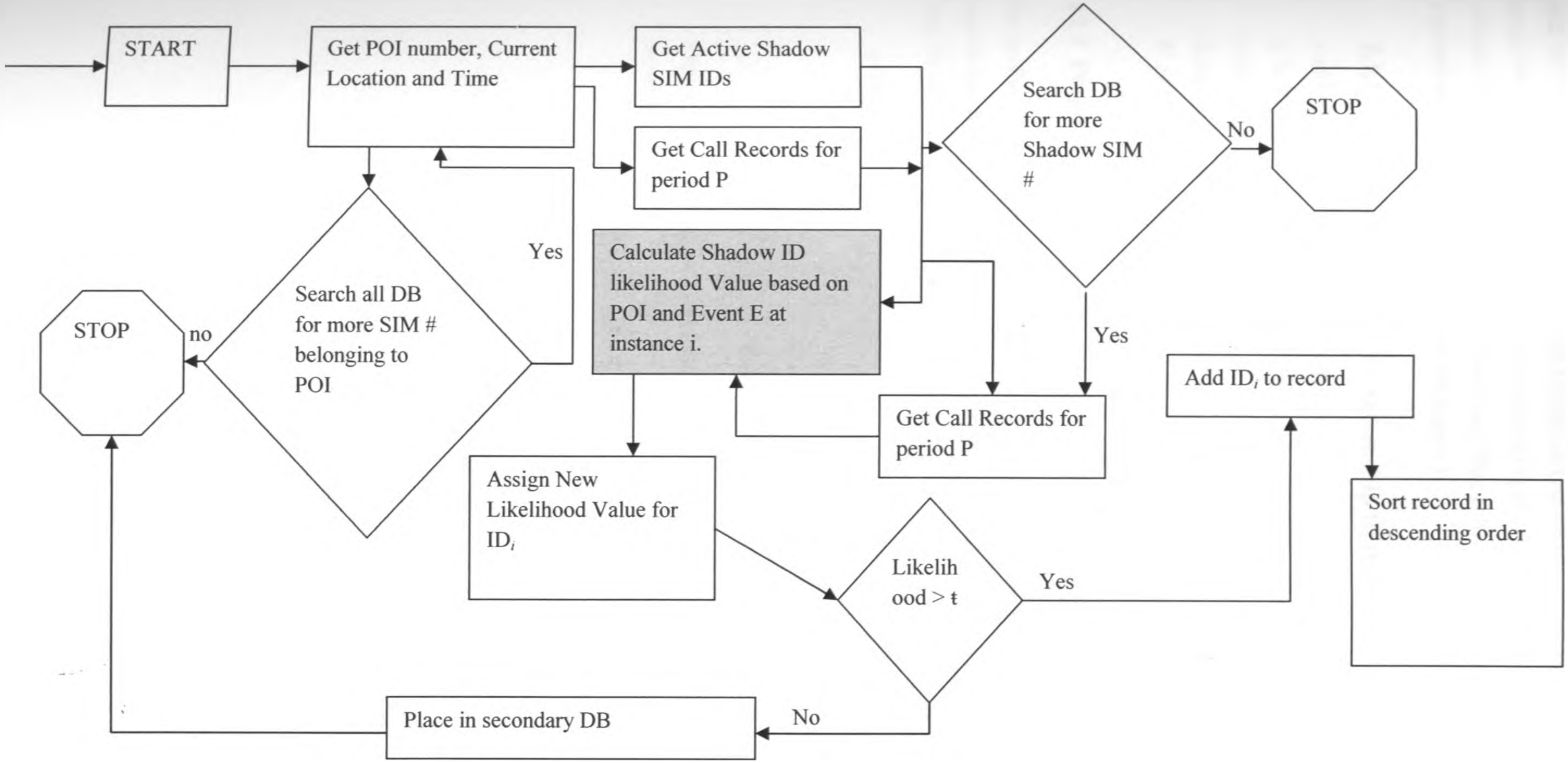


Figure 3 the overall system framework model

Using the Radial Based network functions for the Current GSM network tracking technology structure, It can provide a strategy for automating the computation for the ranking of collected numbers based on Several independent and dependent attributes which require separate computations and their collected sum to finally compute the overall association value of the several individuals to events as shown (in Figure 3).

3.1.1.3 The proposed model relationship to the Figure2 model represented as Figure 3.

4 x will represent the ID from the Shadow Mobile number collected during the observation period.

5 x_u is the instance at which the ID's association value is computed at any one time considering a particular attribute of association.

6 $K,(d(x,, x))$ represents the isolated relation computation of the association value based on a single collected attribute.

7 W is the current weight computed for each of the collected numbers before, during and after the onservation period.

3.1.2.4 Proposed Output

After the final computation of the final instance value $f(x)$ of the collected phone numbers and ID tag information using the proposed model. The results is a list of filtered numbers with the highest association values from all the collected numbers during the observation period above the threshold Value.

t=Threshold

$$\{ x_1, x_N \} \mid f(x_i) > t$$

Equation 14: Threshold model representation

Figure 3 repesents the gaphical format of the flow of computation which applies the above mathematical principle(Equation 14).

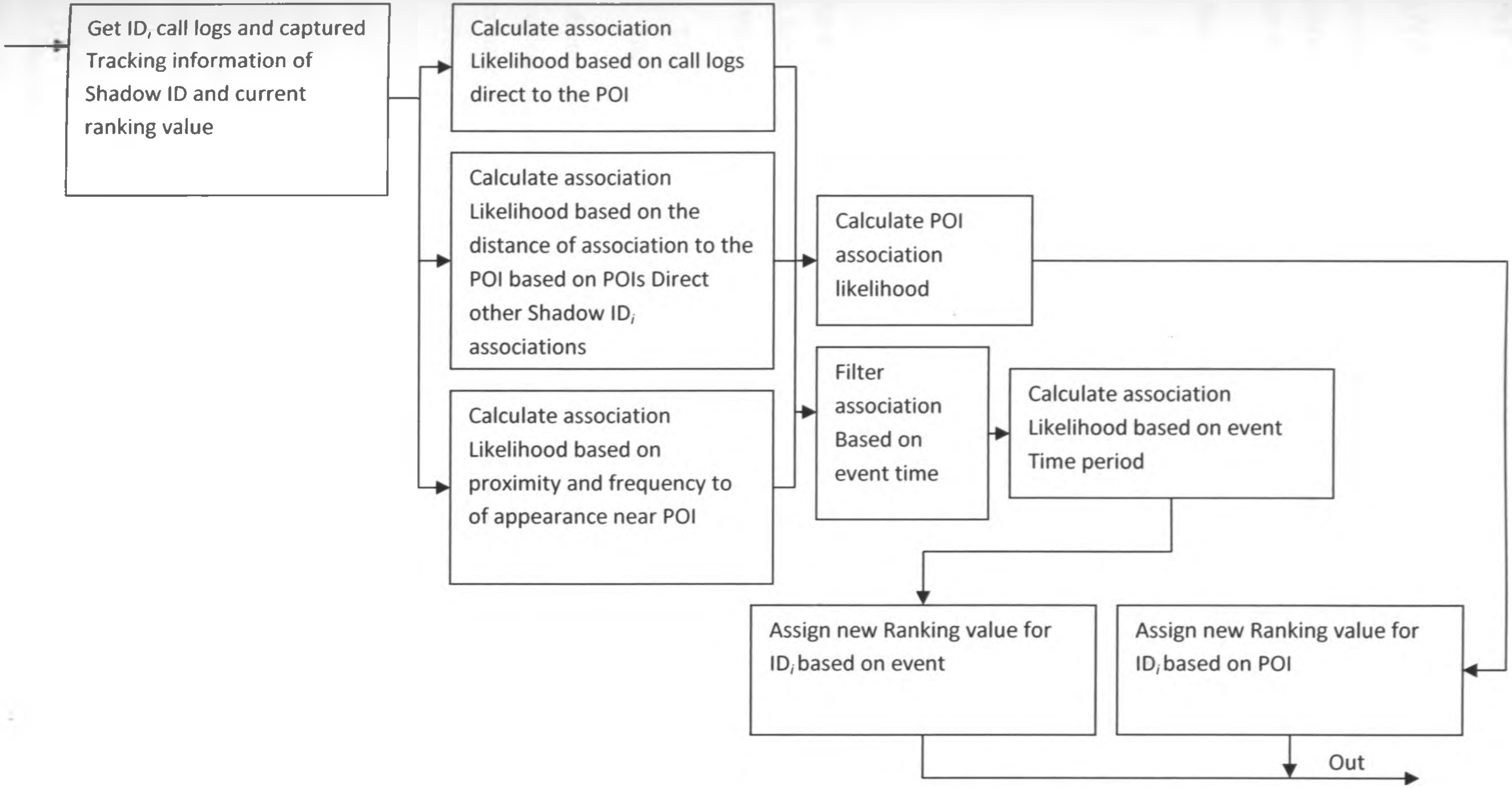


Figure 4 Core system framework Diagram

CHAPTER 4:

4.1 ANALYSIS

Based on the area of focus *Surveillance* and the problem of interest Entity – Entity Social possible Relationships identification and Entity – Event possible Involvement identification the Following statement has been formulated that describes the main focus of this particular research.

4.1.1 Rules guiding the identification of Events

Rules observed during the Research and framework Development Life Cycle:

1. Event **E** is defined as an occurrence as a result of Human **P** activity over a period of time **T** at Location **L.**
2. For an Event **E** to occur there has to be a Location $\mathbf{L_i}$ or group of locations $\mathbf{L_i},\ldots,\mathbf{L_n}$ where the Event **E** occurs.
3. For Event **E** to occur there must be the Time $\mathbf{T_i},\ldots,\mathbf{T_n}$ to which Event **E** occurs where ***i*** represents the number of Times and or the duration to which **E** occurred.
4. For Event **E** to occur it must involve Entity **P** or Groups of Entities $\mathbf{P_i},\ldots,\mathbf{P_n}$.

From the above defined statements it becomes clear that the Event of interest in this Research is one that strictly focuses on Human Activity based occurrences over a period of time and it has been the main Guideline to the research and has greatly guided the Development of the Framework Model and subsequently the prototype described as follows.

1. The Problem
2. The Solution.

The problem is the Current status and limitations that the desired state needs to improve or eliminate. In this research the main problem of interest is the unavailability of social Relationship and possible event association likelihood computing methods that can enable the interested party in this case the National Security Agencies find the list of possible associations between People either as a social Relationship or based on particular events using the currently available Mobile telecommunications technology in which is accessible to the majority of the Adult population within a specific country or physical domain.

The desired state is the availability of a framework that can use readily available data from the readily available technology to compute the possible social relationships between individuals or groups and individuals associations to specific events with a desirable justifiable and reproducible level of accuracy based on the human activity patterns and communication habits between each other.

The process is to study various events, identify the different conditions for the various events, factors involved, how the factors relate to each other to influence conditions, quantify the factors into variables, Develop a Mathematical Model that uses the Variables to produce the end result and the desired framework strategy, then develop an automated system prototype based on the model Framework developed, feed in data and compute the results to solve the problem identified by this research.

Variables and conditions of interest should be able to affect the final outcome of the results and also reflect the true situation on the ground.

The variables relationship between other variables and conditions should and has been mapped individually to the end result.

The final outcome Result should in itself be a quantifiable Variable that should be dependent to all the other variables of focus.

The process of computing the end result should not only be based on the designed framework but should be automatable by the use of a software system.

4.2Design

The System Framework is supposed to use a Mathematical model based on the Radial Basis Function principle. The main input or data required for the system framework is of two kinds. The Call logs and coverage Area active MS data, both Criteria of data can be collected independently computed independently but for the purpose of the developed framework they both contribute in unequal measures to provide the final value which is the association Likelihood value to the collected ShadowIDs in relation to either a particular Entity or event of interest.

In general the framework developed should compute the results of both the call logs and proximity logs independently and compound the result into one final result value for each and every entity that has been observed during the collection of the data.

4.2.1 The Computation Principle

Several models have been observed to determine the most suitable model applicable to compute the end result and reflect the application process of interest.

This research and framework development has been greatly influenced and guided by the Event Conditions which include a Primary set of Variables of interest as shown below:

1. **Time:** The time in which the active MS was observed during the data collection period, this included the time the ShadowID was observed to be either in near proximity to the POI or in contact with the POI.
2. **Location:**_ This is the specific location which the ShadowID was observed to be at a particular instance of time during the data collection process.
3. **ShadowID:** Parties involved, in this case they are all the ShadowIDs observed during this process. This includes the Person of interest and all other ShadowIDs observed during the Data collection period. It also includes the contacts that were identified from the call logs.

The above are the three most important factors that have been taken into consideration when building the system Framework. Due to their independence and differing Weights contribution toward the final outcome, some of the Variables have been combined to form new Secondary Variables that have been used in the designed model to compute the final outcome. The development of the new variables is a combination of two or more Primary set Variables as detailed below:

1. **Duration:** This is the time range $\{\Delta t = t_2 - t_1 \mid t_2 \geq t_1\}$ that will be selected by the party to determine the duration of the event, although it isn't directly associated to the Framework model, it is used as a filtering mechanism that will be used to select only the records that conform to the set time conditions Duration can be represented as a time range of a date range. But in call logs Rankings Duration D is the duration in which the POI communicated with the observed shadow id.
2. **Run_Level:**_ This is the Depth of computation to which is used to reveal ShadowIDs that might be associated to the POI but were not detected during the first Pass. In this case the Run level is computed as the iteration of the computation loop, where if the primary POI is SID1 and after the first pass the collected ShadowID list contains SID2,SID3....SIDN, The Run_Level will be increased by one after each time the set POI moves from SID1 to SID2 to SID3 and so forth.
3. **Number of Appearance:**_ This is the number of times the ShadowID_X has been identified, it is usually as a result of adding all the number of times a particular shadow ID has been identified when each run is being computed, although it is independent to the Run_Level due to the simple fact that the depth of the Levels of association do not affect the value for the number of appearance, it does affect the final Ranking outcome.

All the above have been used in varying weights to compute the final Ranking outcomes as their relationship to the Final Ranking which is represented as $\mathbf{R_{SIDi}}$ where **R** is the Final Rank Value and $\mathbf{SID_i}$ is the particular individual ShadowID_i

- As Duration value D_i increases so does the Call Rankings and subsequently $\mathbf{R_{SIDi}}$.
- As Number of Appearance $\mathbf{P_i}$ increased so does the Likelihood Rankings $\mathbf{R_{SIDi}}$.
- As the Run_Level value increases, the Likelihood Rankings $\mathbf{R_{SIDi}}$ decrease.

The above three points were used to guide the formulation of the model:

The above three points were used to guide the formulation of the model:

$$R\_SID_i = \sum_{i=0}^{\infty}\left[\left(\frac{1}{2^{n_{pi}}}\right) + \left(\left(\frac{1}{2^{n_{ci}}}\right) + D_i\right)\right]$$

Equation 14: Main Entity and Event Ranking Model

Where:

- R_SID_i is the overall Rank Value for ShadowID_i.
- i is the instance to which that particular Process is at a point in time in this case the number of appearance is also represented as *i*.
- *n is* the Run_Level for Proximity Rankings ***pi*** and for the Call Logs Rankings.

- D_i is a function of the Call Duration Interval and Rankings represented by the formula $\left[\frac{(d+Ci)}{\left(\frac{d+Ci}{Ci}\right)}\right]$ where d is the actual duration of the call and C_i is the overall Call Rank value up to that particular instance i which is being made between the POI and ShadowID

The above model shows that the final Rank value for any particular ShadowID association computation is as a result of finding every individual ShadowID based on the Level it appears at each level to which the user can specify and summing up all the instance computations gotten by for that particular ShadowID.

It also shows that the lower the Level (the Higher the Run level value) from the POI, The lesser the final Rankings of that particular ShadowID. This is suitable when the computation should distribute the Levels of association into an equally distributed separation that can be self divisible and also divisible by the Levels without resulting in irrational numbers which creates a problem when trying to compute the number of occurrences based on both the initial ShadowID Rank Level and Number of appearances.

It also shows that the model is strictly based in the iterative recompilations of the same model value whenever the required conditions are met.

The Machine Learning principle Borrowed by this Model represented below is as follows:

$$\hat{f}(x) = w_0 + \sum_{u=1}^{k} w_u K_u\big(d(x_u, x)\big)$$

Equation 15 Radial Basis Function General Learning Rule

- $f(x)$ is represented as R_SID_i
- $\sum_{u=1}^{k} wu(d(xu, x))$ is represented as $\sum_{i=0}^{\infty} Ri\,(1/2(ni, n))$
- k_u is not considered in this model as that is its main Variation between the Radial Based Function and the proposed framework model.

Reason for ignoring the k_u constant multiplier is because of the intended result will be dependent of the result value of the main POI as the Likelihood association will be in relation to the final computation of The POI's Rank Value computed based on the number of appearance and timing on the identified activities. And since the Radial Based Model is not strict on which specific operations are to be applied so does it give desirable flexibility in its specific application.

CHAPTER 5: PROTOTYPE IMPLEMENTATION

5.1 Introduction

The implementation of this framework has been done by the development of a prototype using the Java JDK1.6 Object Oriented Language.

The reason of this language selection is due to its Robustness, Object Oriented implementation principle and its platform independence not forgetting my Familiarity with the Programming Language.

The Framework system Breakdown involves the Definition of several classes and Functions that have been used to perform individual tasks that in all result to the implementation of the System framework generally the system is supposed to have several components working together to solve a particular sub-problem in which combined will result in to a system that reflects the Framework which will ultimately solve the entire problem at hand.

The main Strategy of focus will be to develop an Automatable Framework that will enable the interested party to compute the likelihoods of people being associated to each other based on Events and also being associated with Events of interest based on their Interaction with each other and their physical proximity to each other at a particular time using the Mobile phone technology or any other two-way radio based technology.

5.2 Considerations

When designing the system prototype several factors were taken into consideration which includes:

- Event computations were dependent on the Location(s) time(s).
- Association computations were dependent on the events related to a selected particular POI.
- Entity Likelihood associations are dependent on a Selected ShadowID which becomes the POI.

5.3 Inputs

The required inputs were data from a database in the form of two separate tables which as described earlier they are the:

1. *Proximity Logs:* this was a list of collected ShadowIDs Active at a particular time instance and Location.
2. *Call logs:* This is the list of all collected Communications between every entity that made or received to and from each other.

The data collected in the Proximity log table include:

1. *ID:* The record ID (Integer and auto increments as data in the table is increased). It's not of much use for the sake of the computation process.
2. *POI:* This is the Person of Interest (String and not unique). It was the initial focus for the Ranking computation but subsequently became redundant when mass undirected Active ShadowID computation was performed.

3. *ShadowID:* (string) this is the particular Active MS that was picked up from a particular location at a particular time. It is the most important field as all the computations and Ranking is based on this Entity.
4. *Time of Observation:* (Time) this is the time the Collection of the Active MS were picked up by the parties. This field is needed in the event specific Ranking computation filtration process.
5. *Date:* (Date) this is the date the Collection of the Active MS were picked up by the parties. This field is needed in the event specific Ranking computation filtration process.
6. *Location:* (String), this is the actual location where the Active MS was detected at that particular instance in time. This field is needed in the event specific Ranking computation filtration process.
7. *IMEI:* (String) The phone IMEI number though not needed in this framework at this time.

The data collected in the Call log table include:

1. *ID:* The record ID (Integer and auto increments as data in the table is increased). It's not of much use for the sake of the computation process.
2. *POI:* This is the Person Of Interest(String and not unique). It was the initial focus for the Ranking computation but subsequently became redundant when mass undirected Active ShadowID computation was performed.
3. *Contact:* (string) This is the particular Active MS that was picked up from a particular location at a particular time. It is the Most important field as all the computations and Ranking is based on this Entity.
4. *Time of Contact:* (Time) this is the time the Collection of the Active MS were picked up by the parties.
5. *Date:* (Date) this is the date the Collection of the Active MS were picked up by the parties.
6. *Duration:* (Double) this is the actual length of time of communication between the POI and the Contact. It is required in the final computation of the ShadowID Rankings.
7. *Location:* (String), this is the actual location where the Active MS was detected at that particular instance in time.

5.4 Design

5.4.1 Functions

The following functions were used in the Entity Based Rankings and Event Based Rankings between the POI and all the other SahdowIDs.

1. *Entity Based Rankings:*
   a) *ComputeEntityRankings():* This is the Main ShadowID to POI ranking Function as it performs the Rankings computation based on the Model Described earlier, it has only two inputs. The POI and The Run Level,
2. *Event Based Rankings:*
   b) *Compute_TimeBasedRankings():* this function is used to compute the Likelihood Ranking associations of the ShadowIDs in relation to the POI based on time ranges irrespective of the date, Its inputs include the POI, the event start time, Event end time and the Run Level.

c) *Compute_DateBasedRankings():*this function is used to compute the Likelihood Ranking associations of the ShadowIDs in relation to the POI, Its inputs include the POI, the event start date, Event end date and the Run Level.

d) *GetValuesLocationBased():* this function is used to compute the Likelihood Ranking associations of the ShadowIDs in relation to the POI based on the Locations from which they appeared as active, Its inputs include the POI, the Location and the Run Level.

e) *GetValuesAll():* this function is used to compute the Likelihood Ranking associations of the ShadowIDs in relation to the POI based on the Locations from which they appeared as active, Its inputs include the POI, the Location and the Run Level. The main difference between the *GetValuesAll()* and the *GetValuesLocationBased()* function is that the *GetValuesAll()* also strictly considers the time to which the particular POI selected was at a particular location and also focuses on that time too.

f) *Sort():* This function is used to sort all the ShadowIDs in desending order based on their total Rankings.

All the above stated functions have been uniquely designed to consider the different variations between events of interest.

5.4.2 Function Results

All the above functions have a common output format which is a list of all possibly related or associated ShadowIDs that were identified and computed to have possibly been involved with the POI either as a social association or based on a specified event of interest.

The output is a list of ShadowID objects that contains Fields that hold the various valued of the different computations performed on them which is as stated below:

ShadowID Fields and Values

1. *ShadowID_Name(String):* this is the name or ID of that particular shadow ID.
2. *ShadowID_ Rank_Value(Double):* This is the final Ranking value of that particular ShadowID based on either the POI only or both the POI and Event strictly. In the Model above this is represented as R_SID_i .
3. *ShadowID_Proximity_Rankings(Double):* this is the Ranking value of the ShadowID based on the Proximity table data.
4. *ShadowID_Call_Rankings(Double):* This is the Ranking Value of the ShadowID based on the Call Logs table data.
5. *ShadowID_Rank_Level(Integer):* This is the Highest Level in which the ShadowID was identified. In this computation it is represented as the Run_Level.
6. *ShadowID_Total_Rankings(Integer):* This is the total Number of times the ShadowID has been detected or Identified during the computation process.

5.5 Function Components of the System

The functional components of the system will include:

a) **Select_Criteria() Function**

- This Function will collect the user specified Filtering Criterion required to compute the Likelihood association Rankings of ShadowIDs to the Person of interest also known in this case as the POI.
- The user specified input Criterion will be based on either of the two Rakings types which are the
    - Entity Association Based Rankings
    - Event Association Based Rankings
- The Variable of interest during this Criterion selection process will be the Time range, Location or even both if convenient.

b) Establish_Connection() Function

- This Function is used to establish the Connection between the System and the Database for the purpose of providing data that will be used for Filtering purposes in the *Get_Values()* function and the computation of ShadowID Rankings. It contains the Connection Variables needed to get data from the database.

c) Get_Values() Function

- This Function is supposed to acquire all the records that meet the filtering Criteria that have been selected by the user. In the Entity Based Association Rankings computation the Criterion of interest will include the time of proximity toward the POI, The Location in which both the POI and the ShadowID appeared together and the Number of times the POI and the ShadowID have been in contact with each other irrespective of the direction of communication (Who called who).
- The Event Based Rankings is generally an Entity based Ranking that doesn't necessarily have a POI but can must have a Location and the Time of observation, this means that the computed Ranking of the ShadowIDs will be based on the Likelihood of involvement to a particular event of interest.

d) Compute_Rankings() Function

- This is the main Computing function that will get its values from the *Establish_Connection()* function and the *Get_Values()* function, this function will compute the Ranking Values-per-Level using a looping approach as it traverses through both the set list of condition Variables and the Database records crosschecking for matches and computing the new Rank and Level values of the Existing Shadow ID and Adding new ShadowIDs that meet the specifications but are not within the list of existing Select_ShadowIDs.
- As opposed to all other Functions this function is a recursive one due to the Depth Level in which the user might require the system to compute Associations, and as described in the Methodology earlier the Ranking levels will be represented as letter *n* in which the final outcome will be:

$$R\_SID_i = \sum_{i=0}^{\infty}\left(\frac{1}{2^n}\right)$$

Equation 16: General Ranking Model

Where:

- ***R_SID*** is the final Ranking Value of the ShadowID,
- ***I*** is the ShadowID number in the list of shadowIDs and as denoted, the list of shadowIDs can Range from 0 to ∞,

- n is the Level of association where in this case n is the Recursion level. This means the more the number of Recursions the Higher value of n and subsequently the lower the Ranking Value at that point.
- The reason for using the $\left(\frac{1}{2^n}\right)$ model is because it splits down the levels of association in a equally divisible sequence which can be easy to associate and substitute between the final result and the initial Run Level.

The above are just a brief description of the functions of the Framework model used.

5. 6 Conditions for Computations

Apart from the Framework Functions there are the Conditions to which each function operates, the conditions in this case include:

1. Time Range
2. Date Range
3. Location of Observation
4. Person of Interest

5.6.1 Descriptions of Conditions for Computation

5.6.1.1 Time Range

- In the time Range Criteria the time is represented by two Values, the **StartTime** and the **EndTime,** They both are represented as of the type *Date*, they will be used to compute the duration of the observation that the user may wish to focus on so as to compute the Entity and Event based associations Rankings.

5.6.1.2 Time Range

- In the Date Range Criteria the **StartDate** and **EndDate** values of type *Date* will be used to filter the level of associations between the ShadowIDs and either an Event or the POI. They will define the limits to which the rankings of the ShadowID will be selected during Computation.

5.6.1.3 Location

- **Locations:** Since an event to occur there must be a location or list of locations, the **Location** Variable *array* of type *String[]* will be used in the filtering of the list of possible ShadowIDs to either an Entity or Event based on the Time, Selected_POI or even the Locations of interest if not all.

5.6.1.4 Person of Interest

- Finally the Person of Interest **POI**, this is the main Entity of focus in relation to computing the associations between either other Entities or Both other Entities and Events of interest. The Representation of this **POI** will be by the use of an array Data structure of type *String []*.

5.7 Tools for Prototype Development

The above functions will use the immediate data stated to compute the various Rankings as stated earlier as it does reflect the framework model designed and developed in this study.

The Main reason for developing a prototype of this Framework is to demonstrate the Automation possibility, applicability and accuracy of this model in developing a Reproducible and Reliable Entity-to-Entity, Entity-to-Event, based possible associations Rankings. In other words the developed prototype is just a proof of concept for the Framework model to be developed and used in improving the Surveillance strategies of government Security organizations using Currently Available Technology and readily available data for better management of its populations and improvement of National and International Security.

The Prototype was developed using the Java JDK 1.6 OOP-Language using the Netbeans 7.1 Development Environment.

The data management was done using the MySql technology which is part of the Lampp web Service system Framework.

The structure of the data in the database is divided into two Primary tables

1. The **CallLogs** table: This table contains all the call logs made from a specific time which is not limited in this case 1 Month, the table structure for the call logs includes the field which are required by the system to compute the Rankings mainly Entity Based Rankings. The Fields of interest to this framework model include and are limited to:
    a. **POI :** This is the person of interest's number and Normally the Call direction can shift to and fro the POI to the contact.
    b. **Location :** this is the Current location of the POI, although in this case for the sake of the Entity based Rankings it isn't Relevant.
    c. **Contact:** This is the other party to which the call is being made to or from by the POI, normally the roles will switch between the two based on the type of Ranking computations being performed.
    d. **Time_of_Contact:** This is the Particular time and date to which the communication between the **POI** and **Contact** was being done. It is required for both the Entity Based Rankings and Event Based Rankings.
    e. **Duration:** Although initially required during the initial Research stages, for simplification and minimization of computing requirements, it was discarded.

2. The **Proximity Table:** This table contains all the tracking information needed for the computation of the data to establish the Likelihood Associations between Entities and Events of interest. The data contained in this table includes and is limited to:
   a. **POI:** This is the person of interest to which the focus of observation has been conducted, but as the research Progressed this field became redundant and is no longer the field of focus.
   b. **ShadowID:** This is the collected active cell phone number that was detected at a particular **Time**, **Date** and **Location** of interest during the observation. This Field replaced the **POI** since both the **POI** was also represented in this field. This was arrived at during the research as it was found to give the same effect when computed but was much more efficient than observing from POI based computation.
   c. **Location:** This is the particular place that the observed surveillance data was collected. This is one of the MAIN variables which was and is used to filter the search for the purpose of both the Entity and Event based Rankings. Its weight is due to the fact that every event has to occur at a particular place in time and to associate people to each other who might know each other they have to be at the same place multiple times over a period of time.
   d. **Time_of_Observation:** This is the particular time to which the parties observed to be active were detected. This field is inform of the 24hr Time format.
   e. **Date:** This is the particular date to which the detected Active Cell phone numbers were observed. It is also a requirement for the filtration purpose for the Entity based and event based rankings of the collected **ShadowIDs**.
   f. **IMEI:** Though initially was considered to be a requirement to the computations of the **POI – ShadowID** rankings, It was deemed to be redundant as the **ShadowID** has proved to be sufficient enough to compute the same Rankings. As the factoring of the **IMEI** s effect on the Final Rankings computation Value is Negligible at best.

The main focus of this framework is to be able to compute the Likelihoods of associations between the entities and events, and due to the fact that the following Assumptions guide as rules for identifying events or associations, some basic rules stand and have acted as a guide to throughout the research life cycle,

5.8 Testing

To test the model against data and verify its validity of providing the solutions, two main tests were performed on the model prototype which involved the generation of Scenarios and list of associated in descending order of likelihood of association ShadowIDs. In every Senario there were up to 10 ShadowIDs listed and placed separately.

5.8.1 Testing Event Scenarios

For the model testing purpose three scenarios were generated which had different results that were supposed to be computed by the system. The three include the following.

Event1:

ShadowID's association is determined by the number of times the ShadowID appears within the same location as ShadowID had appeared irrespective of the time difference between the two.

- Such scenarios can be used to describe situations like Deaddrop Events where thePOI instead of communicating directly to the ShadowID, they leave each other messages or items at specific drop points known as Deaddropes.

  Condition for Event1:

  Same Locations at different times.

Event2

ShadowIDs association to the POI is determined by the number of times both the POI and the ShadowID appear at the same Locations at the same time.

- Such scenarios show possible and direct interactions between the POI and the ShadowID and are exhibited by groups conducting the same or similar activities within different locations.

  Condition for Event2:

  Same Locations at the same Time.

Event3

ShadowIDs association to the POI is determined by the number of times both the POI and the ShadowID appear at the same Location at the same time in multiple instances.

- Such scenarios show possible and direct interactions between the POI and the ShadowID and are exhibited by groups conducting the same or similar activities within different locations also show close social associations such as family members at home where they meet every day and classmates in school where they meet at a particular time over a duration days.

  Condition for Event3:

  Same single Location at the same multiple time occurrences.

5.8.2 Accuracy Testing Results Based on display of all the actual ShadowIDs in the Listing by the system

For the purpose of this research three events were generated to simulate and validate the proposed model's accuracy.

The events generated were based on actual events but due to the sensitive nature of some of the events, the locations, parties involved (ShadowIDs) were altered.

The table below describes the three separate events of interest and their unique quantifiable properties.

From this criterion divided it into two tables, the Proximity logs where we used the POI or ShadowID, the Location to which the particular ShadowID appeared, the Time and Date of appearance as the key features of interest. And the Call logs where the POI or ShadowID, Contact, Date and Time of contact, Duration Were the key features of interest. They were computed independently based on the ranking formula.

The generated event properties table and their unique conditions for the three separate events were as follows:

Event	Condition	Total Number of ShadowIDs	Number of intercommunicating ShadowIDs
Event_1	**All ShadowIDs appear within the same Locations at the Same Time.**	**10**	**3**
Event_2	**All ShadowIDs appear within the same single Location multiple times at the Same Time.**	**10**	**4**
Event_3 Deaddrop Event	**All ShadowIDs appear within the same Locations at Different Times.**	**10**	**1**

Table 1: Event Criteria and their unique Characteristics

The conditions set in the model test and evaluation was based on increasing data size. In this case we set all three events to contain 10 individuals with varying degrees of associations to each other. Each generated event contained multiple locations and time of appearance within those locations in which some overlapped with other events.

5.8.3 Performance Results

Figure 5: Event 1 Framework Performance Results

As Shown in the table *Deaddrop:* above the general listing accuracy remained optimal as all the ShadowIDs were listed during the Event Simulation but Ordered listing Accuracy Decreased down to 80% of the possible listings. This affected its overall Model Accuracy to 90%. This means that only 90% of all the Associated ShadowIDs were to be listed in the first slots during the model simulation process. This was due to the presence of Noise in the data. The above event was generally best suited for Entity Based Rankings.

Figure 6: Event 2 Framework Performance Results

As Shown in the table *Figure 6:* above the general listing accuracy remained optimal as all the ShadowIDs were listed during the Event Simulation but Ordered listing Accuracy Decreased down to 50% of the possible listings. This affected its overall Model Accuracy to 75%. This means that only 75% of all the Associated ShadowIDs were to be listed in the first slots during the model simulation process. This was due to the presence of Noise in the data. The above event was generally best suited for Single Location Based Events Rankings like Daily or periodic Meetings.

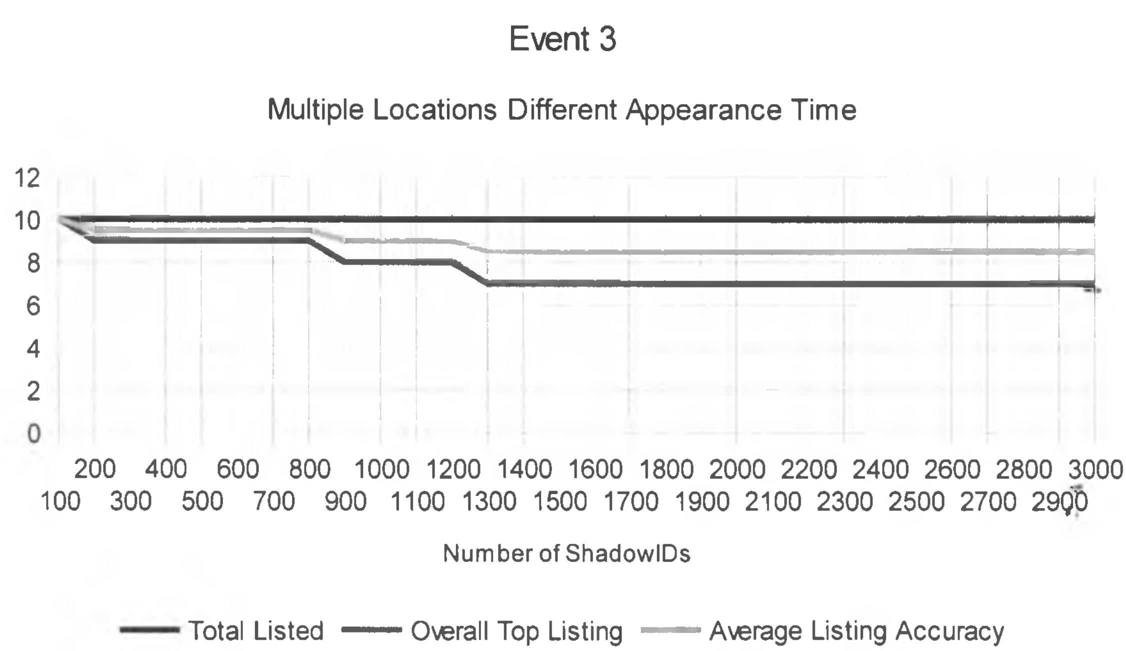


Figure 7: Event 3 Framework Performance Results

As Shown in the table *Figure 7* above the general listing accuracy remained optimal as all the ShadowIDs were listed during the Event Simulation but Ordered listing Accuracy Decreased down to 70% of the possible listings. This affected its overall Model Accuracy to 85%. This means that only 85% of all the Associated ShadowIDs were to be listed in the first slots during the model simulation process. This was due to the presence of Noise in the data. The above event was generally best suited for Single Location Based Events Rankings like the DeadDrop Event where one person would place an Item in a hidden location for the other party to come and collect later on several occasions.

From the above generated events we injected each individual event into a separate database table of 100 records, we tested the general listing accuracy, and ordered listing accuracy of the model by increasing the record counts both the call logs and the proximity logs by the order of 100 from the first 100 records (including the event specific records) to the final 3000 records.

The Event 2 showed a lesser Ordered listing accuracy due to the observed fact that there are people who are usually within the same location as a result of either residence or location of occupation. The above average entity listing order accuracy faults were however overcame to a great extent by the included computation association depth levels which is the computing of association levels based on both the ShadowID and all the listed associations in a tree format where the parent was the POI and the child was the ShadowID. This was done by giving the ShadowID unique fixed values-per-level. This resulted into higher rankings for highly associated ShadowIDs and the coincidental ShadowIDs that are not really associated with the event remained with their ShadowID values reducing their overall rankings.

CHAPTER 6: DISCUSSIONS

From the above illustrations and results representations it shows that the General order less listing of possible ShadowIDs is highly accurate since the system Framework can represent with a high level of accuracy the applicability of the Framework with Justifiable results, But in the Ordered Listing of the Results it becomes a problem since it does ranks others with a higher or the same value as a more likely candidate thus reducing the overall system accuracy.

6.1Key Findings

Some key issues have become apparent during the research based on surveillance and below are the noting.

1. Individuals who are socially related (like friends, family business colleagues) always do exhibit patterns in which involves appearing at the same location at the same time.
2. Events are as a result of active human activity over a period of time.
3. It is easier to identify events when there are multiple parties involved than when created by a single individual. The more the parties involved the easier it is to identify the event.

6.2Key Achievements

The research has resulted to some noted achievements which include:

1. The development of a quantifiable mathematical Model that can be used with desirable confidence to compute the possible social hierarchy of relationships between individuals from observing single Persons movement activities only.
2. An Automatable framework has been developed that can externalize the human computation process in using the mathematical Model to compute the rankings.
3. Events based on human activities have now been defined based on symbolic relations between factors of interest.

6.3Key Challenges

This research like any other has not been without its own challenges, both technical and resource wise as they include.

1. Unavailability of actual real world data to compute real world events thus limiting the models accuracy.
2. Limited Research time as this research has proved to be wide and has a lot of factors that still need both interpretation and analysis.

6.4Assumptions and Limitation of scope

The completion of this research and development of this framework and its prototype has come with its own challenges that have been overcome and some ignored due to its near impossibility to tackle natures which are described below.

The unavailability of the actual mobile subscriber data from the Telecommunication networks due to subscriber privacy and Business ethics means that the data generated is not be the actual data from the mobile network providers, not actual people but fictional characters simulating real events and associations. The time of observation and the number for records generated is minimal just to reflect at least two scenarios where the system framework will work.

Also it is assumed that the data collected has been generated from the previous collection of records from the mobile BTS and filtered based on a GSM tracking system that is installed on the Telecommunication networks.

The above provide limitations to the accuracy of the framework model suggested results as not the actual data has been used but a generated and less accurate data have been used.

The nature of this research is not without its controversial aspect of invasion of privacy to which it is illegal by some government laws. Thus the adoption of this framework means policies within the law will have to be altered to allow the successful implementation of this system.

Finally this research computes the depth of recursion and Association computation up to 1 Level from the initial POI in the span of up to one week time span. This means that the associations of likelihood of association will focus on the network hierarchy depth of only one level this is due to the immense computing power required described below.

6.5Conclusion and Recommendations

With the development of this framework it has enabled the automated computation framework strategy development that if used will result in the better identification of possible associations between individuals and Events. As it uses the factors that are of interest to events and Individuals. If implemented it would considerably increase the security of any physical domain or country as it would be easier and much cheaper to identify criminals or Persons of Interest accomplices without even having the need to send Agents to the ground and/or interrogation of the Person of Interest in which they are both quite Risky and expensive measures that might expose the Security agencies to the Person of Interest's detection thus hiding or evading capture which might result in the failure of securing the public against Dangerous elements within society.

Due to limited resources and data for this research it does not go without saying that there has been some challenges and limitations in this research both during and as a result of this research. The observed and experienced Limitations include but are not restricted to the following.

- Lack of real world data from the Countries telecommunication companies. (in other words they were unwilling to provide the data due to their business policies that do not allow the release of such data).
- Lack of adequate Computing power thus limiting the depth of computation and design of the system. Since the framework prototype was developed using a laptop that has a computing CPU speed of 2.2GHZ and a Physical Memory of 2GB which is shared between other processes including the Operating System, the use of Recursion has been purely restricted to 2 levels as the higher the levels of recursion the higher the minimum computing requirements since this System prototype is Currently of type $\Theta(n) = \mathbf{n^3}$ which the first recursion is due to the Levels computation, the second Filtering Requirements and the third for the computation of new and existing ShadowID Ranking Values
- Limited Time for the research due to the complexities involved in events identification and computation.
- Legal Privacy implications that will be breached due to the implementation of such frameworks.

Accuracy Based on general listing of ShadowID Rankings

Based on the results shown from the computation process of this framework it is easy to present the list of possible accomplices and associations to an accuracy of .99 percent if they have been identified, have been in the same location with the POI and have been in contact with the POI. Factor that affects its accuracy is the appearance of other entities which might be within the same location as the POI maybe due to the possibility that that might be the area of the uninvolved ShadowIDs place of residence of occupation as this will also affect the particular ShadowIDs Ranking Value.

Accuracy Based on general ordered listing of ShadowID Rankings

But in the order of likelihood associations it's accuracy is reduced due to the fact that Several other uninvolved ShadowIDs can be identified several times within a specific location due to either them being there for other unspecified reasons such as Their residence that's (*where they live)* or their place of occupation, (*that's where they work).* The accuracy in this case is not limited to a specific value but is greatly affected by the Size of the population observed per location, number of locations observed, Time Range. Thus it wouldn't be appropriate to state with much justification the accuracy order in this stage.

6.5.1 Recommendations

- Further Research on the possible use of direction based Contact computations should be considered as it will provide better and more accurate results of the rankings.
- Actual integration methods that use this framework to various other forms of surveillance such as CCTV and online activity patterns should be considered.
- The use of IMEI should be factored in during the computing process due to the fact that some individuals might use several Different sim cards and switch between handsets, as it is a commonly observed trend.

References

- Tom M. Mitchell (March 1, 1997). *Machine Learning.* Canada: McGraw-Hill. 32,38, 81, 166-169,230-240.
- Paul Deitel, Harvey Deitel (2009). JAVA^{TM}- How to program 8th ed. 156-192,289-291,811-836,1184-1238.
- Dragos Margineantu (Jan 1, 2010). *Machine learning algorithms for event detection*: Springer. 257,258,259.
- Anders Jonsson, et al. (Sept 4, 2010). ICDSC 2010 August 31. *A Learning Approach to Interactive Browsing of Surveillance Content.* 2,3,4.
- Dimitrios Lymberopoulos, et al. (Sept 4, 2010). PETRA 2008 July15-19, 2008,. *Extracting Spatiotemporal Human Activity Patterns in Assisted Living using a Home Sensor Network.* 1,2,3.
- A. Prem Kumar, et al. (2009). *Object tracking using Radial basis function networks.* 3,4,5,6.
- Qi Zang and Reinhard Klette. (2003*). Object Classification and Tracking in Video Surveillance.* 3,4,5,6,7.
- Michael Luke Bullock. (March 2009). *The evolution of surveillance technology beyond the panopticon.* 3-9
- Kalpesh A. Popat. (December 2011). • Location Update Strategies in Mobile Computing. *International Journal of Computer Science and Communication.* 2 (2), 305-310 .

APENDIX:

System Structure and Relevant code Snipets

From the Top:

The system Prototype defined from the above general Functions consists of nine Classes which are:

- *RankGUI.java*

This class is the main user interaction interface class which was developed using NetBeans IDE 7.1.1 The main function of this class is to provide an interface between the user and the system, it contains the Main GUI in which the user will issue instructions and compute the rankings based on the selection Criterion and Rankings Basis (*Weather it is Either Entity or Event Based Rankings*).from the time it loads It executes the Function calls from other classes Mainly the *Proximity.java* and *Calllog.java* classes. Due to the GUI having two tables showing all the data contained in both the *Calllogs* and *Proximity* table, it calls the above two classes and loads them immediately so that the two classes can establish connection to the database and pass the data from the database to the two table for presentation to the system user. It also contains the Function Call for the *RankShadowID.java* class which contains the query and Ranking computation Classes. Apart from that the *RankGUI* contains its own MySql connection and data acquisition functions which Filters the Database Records to provide Unique data and eliminate repetition of the data on the combo Boxes which are used to enable the Selection of specific values of interest that will serve as a conditional filtering variable basis. Once the *cmdComputeEvent* button is clicked this gets the *Criteria, POISelect, Location,* values which are represented as Strings, It then defines a new function object of class RankShadowID and names it *Compute* then finally invokes the Computes' object *Compute.GetValues(POISelect,Location,0)* method, passes the value to the Method and executes the Rankings based on the settings defined. This class also contains the Reset Command which calls the *Compute.Reset()* method which disposes the array of computed ShadowIDs and their consequent values ready for the next computation which the user might reissue to the system. It also contains the *main()* function which is the point of execution from which the package is loaded from. The *main()* function only contains the *run()* function which loads the GUI and the Look and feel settings for the systems GUI.

Proximity.java

This class is the main data acquisition class that contains functions that Data from the proximity table to the *RankGUI.java* proximity table displayed at the bottom right of the GUI. This class is usually automatically generated by the netbeans IDE when you use the configuration wizard set the data source and data elements that are to be displayed on the table.

The main function of his class is to:

1. Query the Database

2. Get all the Values from the Database table Proximity
3. Pass the Values from the table to the RankGUI.java Class

Calllog.java

This class is the main data acquisition class that contains functions that Data from the calllog table to the *RankGUI.java* calllog table displayed at the bottom left of the GUI. This class is usually automatically generated by the netbeans IDE when you use the configuration wizard set the data source and data elements that are to be displayed on the table.

The main function of his class is to:

1. Query the Database
2. Get all the Values from the Database table calllog
3. Pass the Values from the table to the RankGUI.java Class

LoadCboItems.java

This class is used to pass the various database record variables from the Database tables to the Comboboxes that are used for event based and entity based filtering purposes.

The functions included within this table include the *CallValues()* and the *AddCboValue()* functions

Basically the *CallValues()* function is used to connect, query and get the data(Should be UNIQUE or DISTINCT) from the database and pass them to the *AddCboValue()* function.

The code snippet is displayed below:

```
public void CallValues( ) throws SQLException
 {
        Connection connection = null; // manages connection
        Statement statement = null; // query statement
        ResultSet resultSet = null; // manages results
        String[] Collums = new String[8];
        Collums[0]="ShadowID";
        Collums[1]="Location";
        Collums[2]="Time_of_Observation";
        Collums[3]="Time_of_Observation";
        Collums[4]="Date";
        Collums[5]="Date";
        Collums[6]="Contact";
        Collums[7]="Duration";
```

```
// connect to database rankpoi and query database
   // establish connection to database
            connection = DriverManager.getConnection(DATABASE_URL, "root", "" );
// create Statement for querying database
            statement = connection.createStatement();

// query database

resultSet = statement.executeQuery("SELECT DISTINCT ShadowID FROM proximity" );


  AddCboValue(RankGUI.cboPOI, statement.executeQuery("SELECT DISTINCT ShadowID FROM proximity"
),Collums[0]);
  AddCboValue(RankGUI.cboLocation, statement.executeQuery("SELECT DISTINCT Location FROM
proximity" ),Collums[1]);
  AddCboValue(RankGUI.cboStartTime, statement.executeQuery("SELECT DISTINCT Time_of_Observation
FROM proximity" ),Collums[2]);
  AddCboValue(RankGUI.cboEndTime, statement.executeQuery("SELECT DISTINCT Time_of_Observation
FROM proximity" ),Collums[3]);
  AddCboValue(RankGUI.cboStartDate, statement.executeQuery("SELECT DISTINCT Date FROM proximity"
),Collums[4]);
  AddCboValue(RankGUI.cboEndDate, statement.executeQuery("SELECT DISTINCT Date FROM proximity"
),Collums[5]);
  AddCboValue(RankGUI.cboDuration, statement.executeQuery("SELECT DISTINCT Duration FROM calllog"
),Collums[7]);

  RankGUI.txtOutputCalls.setVisible(false);
  RankGUI.txtOutputProximity.setVisible(false);
  RankGUI.cboDuration.setEnabled(false);
  RankGUI.cboEndDate.setEnabled(false);
  RankGUI.cboEndTime.setEnabled(false);
  RankGUI.cboLocation.setEnabled(false);
  RankGUI.cboStartDate.setEnabled(false);
  RankGUI.cboStartTime.setEnabled(false);
  RankGUI.cboCriteria.setEnabled(false);
  RankGUI.cboTimeOptions.setEnabled(false);
```

```
RankGUI.cboPOI.setEnabled(false);

try
{
resultSet.close();
statement.close();
connection.close();
} // end try
catch ( Exception exception )
{
exception.printStackTrace();
} // end catch
} // end finally
```

The code for the AddCboValue() is shown Below:

```
void AddCboValue(JComboBox CboItem, ResultSet FieldQuery, String CollumName) throws SQLException
{
  try
    {
  while(FieldQuery.next() )
  {
   CboItem.addItem(FieldQuery.getString(CollumName));
   //ComboBoxes[0].addItem(resultSet.getString(Collums[0]));
   //AddCboValue(RankGUI.cboLocation, resultSet[i],Collums[i]);
  }
  } // end try
 catch(SQLException sqlException )
 {
 sqlException.printStackTrace();
 } // end catch
 finally // ensure resultSet, statement and connection are closed
 {
 }
 }
```

These two main functions are what is used to provide the Comboboxes in the RankGUI.java with their Values.

ShadowID.java

This is the main data Structure representation entity. It is the main data that is to be created, field values computed and finally presented to the user of the system based on the particular search criteria and Selected POI. It contains all the relevant fields that are needed during the computation and ranking process of the ShadowID as its main code is as presented below.

```
public class ShadowID
{
    //Publicly Accessible Varriables
    String ShadowID[]; //Shadow ID
    public double RankValue[];
    public int Rank_Level[];
    public int totalRankings[];
    public int Total_Rankings[];
    public double Call_Rankings[];
    public double Proximity_Rankings[];
    public double ProximityRankings[];
    public double CallRankings[];
    //public String Process;
    String[] Shadow_ID;
    double[] Rank_Value;
    int[] Level;

    //Create Construdtor
    ShadowID(int size)
    {
        //varriables with an Underscore Between the Names will be the publicly assigned or referenced Varriables
        Rank_Value = new double[size];
        Shadow_ID = new String[size];
        Rank_Level= new int[size];
        Total_Rankings=new int[size];
        Call_Rankings=new double[size];
        Proximity_Rankings = new double[size];
        ShadowID=Shadow_ID;
```

```
    RankValue=Rank_Value;
    Level = Rank_Level;
    CallRankings = Call_Rankings;
    ProximityRankings = Proximity_Rankings;
    totalRankings=Total_Rankings;
  }
}
```

RankShadowID.java

This is the main computation Algorithm class that contains all the various functions that are required to compute the different types of events. This class instance is created from the RankGUI.java class, the relevant function called and assigned the relevant Values needed for the computation process.

The functions contained in this class include:

EstablishCalls_Connection( ) and *Establish_Connection( ):* this are the two main functions that Establish connection between the Database and the Functions contained in the RankShadowID.java class. When the required function is called or invoked from the RankGUI.java class and values passed to it the Functions initially call either of these functions and and uses the results from either of these function depending on the Ranking Criterion to compute the final result.

The code Snippet for the above function is shown below:

```
public void Establish_Connection( ) throws SQLException
  {
      connection=null;
      statement=null;
      resultSet = null;
      // establish connection to database
     Connection = DriverManager.getConnection(DATABASE_URL, "root","");
      // create Statement for querying database
      statement = connection.createStatement();
      // query database

      resultSet = statement.executeQuery("SELECT * FROM proximity");

      // process query results
      ResultSetMetaData metaData = resultSet.getMetaData();
```

```
    String[] args = null;

}
```

Due to major similarities within the Ranking Algorithms, one Algorithm will be described in detail from a pseudo code and Source code snippet point of view. The rest will only highlight the important deviations between them and the main Computing algorithm.

The Ranking Algorithms used in this System prototype include the following functions:

- *ComputeEntityRankings()*: This is the main Entity based Relation Likelyhood Algorithm code, it generally performs the POI to ShadowID Rankings computation. Its input Varriable include The POI_Select (POI) AND Run_Level (which is the Depth of the ranking associations).

The pseudo procedure of this algorithm is as described below:

```
ComputeEntityRankings(POI_Select, Run_Level)

{

N= Number of records;

X= Number of locations;

y=Number of ShadowIDs;

ShadowID[y] =the different ShadowIDs Indentified;

Location[x] =the different Locations POI was Identified at;

For i=0 to n;

Get_All_Different_POI.locations and add to Location[x]

If(ShadowID.CurrentLocation=Location[x])

    {

    For j=0 to Y

    {

    If(ShadowID[j].Exist in Generated List)
```

```
{
Compute.NewShadowID[j].RankValue =PreviousShadowID[j].RankValue+(1/(2^Run_Level));
Compute.NewShadowID[j].TotalRankings =Previous ShadowID[j].TotalRankings+1;
}
Else
{
AddNew ShadowID[j+1];
AssignNew ShadowID[j+1].RankValue=1/(2^Run_Level));
AssingNewShadowID[j+1].RankLevel=RunLevel;
AssignNewShadowID[j+1].TotalRankings=1;
}
Next j;
}
}
}End Computation
```

The above pseudocode shows that the Computation requirements or values of interest is primarily the Location which the POI appeared only. And is described by the Code snippet below.

```
int ComputeEntityRankings(String POI_Select, int Run_Level) throws SQLException
  {
    Shadow_ID=POI_Select;
    Location_Exist=false;
    Locations = new String[10];
    SelectedID.Shadow_ID[0]=Shadow_ID;
    SelectedID.Rank_Level[0]=0;
    SelectedID.Proximity_Rankings[0]=0;
    SelectedID.Total_Rankings[0]=1;
```

```
//RankGUI.txtOutputCalls.append("The ShadowID "+SelectedID.Shadow_ID[0]+"\n");
if(Run_Level<=3)
{
  // <editor-fold defaultstate="collapsed" desc="Getting Conditions Varriables">
Establish_Connection();
 try
 {

    for(int i=0;i<20;i++)
    {
    if(SelectedID.Shadow_ID[i]==null)
      {
        //RankGUI.txtOutput.append(i+" It is empty\n");
      }
      else
      {
        while(resultSet.next())
      {
        Current_Location=(String) resultSet.getObject("Location");
        Shadow_ID =(String) resultSet.getObject("ShadowID");
        //RankGUI.txtOutputCalls.append(Current_Location+" Comparing Shadow ids...\n");
        if(Shadow_ID.compareToIgnoreCase(SelectedID.Shadow_ID[i])==0)
        {
        for(int L_Num=0;L_Num<Locations.length;L_Num++)
        {
          if(Locations[L_Num]!=null)
          {

            if(Locations[L_Num].compareToIgnoreCase(Current_Location)==0)
            {
              Location_Exist=true;
            }
            else{}
          }
          else{}
        }
        if(Location_Exist==false)
```

```
            {

               Locations[l]=(String) resultSet.getObject("Location");
               RankGUI.txtOutputCalls.append(Locations[l]+"\n");
               l++;
               Location_Exist=false;
            }
            else
            {}
      }//end while loop
         Location_Exist=false;

    }
      }//resultSet.first();

    }//end For loop
}//End Try
catch(SQLException sqlException )
{
   sqlException.printStackTrace();
} // end catch
finally // ensure resultSet, statement and connection are closed
{
   try
   {

   } // end try
   catch ( Exception exception )
   {
      exception.printStackTrace();
   } // end catch
} // end finally
//</editor-fold>

// <editor-fold defaultstate="collapsed" desc="Actual Entity Rankings Code">
Establish_Connection();
   RankGUI.ProgressBarMain.setValue(0);
```

```
if(resultSet.isLast()==true)
{
   //RankGUI.txtOutputProximity.append("Last record Reached...\n");
   resultSet.first();
}
else
{
   //RankGUI.txtOutputProximity.append("Last record not yet Reached... "+resultSet.getRow()+"\n");
}
int loc_num=0;
String Current_ShadowID=null;
boolean SID_Exists;
boolean LocationMatch;
int Limit=0;
Run_Level=1;
SelectedID.Shadow_ID[0]=POI_Select;
while(resultSet.next())
      {
         int p=0;
         int Lmmt=0;
         SID_Exists=false;
         LocationMatch=false;
         Current_ShadowID=resultSet.getString("ShadowID");
         RankGUI.ProgressBarMain.setValue(resultSet.getRow());
         while(Locations[p]!=null)
          {
             if(Locations[p].compareToIgnoreCase(resultSet.getString("Location"))==0)
             {
                LocationMatch=true;
                int Lmt=0;
                while(SelectedID.Shadow_ID[Lmt]!=null)
                {
                      if(SelectedID.Shadow_ID[Lmt].compareToIgnoreCase(Current_ShadowID)==0)
                   {
                      SID_Exists=true;
```

```
SelectedID.Proximity_Rankings[Lmt]=SelectedID.Proximity_Rankings[Lmt]+(1/Math.pow(2, Run_Level));
                    SelectedID.Total_Rankings[Lmt]=SelectedID.Total_Rankings[Lmt]+1;

SelectedID.Rank_Value[Lmt]=SelectedID.Proximity_Rankings[Lmt]+SelectedID.Call_Rankings[Lmt];
                    //RankGUI.txtOutputProximity.append(SelectedID.Shadow_ID[Lmmt]+" Has been
Identifyed "
                    //+SelectedID.Total_Rankings[Lmmt]+" Times "+1/Math.pow(2, Run_Level)+"
"+SelectedID.Proximity_Rankings[Lmmt]+"\n");
                    // RankGUI.txtOutputProximity.append("ShadowID "+SelectedID.Shadow_ID[Lmmt]+"
Already Exists at index "+Lmmt+" For "+Locations[p]+" \n");
                    //Lmt++;
                  }
                    else
                    {}
                    Lmt++;
                }
                if(SID_Exists==false && LocationMatch==true)
              {
                SelectedID.Rank_Level[Lmt]=Run_Level;
                SelectedID.Proximity_Rankings[Lmt]=SelectedID.Proximity_Rankings[Lmt]+(1/Math.pow(2,
Run_Level));

SelectedID.Rank_Value[Lmt]=SelectedID.Proximity_Rankings[Lmt]+SelectedID.Call_Rankings[Lmt];
                RankGUI.txtOutputProximity.append(SelectedID.Proximity_Rankings[Lmt]+"\n");
                RankGUI.txtOutputProximity.append("SID "+Current_ShadowID+" Doesnt exists Adding at
index "+Lmt+" For "+Locations[p]+"\n");
                SelectedID.Shadow_ID[Lmt]=Current_ShadowID;
                SID_Exists=false;
                LocationMatch=false;
              }
              else{}
              SID_Exists=false;
              LocationMatch=false;
              }
              p++;
              }
```

```
            }
    }
    else
    {
      //ComputeEntityRankings(SelectedID.Shadow_ID[Run_Level+1], Run_Level+1);
    }
        //RankOutput.main(args);
    //RankOutput.main(args);
        ComputeCallLogsRankings(1);
       return 0;

}
```

The Event Based calculations are basically Entity based calculations with multiple Filtering procedures that narrows down the focus and list to the only ones that match the specifications.

The Event based Functions and their Brief description is detailed below.

- *GetValuesLocationBased():* This is the Ranking Algorithm that is used to Identify the ShadowIDs that appeared within the same location to the POI and it is exactly the same as the Entity based Ranking computations.
- *ComputeRankingSingleLocation():* This function is used to identify the list of ShadowIDs that were identified to be within the same single location as the POI at the same time during different occations which the ShadowID appeared.
- *GetValuesAll():*This is the Ranking Algorithm that is used to Identify the ShadowIDs that appeared within the same location to the POI at exactly the same time. The main difference is the time filtering procedure that has been added to narrow down further the list of possible associations.
- *Compute_TimeBasedRankings()*:The Ranking Algorithm that is used to Identify the ShadowIDs that appeared within the same location to the POI at exactly the same time. The main difference is the time filtering procedure that has been added to narrow down further the list of possible associations. This is based on a time range, thus the filtering process is made to narrow down the list that matches the time Range i.e from what time to what time and at which location.
- *Compute_DateBasedRankings().*The Ranking Algorithm that is used to Identify the ShadowIDs that appeared within the same location to the POI at exactly the same time. The main difference is the time filtering procedure that has been added to narrow down further the list of possible associations. This is based on a date range, thus the filtering process is made to narrow down the list that matches the date Range i.e from what date to what date and at which location.

- *ComputeCallLogsRankings():* This is the main callLogs association Computation algorithm that ranks the ShadowIDs likelihood relations to the POI based on the Communications between the two, Direction is not a factor of consideration.

  The pseudocode description is as shown below.

```
ComputeCallLogsRankings(Run_level)
{
n = ShadowID.length;
For i=1 to n
{
If(ShadowID==Contact || POI ==Contact ||ShadowID==Caller||POI==Caller)
{
ShadowID.Call_Rankings[i]= ShadowID.Call_Rankings[i]+((1(2,^Run_lvl))+Duration;
}
else{}
}
 } end
```

The SourceCode Snippet to the Above function is shown below.

```
int ComputeCallLogsRankings(int Run_lvl) throws SQLException
  {
    EstablishCalls_Connection();
        RankGUI.ProgressBarMain.setValue(0);


        if(resultSet.isLast()==true)
        {
          //RankGUI.txtOutputProximity.append("Last record Reached...\n");
          resultSet.first();
        }
        else
        {
          //RankGUI.txtOutputProximity.append("Last record not yet Reached... "+resultSet.getRow()+"\n");
        }

        //String Current_ShadowID=null;
```

```
            //int c=0;
          while(resultSet.next())
               {
                 int loc=0;
                 while(SelectedID.Shadow_ID[loc]!=null)
                 {

                   //for(int c=0;c<200;c++)
                   //{
                     RankGUI.txtOutputCalls.append(SelectedID.Shadow_ID[loc]+" "+resultSet.getString("POI")+
" "+ resultSet.getString("Contact")+"\n");
                     if(SelectedID.Shadow_ID[loc].compareToIgnoreCase(resultSet.getString("POI"))==0 ||
SelectedID.Shadow_ID[loc].compareToIgnoreCase(resultSet.getString("Contact"))==0 &&
SelectedID.Shadow_ID[loc].compareToIgnoreCase(resultSet.getString("Contact"))==0 ||
SelectedID.Shadow_ID[loc].compareToIgnoreCase(resultSet.getString("POI"))==0)
                     {
                       RankGUI.txtOutputCalls.append("Match Found \n");
                       SelectedID.Call_Rankings[loc]= SelectedID.Call_Rankings[loc]+((1/Math.pow(2,
Run_lvl))+(double) resultSet.getDouble("Duration"));

//SelectedID.Proximity_Rankings[loc]=SelectedID.Proximity_Rankings[loc]+((1/Math.pow(2,
Run_lvl))+(2*(double) resultSet.getInt("Duration")));

SelectedID.Rank_Value[loc]=SelectedID.Proximity_Rankings[loc]+SelectedID.Call_Rankings[loc];
                     }
                     else
                     {}
                   loc++;
                 }
               }
          RankOutput.main(args);

       return 0;
   }
```

RankOutput.java

This is the main result presentation class that shows the final results in an easy to understand format. The main Class contains the Table parameter declarations, the table field type declarations and the progressbar renderer that is used to generate various progressbar values based on the value passed to it from the data collection looping statements.

The main output looping code snippet used to pass the values to the Results table is as below.

```
for (int i = 1; i < ProximityRank.length; i++)
  {
    ProximityRank [i] = (int) RankShadowID.SelectedID.Proximity_Rankings[i];
    CallRankings [i] = (int) (RankShadowID.SelectedID.Call_Rankings[i]);
    OverallRank [i] = (int)
((RankShadowID.SelectedID.Call_Rankings[i]+RankShadowID.SelectedID.Proximity_Rankings[i])/2);
  }
```

The main system Structure of its computation process from both the users and system Structures point of view is detailed in the diagram below.

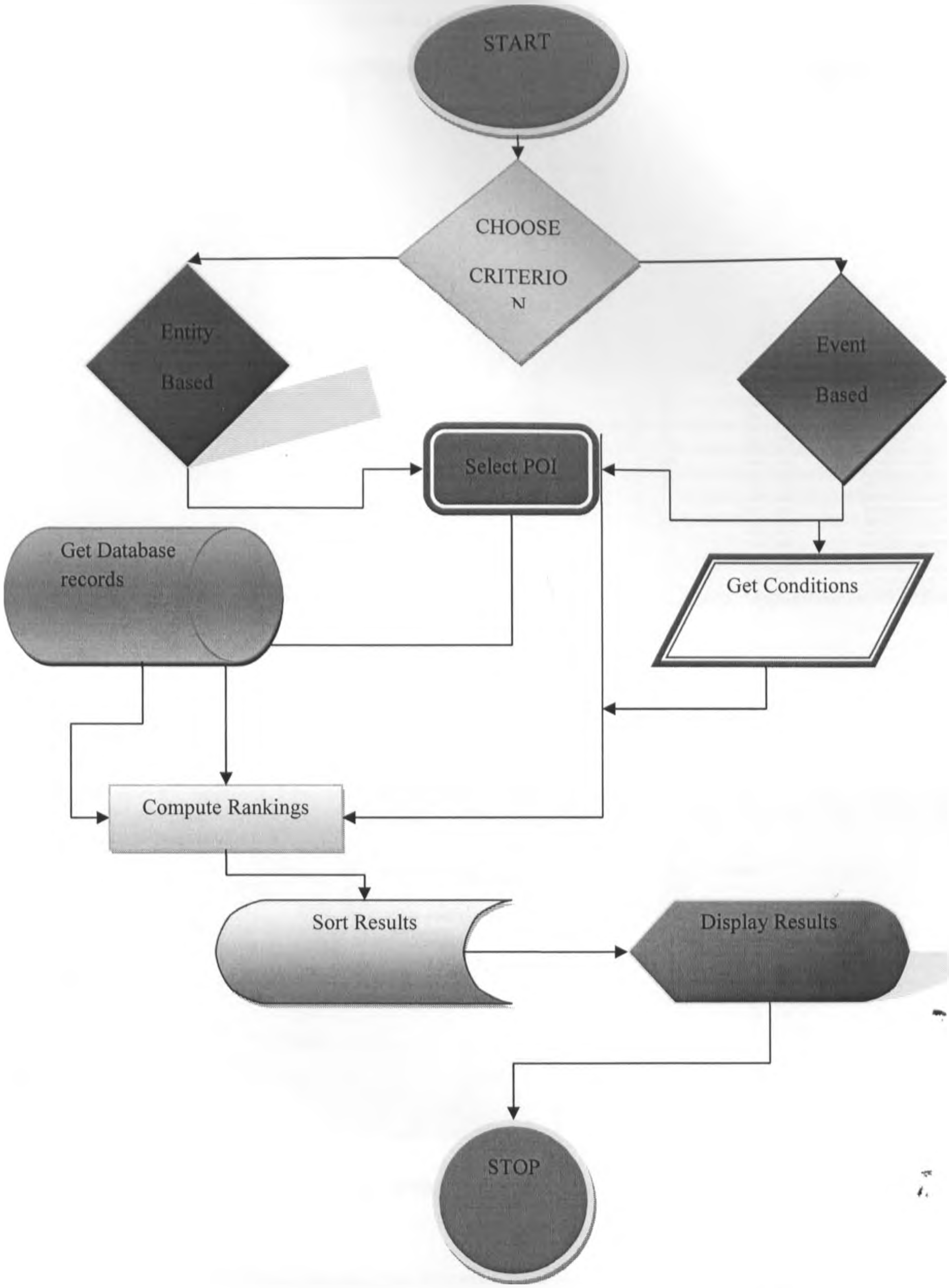


Figure 8 System Overview Flowchart

System Prototype User Manual

Step1: Start system. The system GUI should look as presented below.

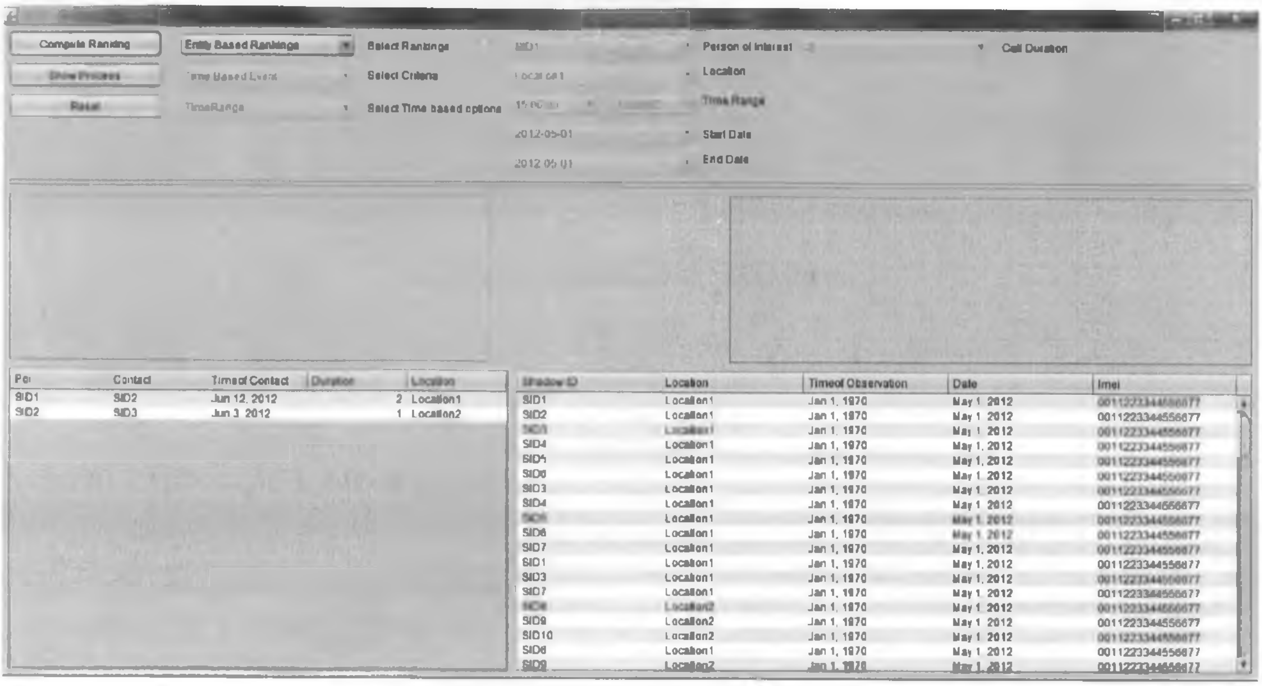


Figure 9 Main Prototype GUI

Step2: Select the Ranking Criterion:

Figure 10 Ranking Criterion Selection

Step3: If it's Entity Based Rankings just select the POI the List of shadowIDs shown in the POI combobox as below.

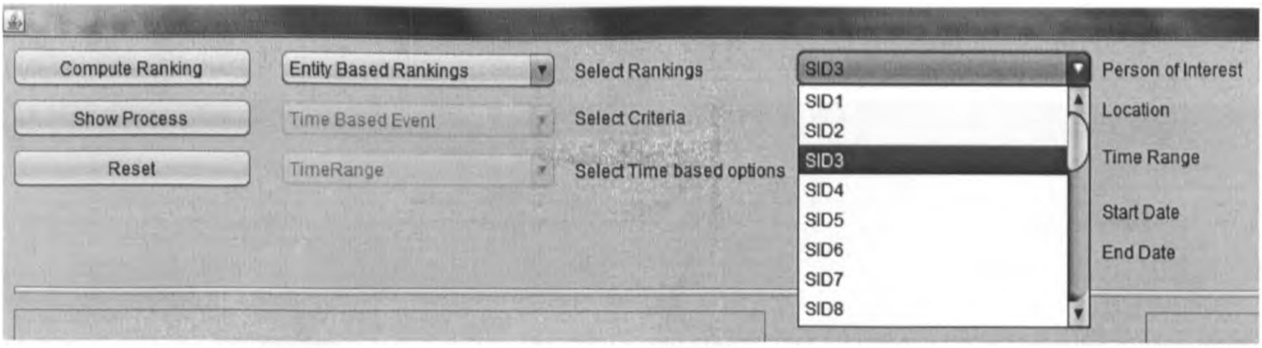
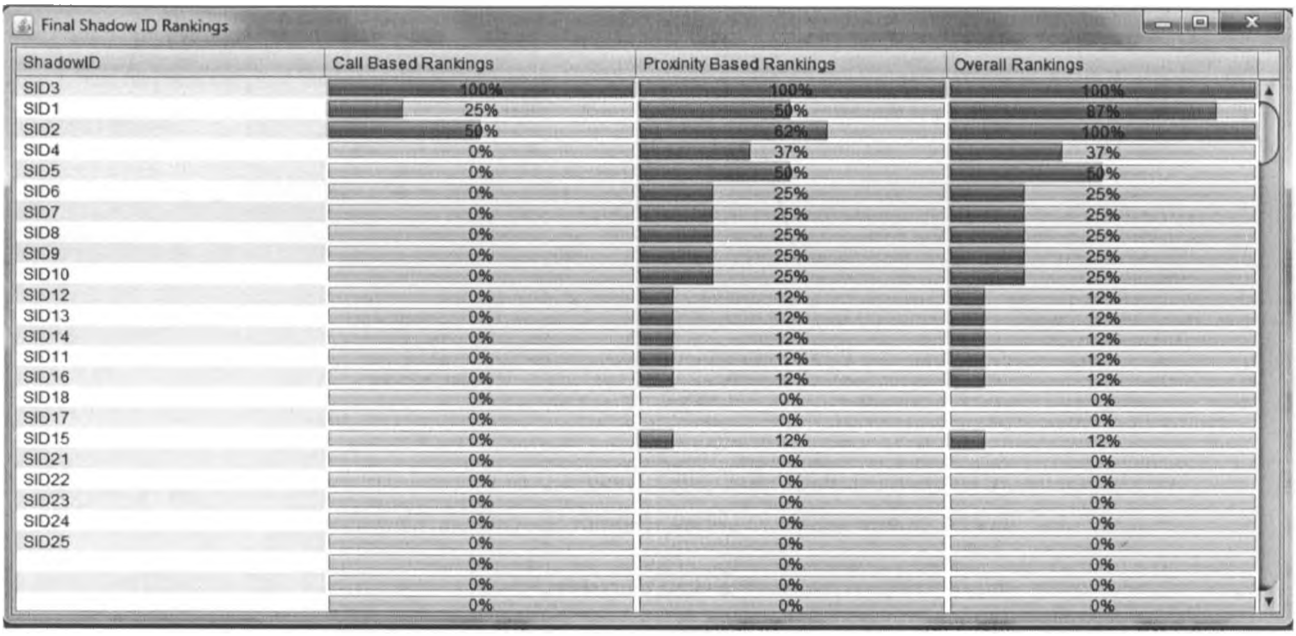


Figure 11 POI selection from ShadowID list

Step4: After finally selecting the POI Click on the Compute Ranking button to perform the Computation process and View the End result as shown below (*Remember the Result is not sorted in descending order*).



ShadowID	Call Based Rankings	Proximity Based Rankings	Overall Rankings
SID3	100%	100%	100%
SID1	25%	50%	87%
SID2	50%	62%	100%
SID4	0%	37%	37%
SID5	0%	50%	50%
SID6	0%	25%	25%
SID7	0%	25%	25%
SID8	0%	25%	25%
SID9	0%	25%	25%
SID10	0%	25%	25%
SID12	0%	12%	12%
SID13	0%	12%	12%
SID14	0%	12%	12%
SID11	0%	12%	12%
SID16	0%	12%	12%
SID18	0%	0%	0%
SID17	0%	0%	0%
SID15	0%	12%	12%
SID21	0%	0%	0%
SID22	0%	0%	0%
SID23	0%	0%	0%
SID24	0%	0%	0%
SID25	0%	0%	0%
	0%	0%	0%
	0%	0%	0%
	0%	0%	0%

Figure 12 Final Rankings Results from Computation Process

For Event Based Ranking Just Reset the system and Repeat Step 1 and 2 but in Step 2 select "Event Based Rankings"

And in Step5: Select Event Criteria e.g. both as shown below

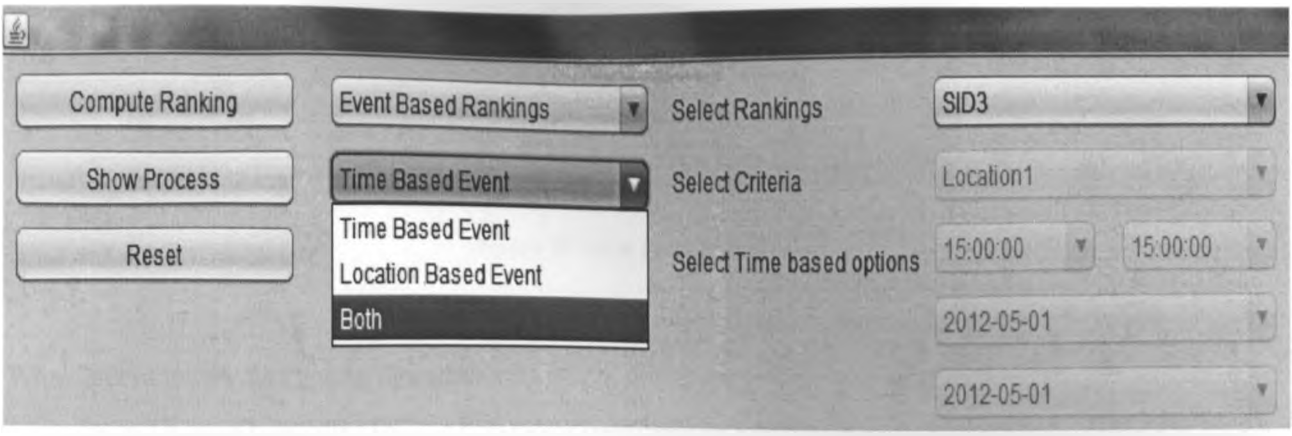
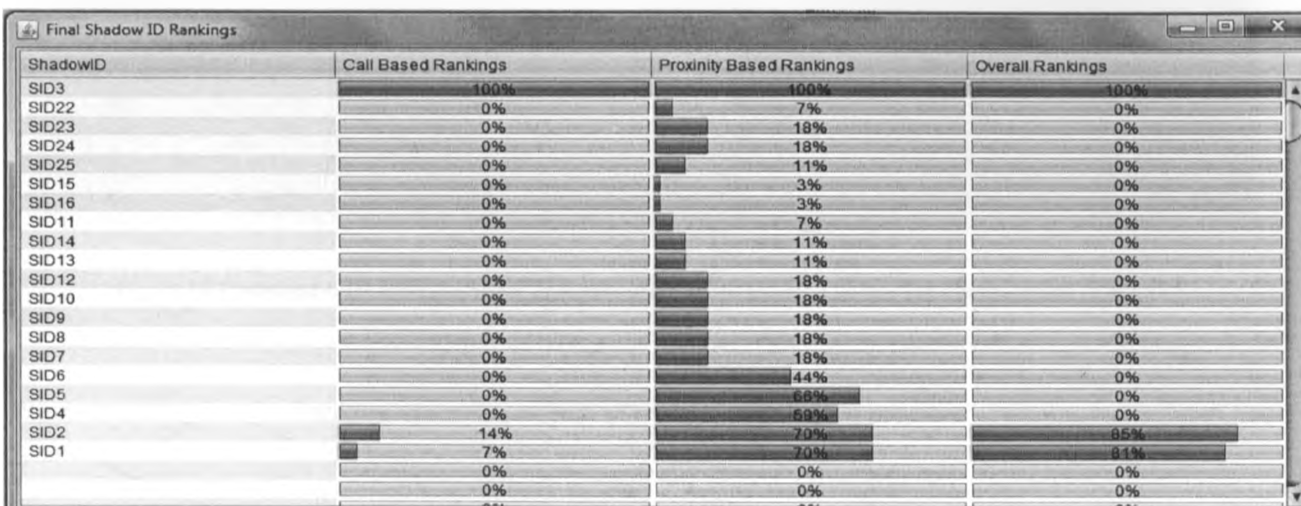


Figure 13 Event Based Condition Filtering Selection

And Click on the Compute Ranking Button to compute the results based on the specific event as needed to produce the result below.



ShadowID	Call Based Rankings	Proximity Based Rankings	Overall Rankings
SID3	100%	100%	100%
SID22	0%	7%	0%
SID23	0%	18%	0%
SID24	0%	18%	0%
SID25	0%	11%	0%
SID15	0%	3%	0%
SID16	0%	3%	0%
SID11	0%	7%	0%
SID14	0%	11%	0%
SID13	0%	11%	0%
SID12	0%	18%	0%
SID10	0%	18%	0%
SID9	0%	18%	0%
SID8	0%	18%	0%
SID7	0%	18%	0%
SID6	0%	44%	0%
SID5	0%	66%	0%
SID4	0%	59%	0%
SID2	14%	70%	85%
SID1	7%	70%	81%
	0%	0%	0%
	0%	0%	0%

Figure 14 Results Presentation from Events based selection

Additional Optional operations:

The Show Process function: This Command is used to show display the actual Outputs During the computation process and is shown below both before computing and after computing the Rankings.

The Show Process Button:

Figure 15 Show Process Button

When clicked Before the Ranking operation:

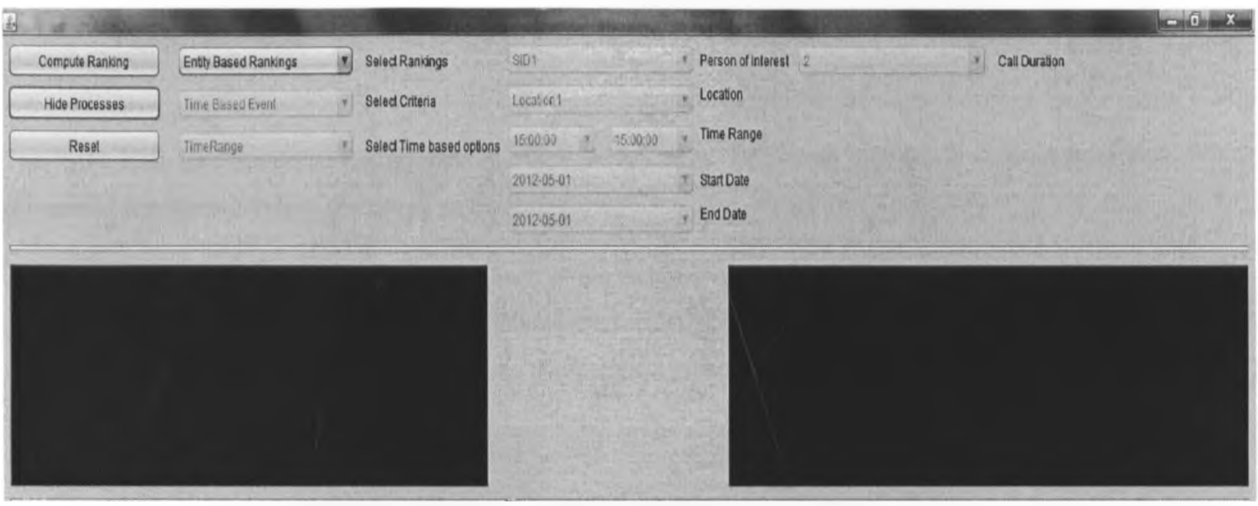


Figure 16 Prototype GUI with the Show Process option set to enabled before the computation process

When clicked After the Ranking operation or Just after the Computation process:

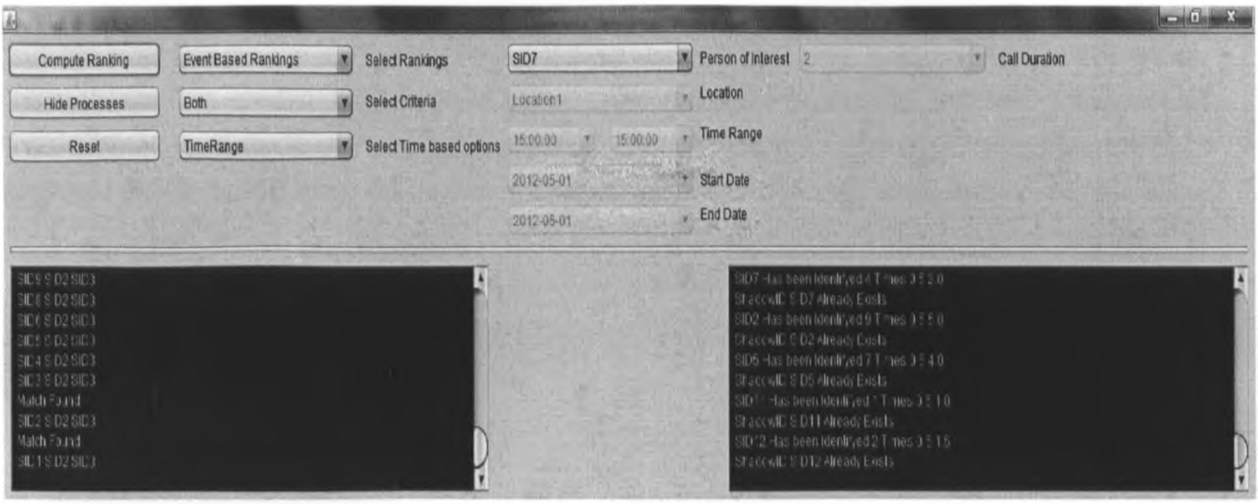
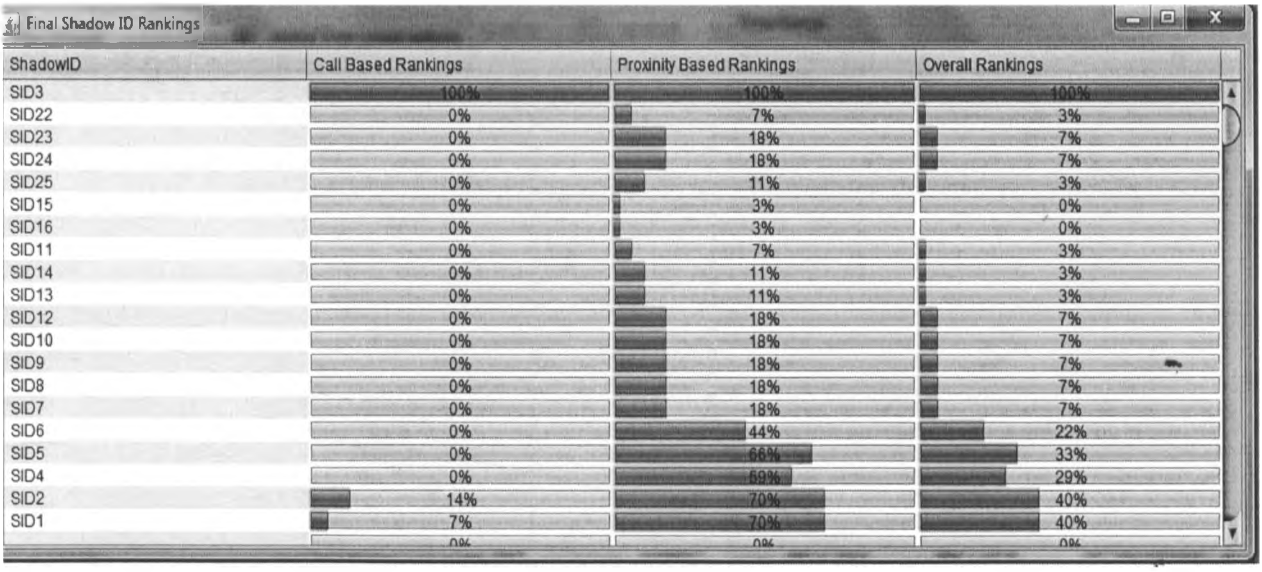


Figure 17 Prototype GUI with the Show Process option set to enabled after the computation process

This is the main java prototype Representation of the Surveillance Framework Strategy in actual Automation. Which the needed functions involves the use of an existing mySql Database to perform.

As shown in the results Form table presented such as the one below the results displayed are a list of all ShadowIDs and their various rankings based on the Criterion selected which is either the Entity Based or Event Based Rankings that have been identified at least once during the computation process.


Final Shadow ID Rankings

ShadowID	Call Based Rankings	Proximity Based Rankings	Overall Rankings
SID3	100%	100%	100%
SID22	0%	7%	3%
SID23	0%	18%	7%
SID24	0%	18%	7%
SID25	0%	11%	3%
SID15	0%	3%	0%
SID16	0%	3%	0%
SID11	0%	7%	3%
SID14	0%	11%	3%
SID13	0%	11%	3%
SID12	0%	18%	7%
SID10	0%	18%	7%
SID9	0%	18%	7%
SID8	0%	18%	7%
SID7	0%	18%	7%
SID6	0%	44%	22%
SID5	0%	66%	33%
SID4	0%	59%	29%
SID2	14%	70%	40%
SID1	7%	70%	40%
	0%	0%	0%

Figure 18 Results Presentation 3

ith real world data this would have been the list of people who are most likely Socially Related to a particular erson of Interest or Associated to a particular Event of Interest in which an Identified individual is known to be voived and it would also show their Likelihood probabilities of being involved or associated to either the POI or e Event in which the POI was involved This is based purely on the Framework model developed in this research.