International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.4, July 2012

DOMINANT FEATURES IDENTIFICATION FOR COVERT NODES IN 9/11 ATTACK USING THEIR PROFILE

S.KARTHIKA
ANNA UNIVERSITY
sk_mailid@yahoo.com

A.KIRUTHIGA
ANNA UNIVERSITY
kiruthiga312@gmail.com

S.BOSE
ANNA UNIVERSITY
sbs@cs.annauniv.edu

ABSTRACT

In recent days terrorism poses a threat to homeland security. The major problem faced in network analysis is to automatically identify the key player who can maximally influence other nodes in a large relational covert network. The existing centrality based and graph theoretic approach are more concerned about the network structure rather than the node attributes. In this paper an unsupervised framework SoNMine has been developed to identify the key players in 9/11 network using their behavioral profile. The behaviors of nodes are analyzed based on the behavioral profile generated. The key players are identified using the outlier analysis based on the profile and the highly communicating node is concluded to be the most influential person of the covert network. Further, in order to improve the classification of a normal and outlier node, intermediate reference class R is generated. Based on these three classes the most dominating feature set is determined which further helps to accurately justify the outlier nodes.

KEYWORDS

Social Network Analysis (SNA), Terrorism, Behavioral profile, Outlier Analysis, Dominant feature set.

1. INTRODUCTION

An event that brought a worldwide attention towards terrorism is the unforgettable 9/11 disaster [14] [15]. This provoked a need for awareness on anti-terrorism based national security. From then on a lot of research has been done on the terrorist networks. This covert network is seen as a social network with a lot of secrecy and influence. It is supposed to be covert or hidden but still has to manage the communication between them periodically. There should be some structure maintained within this network at the higher levels and at the lower hierarchy, they manage a cell structure.

In a covert social network one of the critical problems is to identify a set of key players who are highly influential. The problem of determining the importance of nodes is resolved based on the researches done in node centrality, group centrality and structural measures. The covert networks emphasizes on the importance of the links which presents the relations among the nodes [12]. When a set of nodes are given, automatically identifying the key players helps significantly in homeland security issues. Generally the characteristics of a person are understood based on his/her behavior. These behaviors can be represented in the form of a profile.

DOI : 10.5121/ijnsa.2012.4406

The profile generation uses the semantic graph as its input. A semantic graph is a graph that is used to represent semantic structure in terms of nodes and relations between them. A semantic structure has two nodes that are linked at least by one relation. A node can be any type of object like a person involved in an activity or a location or an event etc. And the link is the relation between the nodes. In a semantic graph the links can exist between different types of nodes and there can be multiple relationships i.e. heterogeneous relations between heterogeneous nodes. For example, the nodes a and b can be colleagues as well as neighbors. Hence such a semantic graph can be called as Multi-Relational Networks (MRN).

The sample 9/11 network such as the one shown in Figure 1 is an MRN that represents hijackers, other associates and the locations involved in this attack as nodes along with multiple relations between them as links.

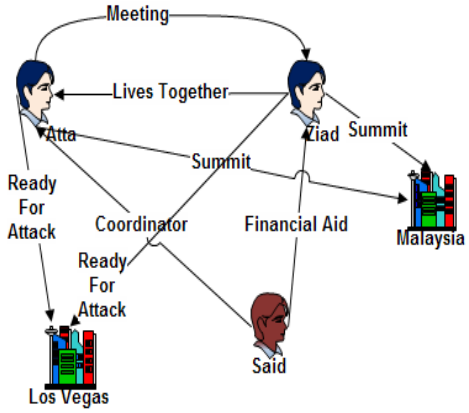

Figure 1. Sample MRN of 9/11 Attack

In this paper an efficient technique for analyzing the behavior of the nodes is presented by generating the profile of each node in the MRN. The profile illustrates the importance and the contribution of that specific node to the event. The links in the MRN generates various paths among the nodes and these paths are condensed to different formats using the variable relaxation approach. Many efficient selection strategies form path types from the condensed paths. The profile is a collection of such path types that describes specific characteristic about the node. This overcomes the issues of the earlier methods like the ensemble problem in which the structural equivalence of the nodes is not efficient enough to determine the key player. Secondly the goal problem where not all the nodes with high centrality values are key players and the distantly connected nodes have less effect on the influential nodes[1][10].

The remainder of this paper is structured as follows. Section 2 provides the various supporting works for the developed system. Section 3 illustrates the problem statement along with the assumptions made and Section 4 explains about profile generation framework. In Section 5 evaluation of the SoNMine system is performed and in the Section 6 the paper is concluded with the future work.

2. RELATED WORK

The semantic graph has been used by Shou-de Lin et.al [2] in an unsupervised framework UNICORN, to generate a profile from which suspicious nodes could be detected and uses a novel explanation system to verify the profiles using natural language processing. Shou-de Lin et.al [3] has also used the interestingness measure to determine the rarity using the node path

and node loop discovery strategies. The same author has implemented the interestingness measure using rarity analysis for the bibliography dataset in [4].

Stephen P.Borgatti[1] discusses about the key player problem which determines the highly important node set in the given network and a cut set of nodes that could fragment the entire network into individual entities. This method uses the distance based metrics to determine the centrality of the nodes and maintains the minimum size for the subset to defragment the network.

Valdis E.Krebs [5] has analyzed the network of 9/11 attack and has used the basic centrality measures to determine the importance of nodes. The author has also focused on identifying the task and trust ties between the conspirators by using the shortcuts and without using it.

Mohammad Al Hassan et.al [6] has focused on link prediction from the supervised learning perspective. The author handles the link prediction problems using classification techniques and uses the information gain, gain ratio and average rank as performance metrics. The author faces the problem in finding trained samples due to the incompleteness and fuzzy boundaries.

Bin Zhao et.al [7] proposes the use of relational Markov networks to describe the entities and the relations among them in an affiliation network. The author has used Profile In Terror (PIT) data base to study the entity and relationship labeling. The author faces the problem in using a supervised database as the random sampling created falls into different subsets.

Lei Zou et.al [8] proposes a sub- graph matching query system in which the similarity of given query is compared to that of the target query using score variable. This system has extensive scalability and quick response time.

Stephen D .Bay et.al [16] proposes the distance based, nested outlier algorithm which uses the neighbor score as the pruning factor. It is being compared with the Ramasamy et.al [17] [18] outlier algorithm which uses the Birch algorithm for clustering. It then determines the upper and lower bounds of the each cluster through which presence of outlier is found in each cluster. The problem of mentioning the number of clusters and outlier detecting in sparse data is overcome in [16].

3. PROBLEM DEFINITION

In this paper, we focus on identifying the key players who are involved in different roles, when compared to others in a covert network based on the profile which is generated using the selection strategies and the path types. To overcome the false positive problem, we use the outlier detection methodology for discovering the highly communicating nodes and the nodes with dense neighbors in the covert network.

We assume that the dataset collected is complete and the time order of the occurrence of the event is not important. We also assume that the social network under consideration is a single community and not heterogeneous. Based on these assumptions the unsupervised framework uses the feature value computation for judging upon the key players.

4. PROPOSED SYSTEM - SoNMINE

The nature of covert networks is to be dynamic in real world. Hence, this property makes the supervised system very expensive and sensitive to human bias. On the other hand, an unsupervised system can efficiently identify the adaptability of hidden nodes to a new domain, as it doesn't demand any prior training to a new set of rules. Figure 2 represents the overall information flow in SoNMine system.

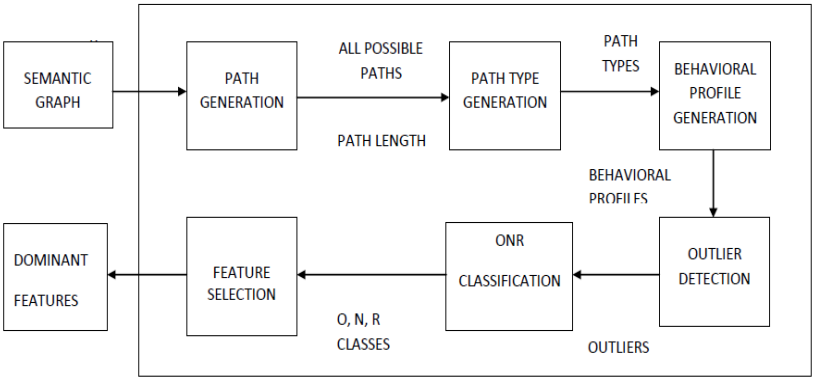

Figure 2. Framework of SoNMine

The unsupervised framework of SoNMine is modeled into three phases Firstly behavior profile generation based on path type and contribution values, secondly, outlier detection based on the frequency of communication, distance between the nodes and by further enhancing the classification based on the intermediate reference class. Thirdly, the dominant feature selection is done based on the contributing and non-contributing categories.

In the first phase all possible 4-step paths are generated from the semantic graph. The paths representing same type of activities are condensed using the variable relaxation approach and called as path type.

The profile is generated using the selection strategies. The selection strategies have been considered here because it helps us to choose a path type which is collection of paths based on which the profile is generated [2]. These strategies are used basically in two circumstances namely path oriented selection strategy and constraint oriented selection strategy. The path selection strategy is more related to the syntax of a path and the constraint strategy is based on the criteria with which a path belongs to a specific path type.

Definition of selection strategies:

A selection strategy $S(V,R,E^{-1})$ is a path where V is a finite set of nodes, R is a finite set of relations and E is a finite set of edges $E \subseteq VxRxV$. E^{-1} is a finite set of inverse edges such that $(v_2\ R\ v_1) \in E$.

The Table 1 shows various types of selection strategies employed in this paper for developing the profile.

Table 1. Selection Strategies

SELECTION STRATEGIES	**PATH TYPE FORMAT**
1. PATH BASED	
Same Sequence of Relation	x(?,?)^y(?,?)^z(?,?)
Single Relation	x(?,?)^x(?,?)^x(?,?)
Loop based	?(x,?)^?(?,?)^?(?,x)
2. CONSTRAINT BASED	
Selection Node	?^?(x,?)^? **Or** ?^?(?,x)^?
Exclusion Node	?^?(!x,?)^? **Or** ?^?(?,!x)^?
Selection Relation	?^x(?,?)^?
Exclusion Relation	?^!x(?,?)^?
Score based	Highest score: x(?,?)^y(?,?)^z(?,?) Second highest score: x(?,?)^y(?,?)^?(?,?)
Path length based	Maxi relations: 4 x(?,?)^y(?,?)^z(?,?)^a(?,?)

The same sequence of relations has all the paths with the same type of consecutive relations as elements in the path and it's defined as

Definition: A selection strategy S (V, R, E^{-1}) is a path where V is a finite set of nodes, R is a finite set of relations such that $R \subseteq r_1, r_2, .., r_n$ in the same sequence among the V and E^{-1} is a finite set of inverse edges such that (V_2 R V_1) where $E \subseteq VxRxV$.

The single relation strategy has paths of only single type of relation among the nodes. It's defined as

Definition: A selection strategy S (V, r, E^{-1}) is a path where V is a finite set of nodes, r is a single relation such that $R \in r$ and E^{-1} is a finite set of inverse edges such that (V_2 R V_1) where $E \subseteq VxRxV$.

The loop based selection strategy has the paths which starts with a node and ends with the same node. It's defined as

Definition: A selection strategy S (v, R, E^{-1}) is a path where v is the starting and ending node of the path where $v \in V$, R is a finite set of relations and E^{-1} is a finite set of inverse edges such that (v_2 R v_1) where $E \subseteq VxRxV$.

The exclusive node based strategies chooses the paths such that the path doesn't have a mentioned node as its element.

Definition: A selection strategy S (!v,R,E^{-1}) is a path where V can be any node other than v , R is a finite set of relations and E^{-1} is a finite set of inverse edges such that (v_2 R v_1) where $E \subseteq VxRxV$.

The selection node strategy forms the path type with the paths having a specific node whose occurrence is at least once in the path.

Definition: A selection strategy S (v^+,R,E^{-1}) is a path where v ∈ V and v occurs at least once , R is a finite set of relations and E^{-1} is a finite set of inverse edges such that (v_2 R v_1) where E⊆VxRxV.

The exclusive relation based strategies chooses the paths such that the path doesn't have a mentioned relation as its element.

Definition: A selection strategy S (V, !r, E^{-1}) is a path where V is a finite set of nodes, R can be any relation other than r and E^{-1} is a finite set of inverse edges such that (v_2 R v_1) where E⊆VxRxV.

The selection relation strategy forms the path type with the paths having a specific relation whose occurrence is at least once in the path and its defined as

Definition: A selection strategy S (V,r^+,E^{-1}) is a path where V is a finite set of nodes , r ∈ R and r occurs at least once and E^{-1} is a finite set of inverse edges such that (v_2 R v_1) where E⊆VxRxV .

The path length based selection strategy is a very needful aspect for generating the profile because it determines the length of the paths that should be considered in the path types [13]. We assume that all the paths in the path type are of uniform length. Hence only, the inverse relation is to be used in path generation. The limits of the path length is curtailed because farther the node it has lesser impact on the starting node. So if a node is distantly connected then it is less influential on the source node. The path length for 9/11 dataset is limited to four.

The score based strategy assigns a score for each path based on the similarity of the sequence of relations/labels in the path with that of the target path [8]. All the paths with the second highest score are also considered when the path type is determined. The Table 1 shows the various selection strategies and their path type formats. The path type is obtained by reducing the path using variable relaxation approach [2].

The contribution value of each node to the specific activity is determined using the random variables s and pt. The random variables are used to generate a single path output for a path type. The variable s represents the starting node of the selected path and variable pt represents the path type of the chosen path. The dependency of each node is determined based on the relative frequency with which a particular path is associated to a specific node. The values are computed using the statistical dependence measure like mutual information (MI) and point wise mutual information (PMI) and normalized PMI (NPMI). These are estimated for the node as well as for the path. The path NPMI given in Eq. 1 represents the significance of the path with the starting node as s in the path type pt and the node NPMI given in Eq.2 shows the contribution of the node to the path type pt.

$$\text{Path PMI}(S=s, PT=pt) = \log\left(\frac{(S=s, PT=pt)/|Path|}{(|S=s|/|Path|) * (|PT=pt|/|Path|)}\right) \quad \text{..... Eq.1}$$

$$\text{Node PMI}(S=s, PT=pt) = \log\left(\frac{\left(\frac{S=s, PT=pt}{|S=s|}\right) * \left(\frac{1}{|Node|}\right)}{(1/|Node|) * \left(\sum_{k \in node} \left(\frac{|S=k, PT=pt|}{|S=k|}\right)\right)/|Node|)}\right) \quad \text{.....Eq.2}$$

Now that the contribution values have been computed, the behavioral profile can be structured. The profile has the path types and the contribution value of the node in which it was involved. The following pseudo code describes the above mentioned two phases of SoNMine formally:

```
Behave_Profile(M,mc,k){
//M is a MRN<V,E,L>
//mc is a meta-constraint for selecting path types
//k is the maximum path length for path types
var profile array[|V|,|PT|] of double
Paths=GeneratePaths(M);
K=FindFrequentPathLength(Paths);
Paths=InversePaddedPaths(Paths,k);
Paths=TrimTok(paths,k) ; //cutshort all paths to length k
PT=extractPathTypes(Paths,mc,k);
for n ∈ V
        for pt ∈ PT
                Profile[n,pt]=getContributionValues(paths,n,pt);
return getOutliers(profile,k);}
```

```
Function GeneratePaths(M){
        AdjacencyMatrix A=CreateAdjacencyMatrix(M);
        for n ∈ V
                Paths=Paths ∪BreadthFirstSearch(n, A);
                //Breadth First Search takes n as source node and all others as destination node
        Return Paths;}
```

```
Function FindFrequentPathLength(Paths){
Var k; Path p;
//count() counts the number of relationships in a path
        For p ∈ Paths
                K[Count(paths)]++;
                k = max(k);}
```

```
Function InversePaddedPaths(Paths,k){
Path p;
Inversepath ip;
//inverse path ip of p has the end node of p as starting node
If pathLength(p)<k
        Append(ip,p);}
```

```
Function extractPathTypes(Paths,mc,k){
    PT=PT ∪PathOrientedPathtypes(Paths);
    PT=PT ∪ConstraintOrientedPathtypes(Path);
Return PT;}
Function getContributionValues(Paths,s,pt){
//pointwise mutual information calculation
        pmiValue=Pmi(paths,s,pt);
//normalized pointwise mutual information
        npmiValue=Npmi(pmiValue,path,s,pt);
    Return npmiValue;}
```

The profile of the node is generated based on the above discussed selection strategies [9]. The Table 2 below discusses about the profile of the node Nawaf who has been involved in various activities like car visit, meeting, ready for attack with higher PMI based contribution values.

Table 2. Profile of NAWAF

```
Profile Of :Nawaf
[Ready For Attack,Ready For Attack,Car Visit With,Meeting,] = 0.9662304535905736
[Nawaf,Nawaf,] = 2.4703078503668476
[Ready For Attack,Car Visit With,Meeting,Meeting,] = 1.6718001541490761
![Meeting,Meeting,Meeting,Meeting,] =

-2.1546649629174235
[Car Visit With,Car Visit With,Meeting,Meeting,] = 0.02796081499764329
```

Now the SoNMine framework has to detect the influential node/actor in the network using the outlier detection methodology. It employs the distance based, nested outlier algorithm which uses the average distance score as the pruning factor. It concludes that a node is an outlier if it is among the top n highly communicated entity [16]. The following pseudo code explains the outlier detection to identify the key players:

```
Get_Outliers(profile,k){
//let k be the max number of neighbors to a node
//Pathtypes considered as Features & contribution values become feature values
//Set of features denoted as F
//Distance(A,B) is the Euclidean distance between A & B
F=ConvertToFeatureValues(PT,Profile)
Cutoff=0
Outliers=ϕ
Let the set of nodes as D
Divide D into B block of elements
while B get-next-block(D)  // load a block of examples from D
 Neighbors(b)= ϕ ; for all b in B

for each d in D
        {for each b in B, b != d
          { if |Neighbors(b)|< k or distance(b,d) < maxdist(b,Neighbors(b))
                {                    Neighbors(b) =Closest(b,Neighbors(b)∪d, k)
                                  if
                            score(Neighbors(b),b) < c
                } remove b from  B
            }          }
 Outliers=Top(B ∪ Outliers,n) // keep only the top n outliers
// the cutoff is the score of the weakest outlier
Cutoff= min(score(o)) for all o in O ;
return Outliers;}
```

In case of outlier detection the SoNMine system shows the nodes KAM, Atta, Nawaf as the outliers. The Figure 3 shows the three outlier nodes Atta, KAM & Nawaf found from the 9/11 terrorist network. It's determined based on the contribution values of the nodes towards a lot of activities. The outlier nodes have been further justified by analying the features manually with the dataset. It's found that the following features makes these nodes as highly influential when compared to others

(i) These nodes are the few who were given the flight training in different aviations

(ii) These nodes were well trained personally by the chief trainer of Al-queda

(iii) All these nodes attended the summit that was conducted in Malaysia and Kola Lumpur

(iv) These nodes had many visits in and out of US and were ready to receive the MUSCLE(other 13 hijackers) in their respective destinations like WA, NY and LA.

(v) KAM is the master mind behind the USS Cole which is considered as the mock attack before 9/11

(vi) Nawaf is one of the veteran associate who seconded and planned for this attack along with the master mind KSM and Osama Bin Laden

(vii) Nawaf is the only node who had moved to 16 different locations within US for making different purposes

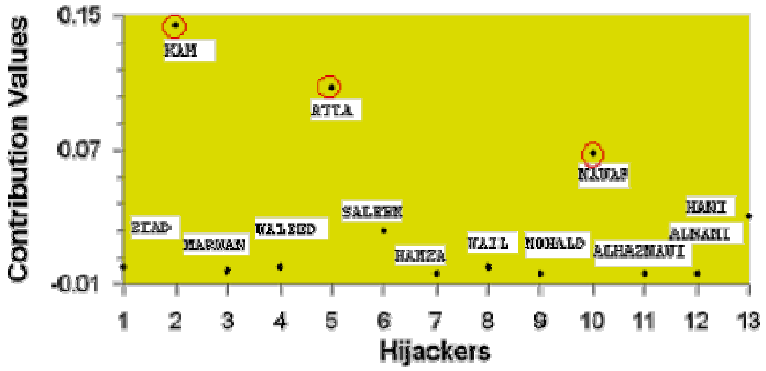

Figure 3. Outlier nodes Atta, KAM and Nawaf

The other way of presenting justification for the identified outlier nodes is to determine the dominant feature set that makes the node to be unique.These features are found by characterizing the nodes into different classes like Normal and Outlier. Inorder to have a more accuracy in expalnation a third class namely Reference class is introduced. This class posses those nodes that are close enough to the outlier, but still has sufficient difference from the outliers.The following pseudocode formally expalins the identification of reference class based on the distance between the nodes and the dominant feature set [19].

```
BKK _Algorithm (Outliers O, Profiles PF)
{
For each o ∈ O
 GroupClasses{}= Function ORNgroupClassifier (Outliers, Profiles,k);
 Classes{}=Function ORNclassifier(GroupClasses,Profiles);
//k= FrequentPathLength
 CategoricalProfiles=Function CategoricalConverter(Profiles);
DominantFeatures=Function FeatureSelector (Classes,CategoricalProfiles);
Return DominantFeatures;
}
Function GroupClass ORNgroupClassifier(Outlier,Profiles)
{
For each node n ∈ Profiles and node ≠ o
```

```
//φ is collection of φn
φn=Find difference in all features of n & o;
                    // φn has difference of all dimensions of node n & o
For each dimension d ∈ Profiles
ksList{}=Top( φ,k); //d ksList's has the top k closest points in dimensions 1..d
For each node n ∈ Profiles
Presencen=[P1n,P2n,…Pdn] ;
                    // Pdn=1 if ksList for dimension d has n, else Pdn=0
DS={n}, if n ∈ ksList{};
For each node n ∈ DS
      Calculate CFDistance; //CFDistance is the distance among common features of two entities
          Sort KS based on CFDistance;
          ReferenceList=Top(DS,k);
          O{}=outlier;
          R{}=ReferenceList;
          N{}=Profiles-O-R;
          GroupClasses={O,R,N};
          Return GroupClass;
}
```

```
Function                    Classes
ORNclassifier(GroupClasses , Profiles)
{
          O{}=ExtractOutliers(GroupClasses);
          R{}=ExtractReferences(GroupClasses)-O{};
          N{}=Profiles-O{}-R{};
          Classes={O,R,N};
          Return Classes;
}
```

```
Function CategoricalConverter(Profiles)
{
For each dimension
For each node n ∈ Profiles


 If exactequal(node n's dimensionalvalue,0)
          Category=NC;
          Else
          Category=C;
          Return CategoricalProfiles;}
```

```
Function
FeatureSelector(Classes,CategoricalProfiles)
{
          TrainingSet=Merge(Classes,CategoricalProfiles);
          SoNMineTree=CreateDecisionTree(TrainingSet);
          DominantFeatures=PostPruning(SoNMineTree);
          Return DominantFeatures;
}
```

5. EVALUATION OF SONMINE

SoNMine generates various path types for the 9/11 attack through a synthetically generated dataset and by following the above mentioned selection strategies which is illustrated in Table 3. SoNMine finds the influential node based on their behavioral profile which has contribution value of the nodes towards a specific behavioral path type. UNICORN is an already existing framework for the above mentioned purpose [2]. It has done its evaluation based on the relation only selection strategy and used the path length as 4 [11]. The following Table 4 compares the performance of UNICORN with SoNMine based on the score based selection strategy and has shown the results for a sample of nodes and path types. By using the score parameter it not only chooses the path that has the exact match to target path type but also considers the second highest matching paths. This increases the number of paths considered for the contribution value computation and increases the contribution value of the nodes towards the path type. Hence this system provides more accurate contribution values and detailed profile for the nodes.

The Figures 4.a and 4.b shows the performance graphs of both the systems for the nodes Nawaf and KAM. The SoNMine system outperforms the UNICORN based on the PMI values for the nodes. Similar results have been obtained for the other nodes in the data set.

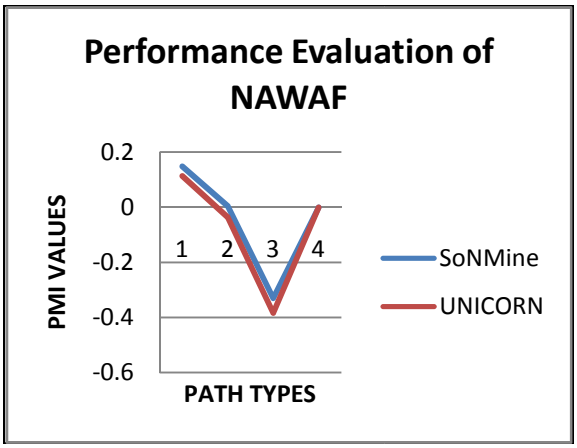

Figure 4.a. Performance Of The Node Nawaf

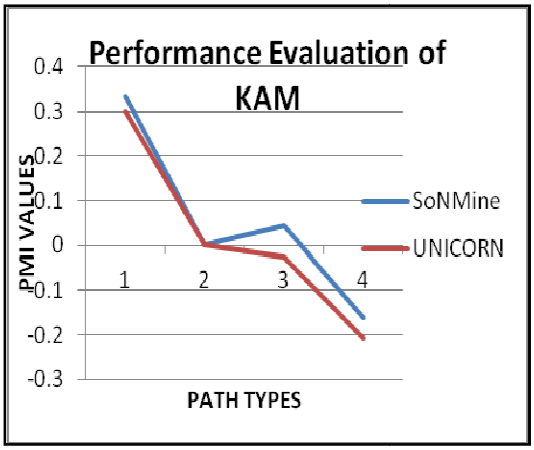

Figure 4.b. Performance Of The Node KAM

Table 3. Path types with and without variable relaxation approach for all selection strategies

SELECTION STRATEGY	PATH GENERATED	PATH TYPE WITH VARIABLE RELAXATION APPROACH
Same sequence of Relation	BrotherOf (saleem,nawaf) ^ReadyFor Attack(nawaf,hani)^CarVisitWith(hani,atta)	Brother Of(?,?)^Ready For Attack(?,?)^CarVisitwith(?,?)
Single relation	Meeting(atta,ziad)^Meeting(ziad,said)^ Meeting(said,atta)	Meeting(?,?)^Meeting(?,?)^Meeting (?,?)
Loop based	Meeting(hamza,marwan)^Pilot_Training(marwan, ziad)^Friends(ziad,atta)^ ReadyForAttack(atta,hamza)	?(hamza,?)^?(?,?)^?(?,?)^?(?,hamza)
Selection node	Meeting(saleem,nawaf)^Attack(nawaf,hani)^Lives In(hani,salem)	?(saleem,?)^?(?,?)^(?,?)
Exclusion node	Atta	-
Selection relation	BrotherOf(ahmed,hamza)^Attack(hamza,hani)^Me eting(ziad,hani)	BrotherOf(?,?)^?(?,?)^(?,?)
Exclusion relation	Meeting	BrotherOf(?,?)^ ReadyForAttack(?,?)^ CarVisitWith(?,?)^FigthsWith(?,?)^ LivesWith(?,?)
Score based	Meeting(saleem,nawaf)^Attack(nawaf,hani)^Lives In(hani,salem) Meeting(ahmed,hamza)^Attack(hamza,hani)^Meet ing(ziad,hani)	Meeting(?,?)^Attack(?,?)^LivesIn(?, ?) Meeting(?,?)^Attack(?,?)^Meeting(? ,?)
Path length based	**Before:** BrotherOf (saleem,nawaf)^ ReadyForAttack(nawaf,hani)^ CarVisitWith(hani,atta)^Meeting(atta,ziad)^Live With(Ziad,Marwan) **After:** BrotherOf (saleem,nawaf)^ ReadyForAttack(nawaf,hani)^ CarVisitWith(hani,atta)^ Meeting(atta,ziad)	**Before:** BrotherOf (?,?)^ ReadyForAttack(?,?)^ CarVisitWith(?,?)^Meeting(?,?)^Liv esWith(?,?) **After:**BrotherOf(?,?)^ ReadyForAttack(?,?)^ CarVisitWith(?,?)^Meeting(?,?)

Table 4. Performance evaluation of SoNMine and UNICORN based on score selection strategy

PATH TYPE With Score(SoNMine) / Without Score (UNICORN)		NAWAF	KAM
[Ready For Attack,Ready For Attack,Car Visit With,Meeting,]	SoNMine	0.1481473023293475	0.33503852980338
	UNICORN	0.1136358	0.30136954911311
[Car Visit With,Car Visit With,Meeting,Meeting,]	SoNMine	0.0042870924813409	
	UNICORN	-0.0360526086212633	
![Meeting,Meeting,Meeting,Meeting,]	SoNMine	-0.3303640456514111	0.04383482537727326
	UNICORN	-0.3842615	-0.02562824632747
[Saleem,]	SoNMine		-0.16119212347979
	UNICORN		-0.208235

The following Figures 5.a, 5.b and 5.c shows the Normal class and Reference class nodes for the outliers Atta, KAM & Nawaf respectively using the SoNMine system.

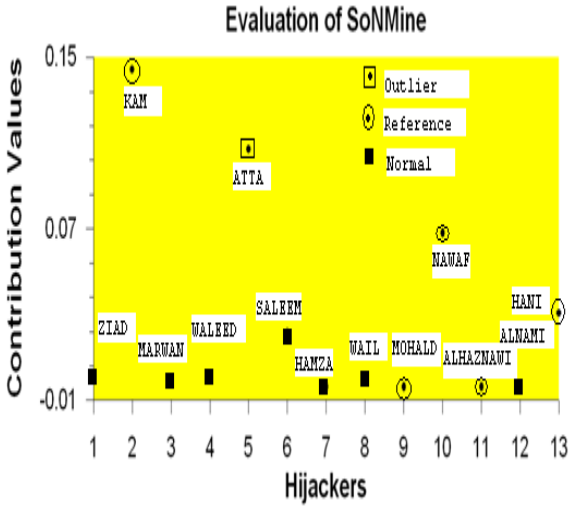

Figure 5.a. Reference and Normal Class nodes for Atta

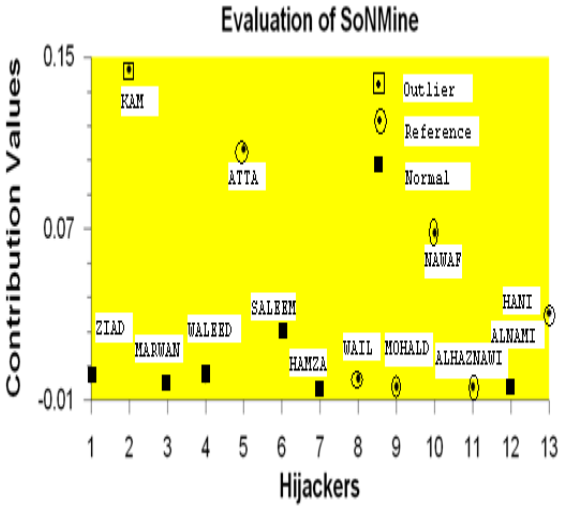

Figure 5.b. Reference and Normal Class nodes for KAM

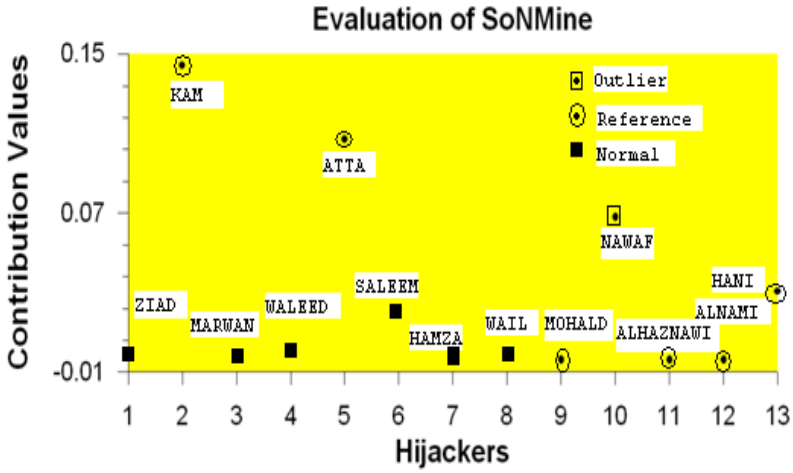

Figure 5.c. Reference and Normal Class nodes for Nawaf

From the above figure, the nodes KAM, Atta, Hani, Alnami, Alhaznavi and Mohald form the reference class and the other nodes are the members of normal class for the node Nawaf. The R class members KAM and Atta are closely related to Nawaf in the execution of the 9/11 attack as they take flight classes together in Huffman aviation and go for frequent car visits to the different target cities like NY, WA, LA. All the three were the members of meetings conducted in Malaysia and Kolalumpur. Hani was member of the Hamburg cell formed along with Nawaf. The rest three nodes were the Muscles who were received by Nawaf when they reached U.S. Similar to Nawaf , for other nodes also the reference class expalnation could be given based on the dominant feature set. The following Table 5 shows the 7 most dominanting features out of the 71 existing features.

Table 5. Dominant Feature Set

Feature Number	Feature Name
1	[READY FOR ATTACK,READY FOR ATTACK,CAR VISIT WITH,MEETING,]
4	![MEETING,MEETING,MEETING,MEETING,]
5	[CAR VISIT WITH,CAR VISIT WITH,MEETING,MEETING,]
23	[MEETING,FLIGHT PRACTICE,CAR VISIT WITH,CAR VISIT WITH,]
66	[MOHALD,MOHALD,]
67	[ALHAZNAWI,ALHAZNAWI,]
6	[KAM,KAM]

6. CONCLUSION

In this paper we have discussed about an unsupervised system called as SoNMine which identifies the key players in the 9/11 covert network. The key players are realized based on the profile generated for the nodes. The profile has the various activities in which the nodes are involved in. These activities are called as path type which uses many selection strategies. These selection strategies are based on the relation types and the constraints. The importance of the node in a profile is found based on the contribution and dependency values. Based on these contribution values the system could identify the outlier nodes. In order to give a better

justification for the detected outliers a special class called as R(Reference) is introduced. Using the O,R,N classes the decision tree is constructed and pruned to get the dominant featuer set. Since it's an unsupervised system there is no need for any training data set but the relations that are to be considered should be already selected and it's relaized as a limitation of this system. To further improve the performance of the SoNMine system the time order of the event occurrence could be included in the data set.

7. REFERENCES

[1] Borgatti, S. "Identifying sets of key players in a social network," *Comput Math Organiz Theory*, vol.12, pp. 21–34, 2006.

[2] Shou-de Lin and Hans Chalupsky "Discovering and explaining abnormal nodes in semantic graphs," *IEEE Transactions on knowledge and data engineering*, vol. 20, no. 8, pp.1039-1052, 2008.

[3] Shou-de Lin and Hans Chalupsky, "Unsupervised Link Discovery in Multi-relational Data via Rarity Analysis," *Third IEEE International Conference on Data Mining*, 2003.

[4] Shou-de Lin and Hans Chalupsky, "Using Unsupervised Link Discovery Methods to Find Interesting Facts and Connections in a Bibliography Dataset," *ACM SIGKDD Explorations Newsletter*, 2003.

[5] Valdis E. Krebs, "Mapping Networks of Terrorist Cells," *Connections*, vol.24, no.3, pp.43-52, 2002.

[6] M. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," *In Proceedings of the Workshop on Link Discovery: Issues, Approaches and Applications*, 2005.

[7] B. Zhao, P. Sen, and L. Getoor, "Entity and relationship labeling in affiliation networks," *In ICML Workshop on Statistical Network Analysis*, 2006.

[8] Lei Zou, Lei Chen, and Yansheng Lu, "Top-k subgraph matching query in a large graph," *In PIKM '07: Proceedings of the ACM first Ph.D. workshop in CIKM*, pp 139-146, New York, NY, USA, ACM,2007.

[9] R. N. Kocsis, "Criminal Profiling: International Theory, Research, and Practice," *Totowa : Humana Press Inc*,pp. 169- 188, 2007.

[10] Nasrullah Memon, Nicholas Harkiolakis and David L. Hicks, "Detecting High-Value Individuals in Covert Networks: 7/7 London Bombing Case Study," *Computer System and application*, 2008.

[11] Nasrullah Memon, Henrik Legind Larsen, David L. Hicks, and Nicholas Harkiolakis, "Detecting Hidden Hierarchy in Terrorist Networks: Some Case Studies", *ISI 2008 Workshops*, pp. 477–489, 2008.

[12] Nasrullah Memon and Uffe Kock Wiil, Reda Alhajj, Claus Atzenbeck, Nicholas Harkiolakis, "Harvesting covert networks: a case study of the iMiner database", *Int. J. Networking and Virtual Organisations*,2011.

[13] Robert S.Boyer and J Strother Moore, "A Fast String Searching Algorithm," *Communications of the ACM*, vol. 20, no. 10, 1977.

[14] Thomas H.Kean,The 9/11 Commission Report. USA:National Commission on Terrorist Attacks upon the United States, 2004.

[15] "Responsibility for September 11 attacks," *http://en.wikipedia.org/wiki/Responsibility_for_the_September_11_attacks*, November 2010.

[16] Stephen D .Bay, Mark Schwabacher, “Mining Distance Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule”, *SIGKDD ’03,* Washington, DC, USA, August ,2003.

[17] S. Ramaswamy, R. Rastogi and S. Kyuseok , "Efficient Algorithms for Mining Outliers from Large Data Sets." *In Proceedings of the ACM Special Interest Group on Management of Data*, p.427-438,2000.

[18] T. Zhang, R. Ramakrishnan and M. Livny , "Birch: An efficient data clustering method for very large databases." *In Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD)*, Montreal, Canada, p.103-114,1996.

[19] Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, CA: Morgan Kaufmann Publishers, 2003.