# PATH PREDICTION AND PATH DIVERSION IDENTIFYING METHODOLOGIES FOR HAZARDOUS MATERIALS TRANSPORTED BY MALICIOUS ENTITIES

Rakesh Nune

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Civil Engineering

Dr. Pamela Murray-Tuite, Chair
Dr. Kathleen L. Hancock
Dr. Montasir Abbas

December 5, 2007
Falls Church, Virginia

Keywords: Hazardous materials, Hijacking, Neural network, Pattern classification, Optimization, route algorithm

PATH PREDICTION AND PATH DIVERSION IDENTIFYING METHODOLOGIES FOR HAZARDOUS MATERIALS TRANSPORTED BY MALICIOUS ENTITIES

Rakesh Nune

# ABSTRACT

Safe and secure transportation of hazardous materials (hazmat) is a challenging issue in terms of optimizing risk to society and simultaneously making the shipment delivery economical. The most important safety concern of hazardous material transportation is accidents causing multiple causalities. The potential risk to society from hazmat transportation has led to the evolution of a new threat from terrorism. Malicious entities can turn hazmat vehicles into weapons causing explosions in high profile locations.

The present research is divided into two parts. First, a neural network model is developed to identify when a hazmat truck deviates from its pre-specified path based on its location in the road network. The model identifies abnormal diversions in hazmat carriers' paths considering normal diversions arising due to incidents. The second part of this thesis develops a methodology for predicting different paths that could be taken by malicious entities heading towards a target after successfully hijacking a hazmat vehicle. The path prediction methodology and the neural network methodology are implemented on the network between Baltimore, Maryland and Washington, DC.

The trained neural network model classified nodes in the network with a satisfactory performance .The path prediction algorithm was used to calculate the paths to two targets located at the International Dulles Airport and the National Mall in Washington, DC. Based on this research, the neural network methodology is a promising technology for detecting a hijacked vehicle in its initial stages of diversion from its pre-specified path. Possible paths to potential targets are plotted and points of overlap among paths are identified. Overlaps are critical locations where extra security measures can be taken for preventing destruction. Thus, integrating both models gives a comprehensive methodology for detecting the initial diversion and then predicting the

possible paths of malicious entities towards targets and could provide an important tool for law enforcement agencies minimizing catastrophic events.

# ACKNOWLEDGMENTS

I am extremely grateful to my advisor Dr. Pamela Murray-Tuite for her constant support throughout my stay at Virginia Tech, even through her most difficult times. The encouragement along with the freedom I received helped me to evolve as a researcher.

My special thanks to Dr. Kathleen Hancock for taking extra effort to guide me in the right direction in the most crucial time. I am also thankful to Dr. Montasir Abbas for his invaluable suggestions. I am also very grateful to Susan Merten in Falls Church and Marija Telbis Foster for helping me through the complex administrative procedures at Northern Virginia Centre.

# DEDICATIONS

The work would not have been possible without the constant love and effort given by my family. I would like to convey special thanks to Neelima Ghanta for constantly critiquing my writing and presentation skills and then taking the pain of proof reading many of my 'primary' drafts inspite of her busy schedule. I am very grateful to the support given by my roommates, Bharath, Baba and, Kiran for keeping me sane after long days of work. The list would be incomplete without adding the names of Sandeep who made my stay at NVC real fun and, Nop who patiently guided me throughout my course work.

## Table of Contents

# List of Figures

# List of Tables

# CHAPTER 1: Introduction

Safe transportation of hazardous materials (hazmat) is presently an emerging area of interest due to a steady increase in their transported volumes over the years and the potential risk they pose to society. One class of hazmats can be defined as industrial products either required for operational processes or generated during the same. Hazmats are transported through all the modes of travel i.e. road, train, air and ships[1] of which roads have the highest share of this transportation. Hazmat carriers have a preset destination from a specific origin. The main concerns of hazmat transportation pertain to crashes which cause multiple casualties and long term effects[2]. In the case of accidents, the area affected and damage caused depend on the kind of hazmat, state of the hazmat, (liquid/gaseous) time of day, wind conditions. From the historical data, it can be observed that hazmat accidents are very rare occurrences, but when they do occur, they might result in damage to the environment, infrastructure, and, most importantly to humans. These potential damages should be involved in the risk quantification for hazmat transport. Route selection for hazmats could be estimated by use of comparative risk levels for each route rather than the absolute risk involved which simplifies the procedure for route calculation.

The routes chosen should minimize the risk of exposure to the people in the case of spills and accidents, from the economic perspective of the carrier, shorter paths are preferred. Thus, the final route selection is a function of travel distance and risk.

Intuitively, safer routes for hazmat carriers should be far from residential areas and should also allow the trucks to move at a reasonable speed to reach the destination. These two factors might not be the only guidelines for route selection, other factors like destination, or kind of hazardous material might also play a role.

Transportation of hazmats at safer distances from the cities and important places could be the key for preventing any catastrophic damage However, in most areas the highway network is intertwined with populated areas, which could lead to some precarious situations when the hazmat driver accidentally or intentionally (in case of hijacking) diverts from his standard route.

The present work is focused on the threat to society from malicious entities planning to use hazmat vehicles for destructive purposes. In this scenario, the malicious entity hijacks a hazmat truck from its pre-specified route moving towards its pre-specified destination, then reroutes the carrier towards his target. It is assumed that as the hijacker would like to maximize the possible destruction, air borne hazmats like ammonia or chlorine or explosive material are selected for this scenario, since they spread faster and have a larger impact.

After successfully hijacking the hazmat truck, the malicious entity moves away from the normal path to reach his target. The path that the hijacker might take between his hijack point and target might deviate from the normal path at the very start of the journey or might follow the normal path for some time and divert at the last moment, depending on the target's location. Detecting the diversion of the hazmat truck at the earliest point would help to identify the hazmat driver's intentions and alert law enforcement officials to take appropriate actions.

The probable paths that can be taken by these hijackers are of prime interest since extra security on the most probable path will likely stop the malicious entity and also minimize catastrophic damages that might occur due to explosion or a hazmat release.

The objective of this work is to develop a model for identifying the hijacking and subsequently to predict the possible paths taken by the malicious entity to reach his target. A neural network model is investigated to identify when a hazmat truck diverts from the normal route. The second part of the thesis involves predicting the probable paths taken by the hijacker towards his target once a diversion has been detected. The paths are predicted through a fractional linear program that allows tradeoffs between travel time and consequence.

The reminder of the thesis is organized as follows Chapter 2 details the previous research done on hazmat routing and includes a literature review of classification algorithms and corresponding applications, neural networks, and the structure and application of neural networks in various fields. Chapter 3 defines the problem in mathematical terms and the notation used. Chapter 4 describes the methodology for developing the parameters used in the neural network, estimating consequences for a given transportation network, and developing the mathematical formulation for a possible hijack scenario. Chapter 5 presents the application of these methodologies to the transportation network between Washington, DC and Baltimore. Chapter 6 presents the results of the neural network model and paths of the malicious entity towards the target. Finally, chapter 7 gives conclusions and recommendations, for future work.

# CHAPTER 2: Literature Review

The present chapter is organized as follows. Section 2.1 provides some statistics of hazardous material and several rules and regulations given by federal and state authorities for choosing routes for hazmat carriers. Section 2.2 details previous research for estimating consequence and safer routes using various methodologies followed by different researchers. Section 2.3 explains the theory behind pattern classification and various techniques used to perform the same. Section 2.4 provides background information on neural networks, various kinds of neural networks and training methods. Finally section 2.5 specifically concentrates on probabilistic neural networks (PNN's), their training methodology and various research works where PNN, were applied.

## *2.1 Background*

Liquefied petroleum, gasoline, explosives and chemicals are heavily toxic for inhalation and are 90% of all hazmat related goods[3]. According to the 2002 Commodity Flow Survey, 311,897 million ton miles of hazmats were transported in the USA in 2002. For example, ammonia was transported over 2674 million ton miles [4, 5].

In a recent (2007) incident in San Francisco, a gasoline truck carrying 8600 gallons of gasoline crashed on Interstate 580. The intense heat generated during the subsequent fire buckled Interstate 580 which fell onto Interstate 880 located below [6]. The incident would have had higher consequences if the fire occurred in a populated area. The incident gives a glimpse of the disastrous capacity of hazmat carriers.

Due to the dangerous nature of the substances, hazmat routing has acquired prime importance and strict regulations are enforced by state and federal agencies. The U.S. Department of Transportation, Pipeline and Hazardous Materials Safety Administration (PHMSA), Federal Motor Carrier Safety Administration and various other federal and local bodies have issued several regulations for integrating safety with hazmat routing.

These regulations ensure that hazmat trucks move away from populated areas to prevent heavy damage in case of spills [3, 7-9].

Routing for a hazmat can be divided into two steps. The first calculates the consequence for the network. The second step determines the route for the hazmat carrier based on a defined methodology, and, in most cases, minimizing the consequence and travel time are among the objectives.

## *2.2 Aspects of Hazmat Routing*

Various authors have developed methods to calculate consequence in a given route. The damage done by hazmats is widespread and often difficult to evaluate. The National Highway Institute has issued several regulations to determine the route for a hazmat carrier by minimizing the population which might be exposed conditioned by accident rates at those places[8].Harwood et al developed a procedure for estimating accident rates and release rates as a function of road type, area type and other data like traffic volume, and road design Release probabilities are also estimated from state and federal accident databases[10].

Garrido and Bronfman determined consequence by using the material specific radius of influence along the route to determine the possible population which will be affected and then converting population damage occurred to monetary terms [11].Kara et al focused on developing a methodology for the accurate estimation of consequence. The authors showed that consequence is counted twice at the intersection of two links in a road network, because the population exposed is included in both the links[12] .In the present work, for developing possible routes of malicious entity consequence is more of a perceived consequence rather than the exact consequence.

Different methodologies have been adopted to determine safe routes using the consequence values, travel time, accident rates, and various other risk minimizing factors. Frank et al developed a visual interface for reducing the number of people exposed on the routes which are travelled by the hazmat carrier. The system minimized the risk using complicated heuristic algorithms and finally visually displayed the solution on the real network[13].Carotenutoa et al developed a methodology for route estimation using equitable distribution of risk among the routes. The methodology distribute the risk of exposure to hazmat material by choosing routes which distributes the risk among geographical zones rather than concentrating on some sets of zones.The algorithm considers the risk propagation which results from closed paths due to hazmat carrier incidents [14]. Nembhard and White applied non-order preserving path selection, which chooses routes whose sub paths of the main paths, might not be optimal [15].Erkut and Ingolfsson developed three catastrophic avoidance models for hazmat route selection to minimize the exposed population and variance in consequence [16]. The same authors developed a model for hazardous route planning that considers the possibility of an accident anywhere on the link rather than considering the accident to affect the complete link[17]. Huang and Fery emphasized consideration of the tradeoff among the conflicting objectives of distance, cost, and population exposure for generating various paths [18].Erkut and Alp examined scheduling and routing taking in to distribution of accident rates, population exposure and travel time as a function of time of day. Risk of transport is minimized based on the duration of the trip and was verified with four realistic real world situations[19].Chang et al developed a multi objective methodology for finding non dominated paths considering the variations as probability distributions. The algorithm was tested on both smaller and larger networks to see the effects of parameter changes [20].

Carotenuto et al addressed the issue of both scheduling and routing of hazmat shipments minimizing the delay of transportation and also minimizing the total risk of transport using tabu search[21].Meng et al, similar to previous work, addressed the issue of scheduling and vehicle routing taking into consideration additional real world constraints such as service time, waiting period, and operational time[22]. Alumur and Kara developed a routing model for hazmats which have to be transported from their generation point to a recycle point and then a disposal point. The model also discusses the optimal location of recycling points to reduce the risk to the population[23]. Jacobs and

Warmerdam developed mathematical routing model for multiple hazmat origin points to a single destination which minimizes cost and risk of transport[24].

Zhang et al used GIS in their routing approach. They computed risks (determined by population exposed) due to air borne hazmats using dispersions models with probability of accidents. The developed model was combined with GIS to estimate the risk of release at any point and calculate the population exposed and risk corresponding to each link[25]. Boulmakoul also used GIS and the telecommunications system to capture civil infrastructure, estimate risk, and evaluate routes using a K-best fuzzy shortest paths algorithm [26].

Sherali et al developed a model considering the situation, where a hazmat carrier undergoes an accident, response units might not be available, and rerouting of the hazmat vehicle to a safe direction has to be done to avoid the risk to the vehicles in the same route. The mathematical model addresses the issue of optimal allocation of response units and also the optimal diversion route for the hazmat carrier[27]. Beroggi mentioned the need for real time routing of hazmats with the help of the latest technology to accommodate unforeseen incidents [28].

Most of the works concentrated on developing routes safer for society completely based on the existing consequence data. One of the recently evolved concerns for hazmat transportation is the risk it faces from terrorism. Hazmat carriers can be easily converted into weapons targeting cities and high impact locations. Huang et al developed a strategy to compute hazmat carrier paths with possibility of hijack [29]. Murray-Tuite et al devised a methodology for estimating the probable path of hijackers based on the consequence values used for calculating the safe route of hazmats[30]. All of the above works except the last one dealt with finding a safer route for transporting hazmat. None addressed the combined issues of possible hijacking, detecting a hijack and subsequent actions taken by the hijacker. The present work provides a methodology that addresses the issue of hijack detection and prediction of possible paths taken the hijacker by considering perceived consequence rather than actual consequence. Depending on the kind of objective function obtained various algorithms are applied depending on their advantages.

Dijkstra's algorithm is one of the commonly used algorithms for route determination. However, Dijkstra's algorithm though has disadvantages such as inability to handle negative cycles and fractional terms. Linear integer programming methods, on the other hand can be used for route estimation, but if large numbers of integer variables are involved, solution of the equation becomes tedious and practically infeasible. The above methods cannot handle fractional terms which are likely to appear in a multi objective function. However Dinkelbach's Algorithm is a fractional programming solving algorithm which can be used to solve problems of this nature [21]. Dinkelbach's algorithm follows an iterative approach using linear equations until it reaches the solution. Similarly Dinkelbach's algorithm would be cumbersome to implement and computationally intensive if applied in a larger network. In the present work, a new methodology is developed to solve the objective function obtained in the present work.

## *2.3 Pattern Classification*

Initial detection of diversion in the hazmat truck's path is crucial for identifying the hijacking event and responding appropriately. Identification can be implied as classifying the hazmat carrier's present position on the transportation network into either a safe or unsafe category**.** The classification thus would be based on the data of the truck's present location, distance from the destination and distance from the origin.

Classification of data is an important aspect of data mining and is performed using various types of algorithms. Classification can be divided into supervised and unsupervised categories. In supervised classification, data is sorted into known groups given by the user, and in unsupervised classification, data is autonomously grouped depending on the relationship between the given inputs and arranged as groups as kind of data . Both types of classification use either a distance based (nearest neighbor) methodology or density based methodology[31], which are the two most important non parametric methods for classification. Distance based classification, as the name suggests, is based on the metric of distance between the data. The distance measurement between the data points is measured using various methods like Euclidean distance, Manhattan distance, and Mahalanobis distance[32]. In density based clustering, the clustering of data points is based on pre defined distance criteria leading to the collection of points with common traits. The clustering leads to irregularly shaped denser areas of data which are separated by sparse areas. A few techniques used for density based clustering are DBSCAN, GDBSCAN, and OPTICS[33, 34]. Many algorithms are available for distance and density based clustering with each method having its own advantages and disadvantages. Hence, algorithms should be selected based on the factors like data format, computing power available and ease of implementation.

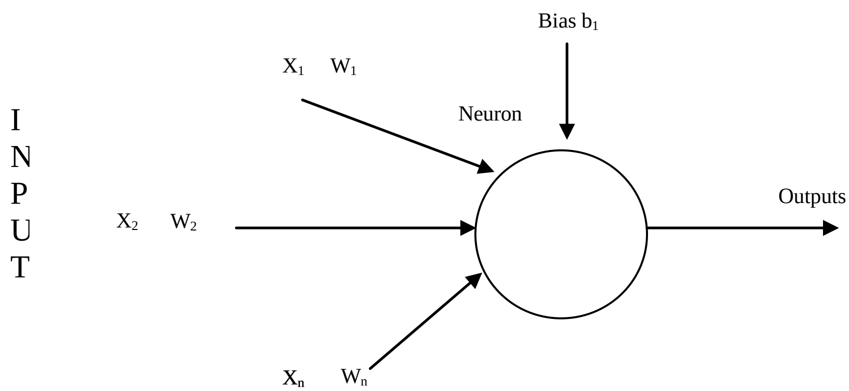
Data which has to be classified can be divided into two categories (in the present context) sequential and non sequential data. Sequential data can be defined as data which is characterized by the arithmetic value. Non sequential data, on the other hand, might not exactly depend on the arithmetic value and might have other components attached to it. For example, consider classification of buildings depending on the height in every city. A skyscraper in New York City might not be an outlier in that geography but the same height building can be classified as an outlier in Washington DC. Thus, in the present example, buildings are classified based on the parameter of height which is a non spatial component but then is also determined by the spatial component of location[32].

The present data contains both the spatial and non spatial component with spatial components being origin, destination, targets, and deviation from the actual route and the non spatial components being the threat level groups. In the present research, density based classification (neural networks) is used to identify and classify any deviant behavior of the hazmat truck. Probable Neural Networks are one of the density based classifiers and do not require any knowledge or prior distribution of the data for classification unlike various statistical methods[35].

Neural networks are easy to implement due to the availability of established and efficient software packages (e.g.MATLAB),and produce results with similar accuracy to other density based clustering methods[36].

## *2.4 Background of Neural Networks*

Artificial neural networks (ANN) can be defined as a group of artificial neurons emulating biological neural networks [37]. The basic functional unit of a neural network is the neuron which performs all the computation. A neuron in a neural network receives many input signals in vector form either directly from the input data or it can receive input from other neurons from other intermediate layers. The representation of a neuron can be seen in Figure 1.




$X_1$, $X_2$...$X_n$ are inputs to the neuron W1, $W_2$…Wn are weights given to each link and $b_1$ is the bias

**Figure 1. Basic component of neural networks**

Each neuron has an activation function which assesses the output depending on the inputs, weights, and bias given to the neuron. In simple notation, the neuron function can be defined as in equation (1).

$$\textbf{Output} = f\left(\sum_{i=0}^{N} W_i X_i\right) + b_n \qquad \textbf{(1)}$$

where

$W_i$ = weight of the $i^{th}$ connection to the neuron.

$X_i$ = input either from another neuron or from the direct input.

$f$ = function employed in the neuron

$b_n$ = bias added to the neuron output [35].

The standard functions used in the neuron are the sigmoid functions which are the tansig, logsig and purelin functions. A Neural network can be summarized as a combination of various neurons interconnected in different layers with weights and bias assigned to each connection as shown in Figure 2

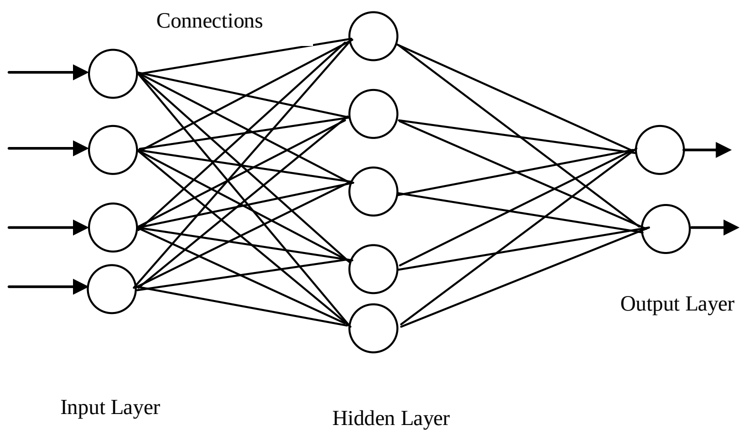



**Figure 2.Typical ANN model with input and output layer**

The architecture of a neural network can be broadly divided into three layers. The first type is called the input layer; the second is the output layer; and all the layers in between the input and output layers are called hidden layers. The number of neurons in each layer may be different and can be varied as per the required accuracy for prediction of results. Each neuron is connected to every neuron in the next layer and the connection is assigned with a weight and a bias. The input is fed into the neurons of the first layer (input layer) of the neural network. The output from each neuron is sent to the neurons in the next layer multiplied by the corresponding connection weights and added with the bias. The input propagates throughout the network in a similar way until it reaches the output layer. The difference between the calculated and observed output is used to reassign the weights which are initially assigned randomly. The algorithm computes the new weights in the network based on the difference between predicted and actual values and then assigns new values based on mean square error. The whole procedure is called training and repeated several times where the neural network model develops a general function to fit the input and output. Training a neural network can be summarized as a process for obtaining the weights of the connections to develop a satisfactory relation between input and output.

Neural network models encounter situations where the trained model fits very well with the training data set, but when data outside the training set is provided as input for prediction, the results are erroneous to an unaccepted level. This is called over fitting and is one of the common problems of neural networks. Over fitting behavior can only be observed when the trained model is tested with new data, and if over fitting behavior is develops, the neural network model will be discarded. The MATLAB 2006 neural network tool box allows the user to monitor the training of network at every instant using an additional set of data called validation data, so that the training can be stopped at a point where the model starts to over fit the data[38].

Once the training procedure is completed, the trained neural network is saved with the obtained weights. The given network is then used for predicting the outputs for new inputs. In this case, the neural network model uses the weights obtained after training. Training the network is generally a time consuming process since finding the correct combination of the number of neurons, training function, and layers in the neural network model is generally a trial and error process. Once the training is complete with satisfactory accuracy with the given data, the prediction for new data is fast.

## *2.5 Probabilistic Neural Networks*

A neural network as described above has the ability to classify data based on the historical data and their pattern examples. The basic principle behind decision making or classification of patterns is reducing the risk of error in classification. This is done by generating density distributions among the points so that they can be used for classifying the new points. Methods like back propagation and heuristics learn statistics from historical data in an incremental way over many iterations. This leads to slow learning in the case of huge data sets and complex function as they require higher number of iterations. The probabilistic neural network was introduced by Specht as an improvement over back propagations to decrease training time[39]. Insertion of different sigmoid functions has helped to improve the speed of the training and also increased the ability to classify accurately. Two different types of neural networks have been developed due to insertion of new activation function. They are probabilistic neural networks (PNN) and general regression neural networks (GRNN). While PNN is used for classification of categorical data, GRNN is used for classifying data into continuous groups [39-41].In the present work, road nodes are classified into distinct groups indicating the threat level depending on the geographic location using probabilistic neural networks.

The architecture of PNN is shown in Figure 3, and is different from other multilayered neuron structures[42]. It contains four layers: input layer, pattern layer, summation layer and output layer. The mathematical function that is employed in the neurons of probabilistic neural networks is called the radial basis function. The function translates the similarity between data to be classified and the existing data group in terms of mathematical output between 0 and 1, where 1 indicates that the given data is very similar to the existing data group and 0 indicates that they are completely unrelated.

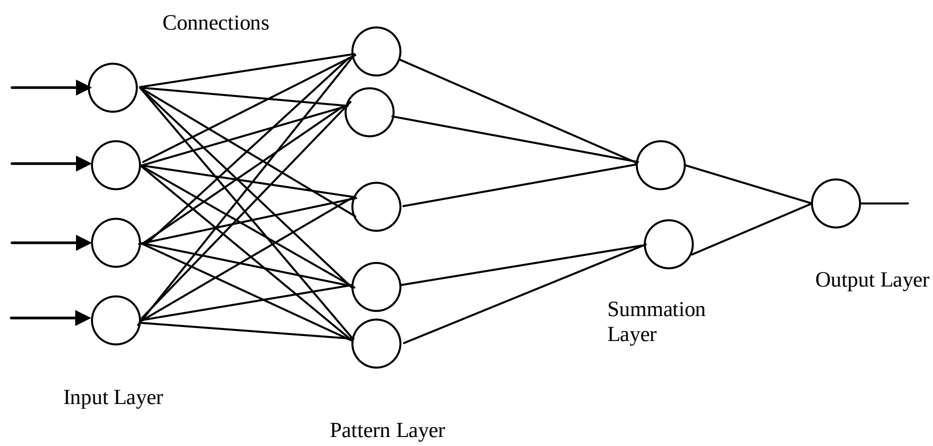



**Figure 3. Typical PNN network with input and output layer**

The output of the radial basis function (RBF) is obtained from equation 2

$$a = radbas(\| w - p \| b) \quad \textbf{(2)}$$

Where

$w$ = weight vector for the PNN model

$p$ = input vector

$b$ = bias

Equation (2) can be defined as the vector distance between weights and input multiplied by bias $b$. The *radbas* function is defined in equation 3

$$\textbf{\textit{radbas}}\ \textbf{(n)} = e^{-n^2} \quad \textbf{(3)}$$

Where $n$ is any integer

Generally, the chosen distribution of a radbas function is the Gaussian distribution which is as shown in Figure 4.

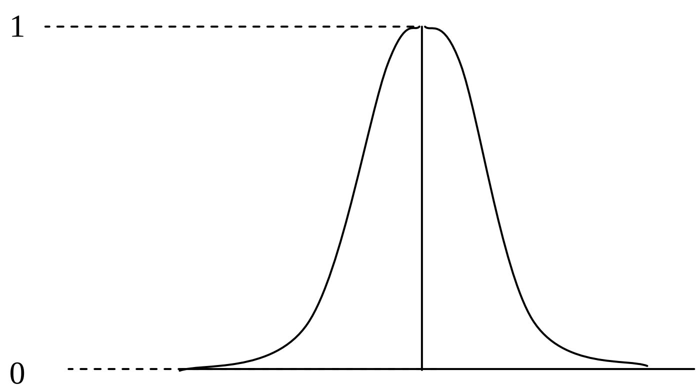



**Figure 4. Gaussian distribution function**

The computation in PNN is similar to the general neural network procedure as described in the previous section, with a small functional difference in the final layer. In the final layer, the outputs from each neuron are added, and a value of 1 is assigned to the category which has the maximum value of all the sum of outputs and other categories is assigned a value of 0.

As the probabilistic neural networks employ the Gaussian function for determining the probability distribution function of the data, the smoothing parameter for

Gaussian curve is important for classification accuracy. SPREAD is the parameter used in MATLAB as the required smoothing parameter for PNN[39]. Choosing the right value of SPREAD for the network will improve the training process and increase the accuracy of classification. The value of SPREAD is chosen by performing various iterations using different values and calculating the error percentage for each corresponding value. The neural network corresponding to the least error is chosen for predicting the new data.

Probabilistic neural networks have been employed in various disciplines because of their high accuracy in learning and classifying data. A few examples from various disciplines follow. Sheu and Ritchie used a PNN based algorithm for classifying freeway incidents based on parameters like queue lengths, duration, and time varying lane changing fractions [18]. Tarek et al developed an algorithm by fusion of neural network and fuzzy logic for identifying whether accidents occurred due to road factors or non road factors, thereby helping authorities to invest funds at the place where accidents occur due to road related factors [43]. Zhang and Luo used two PNN networks in hierarchical order to identify and classify traffic signs. The work addressed the issue of computer vision where the image (traffic sign) from the camera is sent to the first PNN for identifying the background of the traffic sign, and then the image is further sent to the second network to exactly classify the sign as an acceptable or prohibitive sign. The advantage of PNN over back propagation in this case is that it can be trained using less data, but provides higher accuracy if uniformity is found [44]. Marwala used the PNN to classify the faults in structures using a new input called pseudo modal energy with an accuracy of 90.6%[45]. PNN has even been used in bio informatics for classifying the chromosomes with a good accuracy[46]. Jin et al used an advanced PNN called constructive probabilistic neural network (CPNN) for incident detection in a freeway. The model has been implemented for incident detection in Singapore and I-880 in California with satisfactory results indicating a better performance of CPNN over PNN [47]. Hardin et al used the various neural networks(back propagation and PNN) and other density based algorithms with a accuracy over 90% in most cases[36]. These works are only a small sample of the works and fields where PNN is applied.

Examination of the literature reveals that PNN has been extensively used and is a very reliable tool for classifying patterns. There has been much work done in the field of diversion detection using neural networks though other methods were used for outlier detection [32]. In the present work PNN is used to identify the pattern of the hazmat carrier and detect diversion to give an alarm in the case of malicious diversions.

# CHAPTER 3: Problem statement

The present work can be divided in to two topics. The first is identifying the diversion in the normal route of the hazmat vehicle on the way towards destination. The second part is finding the most probable routes taken to the target. It is assumed that a malicious entity will hijack the vehicle on its normal path and then divert the vehicle towards a possible target. The malicious entity might travel on the normal route and divert towards the target at the last minute or divert immediately at the point of hijack. Earlier detection of the diversion allows law enforcement agencies to act faster to prevent /minimize the consequences.

In the road network $G$ ($N$, $A$) consisting of a set of nodes $N$ and a set of directed arcs $A$, each arc $a \in A$ has a perceivable consequence $C_a$ and distance $d_{a.}$. Initially the hazmat truck starts from its origin node $O$ and proceeds towards its destination $De$ driven by the appointed driver. A particular path $Q$ (normal route) is selected to reach its destination, using state and federal guidelines. The truck's path is allowed to change from the pre-specified routes only due to the occurrence of an incident, termed disturbance $\delta$. In other cases, the hazmat truck driver should not change paths from the highways to the connecting routes to escape from the hassle of congestion and then rejoin the original path $Q$. In addition to these benign cases where the hazmat truck driver moves away from his original path with the intention of reducing his travel time, a malicious reason for path diversion also exists. In the second case, the hazmat truck is hijacked and diverts from path $\underline{Q}$ and away from destination $De$.

Once the malicious entity successfully hijacks the hazmat carrier from its normal path, the hijacker heads towards a selected target with the intention of causing the maximum damage possible. It is assumed that the hijacker tends to choose the paths that give him a high probability of success in reaching his targets. The paths might also include secondary locations in case the hijacker has to change his plans at the last moment. Prediction of the hijacker's set of possible paths $P$ is the key to identifying the critical area where most of the routes overlap.

The diversion point $O'$ is the origin of the hijacker route and $T_n$ is $n^{th}$ target towards which malicious entity is headed towards. The path is obtained by maximizing the perceivable consequence $C_a$ and minimizing travel time $t_a$. Different sets of path $P$ are found by varying the weight λ given to each term. The path of the malicious entity would be thus a function of possible consequence and travel time, and thus tradeoff between the factors gives rise to various paths in set P. Table 1 summarizes the notation used in the entire thesis.

**Table 1. Summary of the notations used for the formulations**

| Notation adopted | Implication |
|---|---|
| Sets and Indices | |
| *G(N,A)* | Network consisting of set N of nodes and set A of directed arcs |
| *A* | Set of arcs which are indexed by *a* |
| *N* | Set of nodes, indexed by *j* |
| $O'$ | Origin node(hijack point ) of the malicious entity |
| *De* | Original destination node of the hazmat truck |
| *Q* | Path selected by hazmat truck complying to the enforced rules |
| $T_n$ | $n^{\mathrm{th}}$ target in the given highway network |

| | |
|---|---|
| $\delta$. | Disturbance generated due to congestion on a highway network |
| $H$ | Accessibility distance for a link |
| $P$ | Set of paths for the malicious entity for reaching a particular target |
| Parameters | |
| $C_a$ | Consequence generated due to a hazmat incident on link *a* |
| $d_a$ | Length of link *a* |
| $T_a$ | Travel time on link *a* |
| $w_i$ | Weight vector used in neural networks for $i^{th}$ neuron |
| $b_i$ | Bias constant that will be used in neural networks $i^{th}$ neuron |
| $\lambda$ | Weight associated with conditional expected consequence |
| $s_{min}$ | Minimum distance for the malicious entity's path |
| $s_{max}$ | Maximum distance for the malicious entity's path |
| $v_{min}$ | Minimum conditional expected consequence on malicious entity's path |
| $v_{max}$ | Maximum conditional expected consequence on malicious entity's path |
| Variables | |
| $x_a$ | Binary integer decision variable taking the value 1 if the link is chosen by the malicious entity and 0 otherwise |
| $z$ | Value of objective function for malicious entity's path selection |

# CHAPTER 4: Methodology

This chapter details the methodology used for diversion detection and estimating the probable paths. Section 4.1 describes the development of classification features required for training the neural network. Section 4.2 describes the new approach taken for estimating the consequence perceived by the malicious entity. Section 4.3 describes the assumption and scenario for path selection once a hijacking is successful. Section 4.4 gives the mathematical formulation for the malicious entity's path selection. Section 4.5 details the new algorithm developed to solve the mathematical formulation developed in section 4.4.

## *4.1 Methodology for neural network model estimation*

A hazmat truck is travelling on a pre-specified path starting from a known origin going to a known destination. The pre-specified path is determined by various state and federal regulations combined with the kind of hazmat transported and location of the destination. According to present regulations, a driver is allowed to make a diversion for food, rest, and of incidents[9]. It is anticipated that after the driver deviates from his path, he will return to his original path to complete the trip. To prevent any kind of danger to the society, it is important for security agencies to ensure that the hazmat driver does return to his path with a reasonable time or stop the vehicle from moving ahead on its present diverted path.

The location of the hazmat truck at any instant of time can be mathematically formulated using a polar co-ordinate system. But in the present work, hazmat carrier location is identified only at discrete locations rather than continuous locations; the discrete locations are nodes in the present transportation network. The state of the vehicle can be described as ($r_j$, $\theta_j$) where $r_j$ indicates the travel time the hazmat carrier takes to reach to that particular node $j$ from the origin. The diversion of the hazmat carrier (at some node) is measured as the angle $\theta_j$ between the straight line joining the origin and the present node and the straight line joining the origin and the destination and defined as $\theta_j$. It should be noted that the shortest path (travel time) between the origin and destination might not be the standard path that is allowed for hazmat carriers. Intuitively it can be understood that, hazmat carriers tend to choose prescribed by federal and state laws) which are less congested. As the procedure takes travel time into consideration for determining the diversion, geographical location of routes does not affect the diversion angle.

Each node in the transportation network is assigned a threat level. The threat level in the present work relates to the extent of hazmat carrier diversion. Three threat levels are defined for estimating the intent of diversion of the hazmat carrier. First is the safe level which indicates that the vehicle is either on the normal route or on routes that are taken when an incident occurs on the normal route. Second is the suspicious level which indicates that something is suspicious about the vehicle since the present node is generally not taken to reach destination. Third level is the dangerous level confirms that hazmat carrier is either hijacked or lost completely.

The procedure for defining the above described scenario as a neural network is as follows:

a. Identify the rules required for the neural network to determine the threat levels.
b. Develop a mathematical model to convert geographic space to a vector space equivalent to be understood by neural networks.
c. Train the neural network model with the data generated from step b.
d. Test the neural network with transportation network data outside the training set and measure the percentage error in prediction.
e. Choose the model with lowest percentage of error.

The neural network model classifies the nodes of the road network into groups. As the path of the hazmat carrier can be implied as a collection of nodes, when information of each node reached is added to the algorithm, the status of each node is obtained. The following steps define the procedure used for classifying the nodes of the transportation network in detail.

#### Step 1: Defining the diversion

Two kinds of diversion can be defined for the hazmat carrier. The first is if the vehicle is moving to a place which is neither logically expected nor recommended to reach its pre specified destination. The other case of deviation is when the hazmat carrier enters in, to populated areas right beside the standard path which is not recommended. In both cases, identifying the hazmat driver's intentions is key to preventing any possible negative consequence.

The geographical location of the truck, obtained through GPS technology, is the key data which is used for identifying the hazmat carrier threat level [48]. As mentioned previously, the location of vehicle in the present work is evaluated only at the nodes of the transportation network.

Apart from identifying the proximity to normal route, other safer routes are identified by simulation of transportation network with incidents. Various links on the transportation network are simulated for incidents and corresponding re routing of the traffic along new nodes is identified and termed as safe routes. Thus the data set of the safe routes is built. Dangerous nodes or areas are located far away from the safe routes while safer nodes are geographically located very near the safe routes which might be taken in the case of any incidents or due to human error. All the nodes other than safe and dangerous are termed as suspicious. A few outliers can be introduced at the boundaries of suspicious and dangerous threat level nodes to incorporate some possible error rate.

#### Step 2: Defining the parameters for classification

Intuitively, for a given map with a pre set origin and destination, certain areas can be classified as dangerous, suspicious, or safe based on the travel time to reach a particular node. The division of the nodes only based on the travel time is shown in Appendix A.

To train a neural network, the patterns should be converted to “identifiable” parameters. For example, classifying the nodes using node numbers or latitude and longitude positions will not be suitable as they do not follow a pattern.

In the present work a polar coordinate system is used to represent the location of nodes $(r_j,\theta_j)$ where $r$ is the travel time required to reach that particular node from the origin and $\theta_j$ is the angle between the straight line joining the origin and the present node $j$ and the line joining the origin and destination. The origin of the co ordinate system for this purpose coincides with origin of the hazmat truck. Equation 4 gives the formula for measuring angle and Figure 5 gives an example of the distance.

$$\cos\theta_j = \frac{a_j^2 + b^2 - c_j^2}{2a_j b_j} \qquad \textbf{(4)}$$

Where

$\theta$ is the angle between the origin and the node j

$a_j$ = travel time between the origin and node j

b = travel time between the origin and destination which in present case is the shortest distance between both

$c_j$ = travel time between the destination and node j

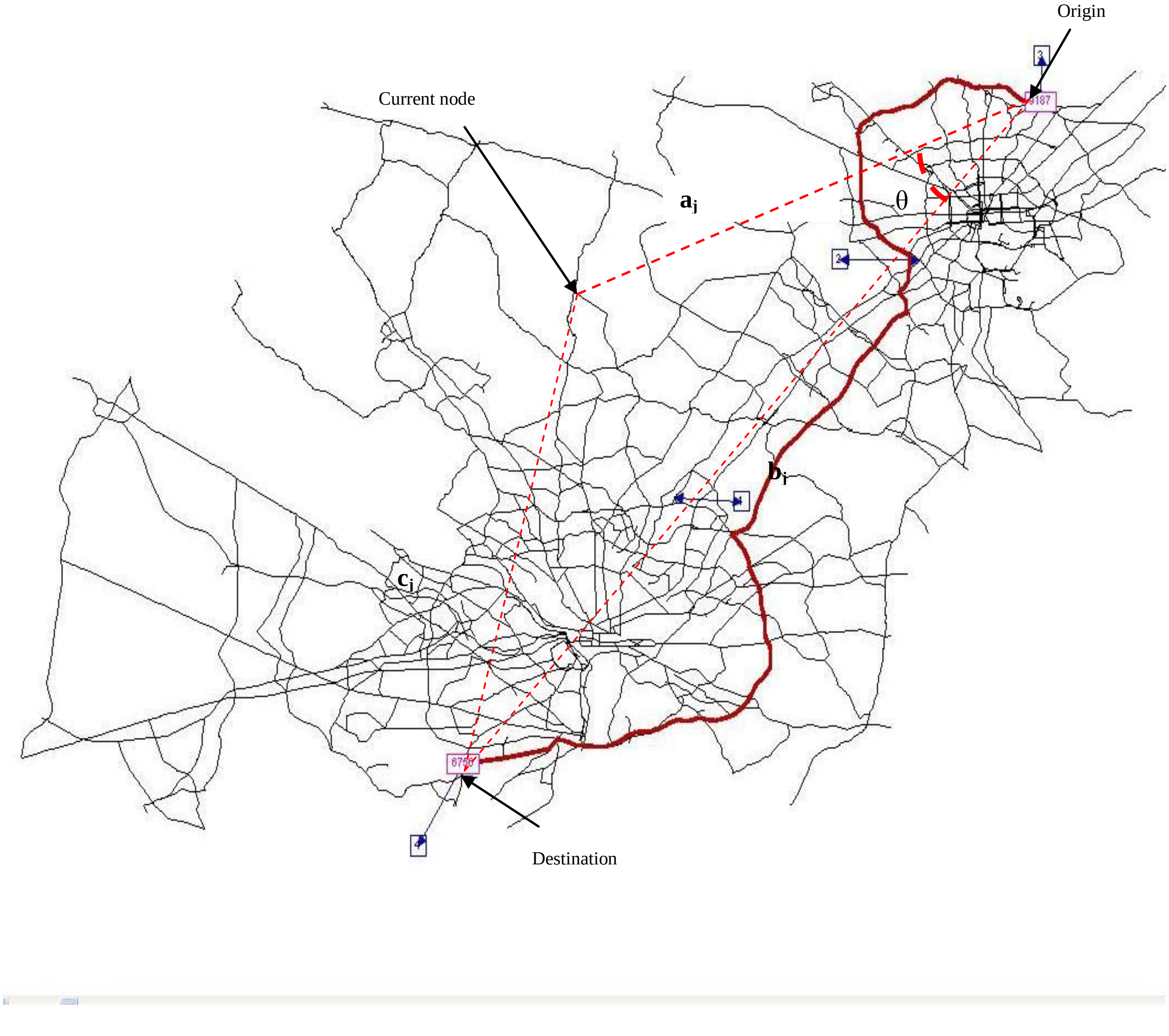



**Figure 5. Example showing distances described in the above equation**

The red line, which is the shortest path in terms of travel time between the destination and origin, is equal to value '$b_j$' and is equivalent to the dashed line joining the destination and origin in Figure 5 . The path '$a_j$' is the shortest path (shortest travel time) between origin and destination and '$c_j$' is the shortest travel time between the present node and the destination. The equivalent triangle formed by the travel times between the three vertices i.e origin, destination and current node is show in Figure 5. The angle θ is the angle between the line joining current node and origin to destination.

From the above description, one can intuitively observe that the value of the angle for nodes on the shortest travel time path will be zero, independent of the distance from the origin. Similarly, the nodes adjacent to the shortest travel time path will have small deviation angles and increase as the hazmat carrier (nodes) moves away from the shortest travel time path. All the paths are obtained by minimizing the travel time between the given origin and destination.

**Step 3: MATLAB programming**

MATLAB is used to implement the probabilistic neural networks. The software provides a neural network tool box and for the present work, the function newpnn is used. The data should be in the form of rows and columns, where the number of rows indicate the number of inputs (in the present case $r_j$ and $\theta_j$ )and the number of columns indicates the total number of records (nodes).

The classification of the nodes in the transportation network with PNN using the deviation angle and travel time $r_j$. For the output data in a similar manner, rows indicate the total number of records. In the present case, the output matrix has three columns, corresponding to three threat levels possible for each transportation node. The array of is filled with 1's and 0's . The group to which hazmat carrier belongs (node location) is denoted using 1 and the other two threat levels are filled with 0.The data is converted in to vector form to fit the requirement for the newpnn function.

**Step 4: Training and simulating the neural network**

The data selected for training the neural network should represent the characteristics of the entire set. Hence, a random set of data is taken from the total data as the training set while the remaining data is taken as the validation set. Several values of

SPREAD are examined. Each time a SPREAD value is chosen and network is trained, and trained neural network (PNN) is then tested with the validated set to measure the error in classification. This process is repeated until a minimum error neural network is obtained which is then used for predicting real world situations.

The path diversion identification methodology can be summarized in the form of a flow chart as shown in Figure 6. The location of the vehicle is sent to the neural network model which decides the threat level If the threat level is suspicious or dangerous the algorithm searches for exceptions If exceptions are found the vehicle is either classified into the “danger” threat level or “suspicious” threat level. In case the vehicle belongs to “danger” threat level an alarm is raised for the authorities. If the threat level is “suspicious” the algorithm checks for either the next threat level from the vehicle or alerts human correspondent. When the threat level is “safe”, the algorithm waits for acquiring the next location of hazmat carrier.

**Limitations**

1. The present methodology is limited to a single origin and destination. The methodology developed for evaluating a position of a hazmat carrier is dependent on the origin and destination, and this is used as the basis for classifying possible threat in to different categories. Change in either the destination or origin would change the parameters assigned to the nodes and thereby the threat levels.
2. In the case of multiple destinations and a single origin, the present model should have multiple training sets to suit the requirement. The improvement and required changes for this situation are discussed in the future work section in chapter 6.

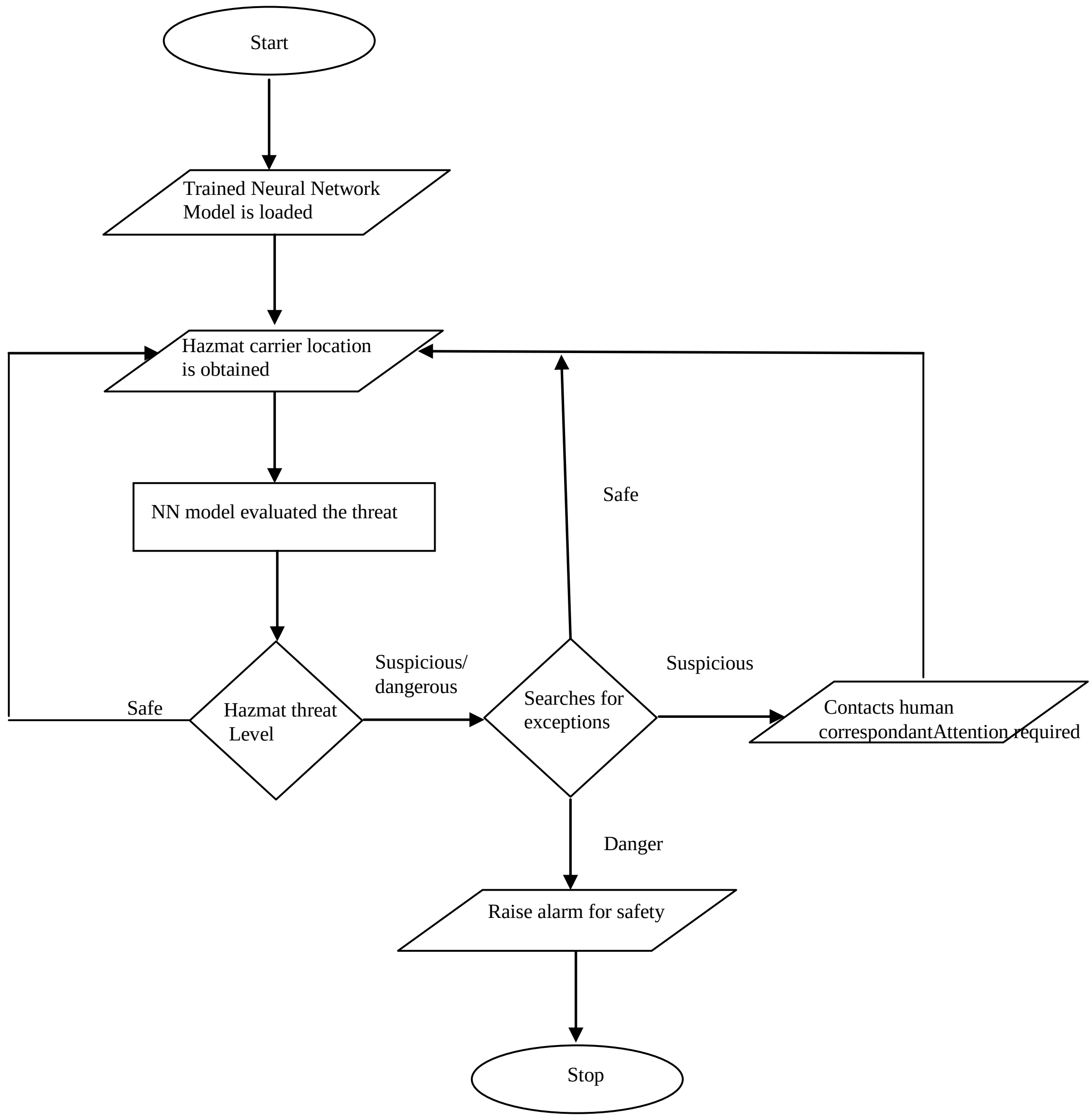



**Figure 6. Flow chart depicting the methodology to be used in neural networks.**

## *4.2 Methodology for consequence calculation*

In a given road network, the hijacker would likely choose a destination that he perceives as having a high consequence level. The route chosen lies between the shortest path to the destination and the one the hijacker perceives to have the highest consequence. The decision regarding a particular route is more dependent on the perception of the area rather than the actual statistics of the area. For example, the Washington, DC area might be perceived to have more population during the day due to working people travelling to the city and tourists. Thus, a hijacker might perceive this to be a more highly populated place than the actual data statistics given by the US Census Bureau. On the other hand, two different areas having a population difference of around 10,000 might be perceived to have an equal population and equal possible consequence in case a hazmat truck is used as a weapon to cause destruction. Therefore, choosing a particular route is more dependent on the hijacker perception of the consequence in that area than the actual population of the town.

The following assumptions were made for calculating the consequence value of a given transportation network.

### 4.2.1 Assumptions:

- The total population of a city is assumed to be located at a single point called the centroid. As the population of a particular place is considered for calculating the attractiveness value rather than estimating the complete consequence value due to a hazmat truck accident, the above assumption is justified. In the case of cities with very high population, concentration of such huge population at a point cannot be justified and hence the city can be divided into subsections with concentrated population centers.
- The population of every city should be adjusted based on perception. This is justified in section 4.2.

- Secondary locations, defined as a location where malicious entity would carry out his plan, once he cannot reach the main target. Secondary locations lie play a role in route selection.

### 4.2.2 Consequence estimation

The estimation of consequence values for a given network can be explained in the following four steps

**Step 1: Procuring the GIS information**

1. Build/procure the geospatial layers of the road network and the town centers.
2. Extract the longitude and latitude of each node in the highway network and population centre from the spatial maps.

**Step 2: Calculating distances between nodes and cities**

Calculate the distance between each city centroid and the nodes of the given transportation network using the Haversine formula. The Haversine formula finds the direct distance between two sets of coordinates as according to equations (5-9):

$$\text{del\_long} = \text{lon}_2 - \text{lon}_1 \tag{5}$$

$$\text{del\_lat} = \text{lat}_2 - \text{lat}_1 \tag{6}$$

$$a = \sin^2((\text{del\_lat}/2)) + \cos(lat_1) * \cos(lat_2) * \sin^2\left(\frac{del\_lon}{2}\right) \tag{7}$$

$$c = 2 \text{ arcsine } (\min(1, \sqrt{a})) \tag{8}$$

$$\text{Distance} = R\, c \tag{9}$$

where

lon2, lon1 = Longitudes of place 2 and place 1, respectively

lat2, lat1 = Latitudes of place 2 and place 1, respectively

R = radius of the earth (6731 km)

Distance = distance between the place 1 and place 2.

**Step 3: Calculating distances between links and cities**

Distance from the centre of the link to the city centroid is calculated using triangle properties. This distance will be used as perceived distance since the exact location of town centre is fuzzy. Calculate the distance using the formula for the bisector of a triangle. The three vertices of the triangle are the ends of the link in the network and the third node is the city population centroid. The bisector of a triangle is given by equation 10.

**Length of median =**

$$0.5 * \sqrt{(((2 * side\_a * side\_a) + (2 * side\_c * side\_c)) - (side\_b * side\_b))} \qquad \textbf{(10)}$$

where,

side_a = length of side a

side_b = length of side b

side _c = length of side c

**Step 4: Calculating the consequence**

An accessibility distance η, is taken for computing the final consequence for network links. For each link in the network, if a city centroid lies at a distance of η or less, add the consequence of the city to that particular link. Repeat this step for the whole network and all relevant cities to assign the total consequence to the links. Once the consequence value is estimated, predict the malicious entity's probable paths are estimated using the equations explained in the next section.

## 4.3 Prediction of probable paths taken by the malicious entity

Chapter 3 provides the notation used for predicting the probable paths using consequence and travel time. A set of assumptions simplify the mathematics involved in the formulation of the path determining equations and include.

(a) Hijack on the given transportation network will occur.

(b) Nodes are the only locations for hijacking.

(c) The hazmat truck, once hijacked moves towards its target and is not stored or hidden for any future purpose

(d) The travel time on any link is depends on the speed limit allowed on particular interstate/highway, rather than congestion.

(e) The probability of interception by the law authorities is constant as it is expected that law authorities constantly patrol the area and would be available upon an emergency call.

The steps involved in determining the most probable paths for a target $T_n$ taken by malicious entities from a hijack point are as follows:

(1) Identify the original route for the hazmat carrier considering the rules and regulations enforced by the federal and state governments.

(2) Calculate the consequence and travel time.

(3) Determine the path(s) by minimizing travel time and maximizing consequence and varying the weights given to each objective. The origin $O'$ for the malicious entity lies on the path obtained from step 1 and the destination is one of the probable targets $T_n$.

## *4.4 Mathematical formulation of hijacker path selection*

As described in previous sections, the hijacker tends to choose routes which maximize the consequence and also reduce the travel time. These two components have different units and cannot be directly added. Normalizing both the components avoids the problem.

The different paths of the malicious entities can be obtained by solving equations (11-14). Table 1 provides the complete summary of notation.

$$\min z = \lambda \frac{\dfrac{\sum_{a \in A} \rho_a C_a x_a}{\sum_{a \in A} \rho_a x_a} - v_{\min}}{v_{\max} - v_{\min}} + (1-\lambda) \frac{\sum_{a \in A} d_a x_a - s_{\min}}{s_{\max} - s_{\min}} \quad \textbf{(11)}$$

where

$x_a$ is the binary variable assigned to each link with a value 1 if the link is selected for inclusion in the hazmat route and 0 otherwise

$\rho_a$ is the probability of getting caught by the law enforcement on the link a.

$v_{max}$, $v_{min,}$ are maximum and minimum conditional expected consequence for malicious entity path between the hijack point and target, respectively.

$s_{min}$, $s_{max}$ are minimum and maximum travel times for the malicious entity path between hijack point and target.

$\lambda$ is the weight given to the consequence term and lies between 0 and 1

$d_a$ is the travel time on link *a*

$C_a$ is the consequence of the link *a*

The first term in equation 11which is multiplied by $\lambda$ is the normalized consequence term conditioned by the probability of capture, Since assumption (e) in 4.3 states that probability of interception is constant, equation 11 can be simplified to equation 12. The term multiplied by (1-$\lambda$) is the normalized travel time factor. The value of $\lambda$ indicates the percentage contribution of each factor and varies from 0.0 to 1.0.

$$\min z = \lambda \frac{\dfrac{\sum_{a\in A} C_a x_a}{\sum_{a\in A} x_a} - v_{\min}}{v_{\max} - v_{\min}} + (1-\lambda)\frac{\sum_{a\in A} d_a x_a - s_{\min}}{s_{\max} - s_{\min}} \tag{12}$$

Subject to:

$$\sum_{a\in F(j)} x_a - \sum_{a\in H(j)} x_a = \begin{cases} 1 & \text{if } j = O' \\ -1 & \text{if } j = T \\ 0 & \text{otherwise} \end{cases} \qquad \forall j \in N \tag{13}$$

$$x_a \in \{0,1\} \qquad \forall a \in A \tag{14}$$

Equation (12) is a linear combination of normalized consequence and distance values in the route and is subjected to constraints in equations 13 and 14

Equation (13) is the transportation flow equation and equation (14) restricts the decision variable to binary values.

## *4.5 Algorithm for route estimation*

The objective function defined in equation (12) contains fractional terms and standard path finding methods like Dijkstra’s algorithm cannot be used. Though various algorithms are available, most of them are computationally intensive and not preferred. A different algorithm has been developed for solving the above equations and is described below. The implemented algorithm in the present case is a modification of breadth first search[30].The algorithm has been coded in MATLAB 2006b version 7.2.0.232. The algorithm can be described in five steps:

**Step 1: Initializing**

(i) A set of variables as follows is created for storing basic network information which is as follows

a. *data (l)* matrix stores information of link *L* showing the nodes it connects.

b. *rep_indices (i)* stores the conversion table between old and new numbers node numbers which are needed for a proper data structure.

c. *dist_conseq (i, j)* stores the consequence value of a link between node *i* and node *j*. A value of infinity is assigned for nodes which are not connected.

d. *dist_time (i, j)* stores the travel time between on the link connecting node *i* and node *j*. A value of infinity is assigned to nodes that are not connected.

e. *pathS* and *pathE* variables store the origin and destination of the scenario, respectively.

(ii) Another set of variables are initialized as null matrices for storing values which will be generated during the execution of the algorithm.

a. *cap (i)* stores the cumulative probability of capture obtained until node *i*.

b. *ccc(i)* stores the cumulative expected consequence obtained until node *i*.

c. *Z(i)* is used to store the objective function values at each node for the malicious entity, and a value of infinity is initially stored in the variable.

d. Variables for maintaining lists of nodes to be visited, visited nodes,and predecessor nodes are initialized as blank matrix.

- *parent(i)* stores the parent predecessor of the node *i* i.e. the node from which the present node is connected
- *queue* stores all the nodes that are to be visited. After the initialization, nodes are added and subtracted once another node is visited.

**Step 2a: Determine Parameter Values**

For calculating the objective function value, initially the values the values $s_{max}$, $s_{min}$, and $v_{max}$, $v_{min}$, are evaluated using same algorithm which is used for path prediction but with a different optimization function. The objective function for each term is as follows

$$S_{max} : -\frac{\sum_{a \in A} C_a x_a}{\sum_{a \in A} x_a}$$

$$S_{min} : \quad \frac{\sum_{a \in A} C_a x_a}{\sum_{a \in A} x_a}$$

$$v_{max} : \ -\sum_{a \in A} d_a x_a$$

$$v_{min} : \quad \sum_{a \in A} d_a x_a$$

**Step 2b: Initial Iteration**

(i) All the nodes connected to the origin are noted in *queue* matrix and the values of cumulative distance, consequence and objective function are updated using the objective function and other variables defined.

(ii) The *z* value of each of the nodes connected to the origin are have updated and all of these nodes have the origin stored in *parent*.

**Step 3a: Traversing the network.**

Each of the nodes stored in the *queue* matrix now act as the origin node(s) and the matrices *cap(i) ccc(i)* and *z(i)* are updated. The *queue* matrix is updated with new nodes to be travelled and *parent* matrix is updated with the nodes it is connected with in each correspondingly. The number of node updated depends on the number of connections that are present to the present nodes in the *queue* matrix. *cap(i)* and *ccc(i)* by adding the probability of getting caught, consequence values of the link starting from node i to node *j* and *Z(i)*is updated using equation 12.

**Step 3b: Forming the connections**

In step 3a when the algorithm visits new nodes using the nodes stored in *queue* as origin nodes, the nodes might have already been visited and assigned a *parent* value. In order to assign a new connection, i.e. assign a new *parent* node, the algorithm compares the objective function value as shown in equation 15. The algorithm is about to form a link from node *j* which is already in the path to node *i*.

$$z(j) > -\lambda \frac{\frac{ccc(i)+c_a}{cap(i)+1} - v_{\min}}{v_{\max} - v_{\min}} + (1-\lambda)\frac{dist(i)+d_a - s_{\min}}{s_{\max} - \int s_{\min}} \qquad \textbf{(15)}$$

where *a* is the link connecting nodes *i* and *j*

(i) If the above inequality is satisfied, the values of *dist (i), ccc (i), cap (i)* and *z(i)* are updated in node *j*.

(ii) The node i is added in the list to be traversed and added to *queue*.

(iii) The node which is forming the connection is stored as the parent node, i.e. node *j* is the predecessor of node *i*. Hence *parent (i) = j*.

**Step 4: Check for Cycles**

In some cases, connections formed because of equation 15 create a cyclic path, or loop. To prevent cycles, the algorithm traverses the path from node i back toward the origin using the *parent* matrix and, if a cycle is formed, the algorithm will never reach the origin node. In the case of cycles, the algorithm removes the connection made in Step 3 and restores the previous values to ccc (j), cap (j), Z(j) and parent(i).

**Step 5: Trace the path**

Once the computation is completed, the algorithm searches for the parent of the node *pathE*. Once the parent value is obtained, it recursively searches for the parent of *parent (pathE)*. The search goes on until the algorithm encounters the node *pathS*. Thus the complete path is identified along with its corresponding cost [49].

The algorithm is coded in MATLAB by improving existing Dijkstra's algorithm by handling negative values and fractional terms [49]. The methodology for the neural network is also implemented using the neural network tool box available in MATLAB. The various functions required for calculating the distances consequence are also implemented in MATLAB and code has been provided in the appendix A of the document.

# CHAPTER 5: Application

In the present chapter, the methodologies described in chapter 4 are applied to a real world road network between Washington, DC and Baltimore. The hazmat carrier moves from northern Baltimore to a destination somewhere south of Washington DC. From a point on the normal route malicious entity hijacks the vehicle and moves towards either the International Dulles Airport or National Mall. Section 5.1 describes the road network to which the methodologies are applied. Section 5.2 details the neural network methodology for identifying the diversion in paths of the carrier. Section 5.3 identifies the possible routes taken to IAD and the National Mall. Finally, Section 5.4 gives opinions of an expert panel about the assumptions pertaining to population, accessibility distance and secondary locations.

## *5.1 Case Study: Transportation Network between Baltimore and Washington, DC*

The network chosen for implementing the methodology developed for predicting the probable paths and the diversion detection model is the area between Baltimore and Washington, DC and can be observed in Figure 7. The network is composed of Interstate 95, 395, 495, and 695, Routes 1, 29, and 295. The network has a total of 4654 directed links and 1776 nodes. As described in the methodology section, the centroid of cities are overlayed on the network as shown in Figure 8.

The size of the symbol for the city centroid is proportional to the population. The data for these figures comes from the Bureau of Transportation statistics (ESRI road map) and the Census website (population)[50]. The origin of the hazmat carrier is a node in Baltimore and the destination is a node in the suburbs south of Washington, DC. Based on the regulations specified by Maryland Department of Transportation I 395, 95 and 495 are chosen as routes for reaching the destination.

**Figure 7. Network used between Baltimore and Washington, DC**

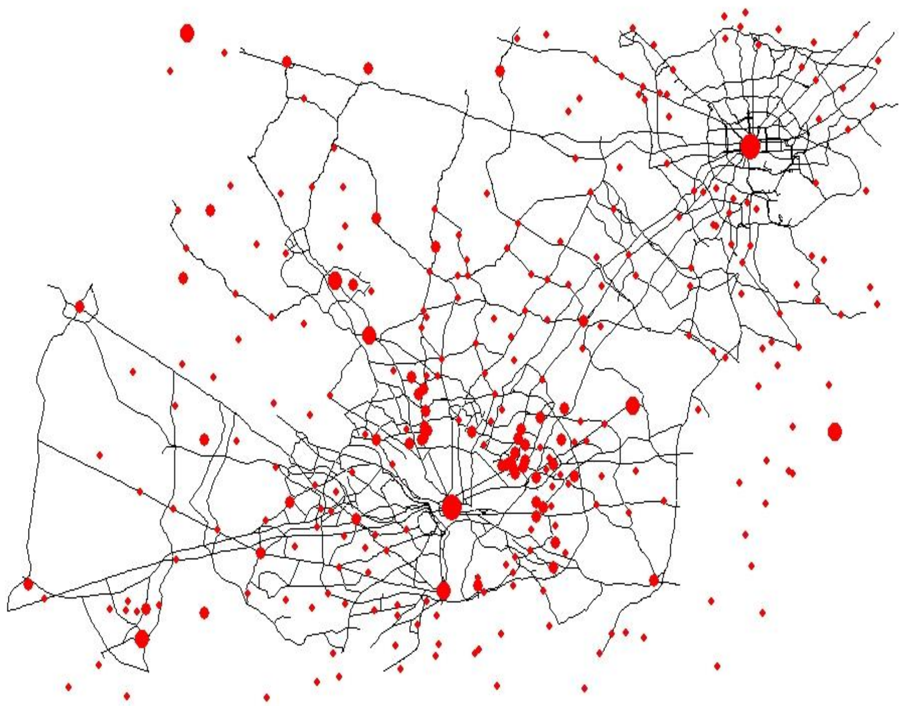


**Figure 8. Network showing the city population location used between Baltimore and Washington, DC.**

The origin and destination are shown in Figure 9. Due to the presence of the Baltimore port, there is heavy movement of hazmat from the Baltimore area to various places on the east coast. The targets considered in the present map are the International Dulles Airport (IAD) shown as "target" and the National Mall shown as "target 2" in Figure 9. The reasons for selecting them as targets are because of the damage that can be caused at those locations. The original path can be seen in Figure 10, indicated using square blocks

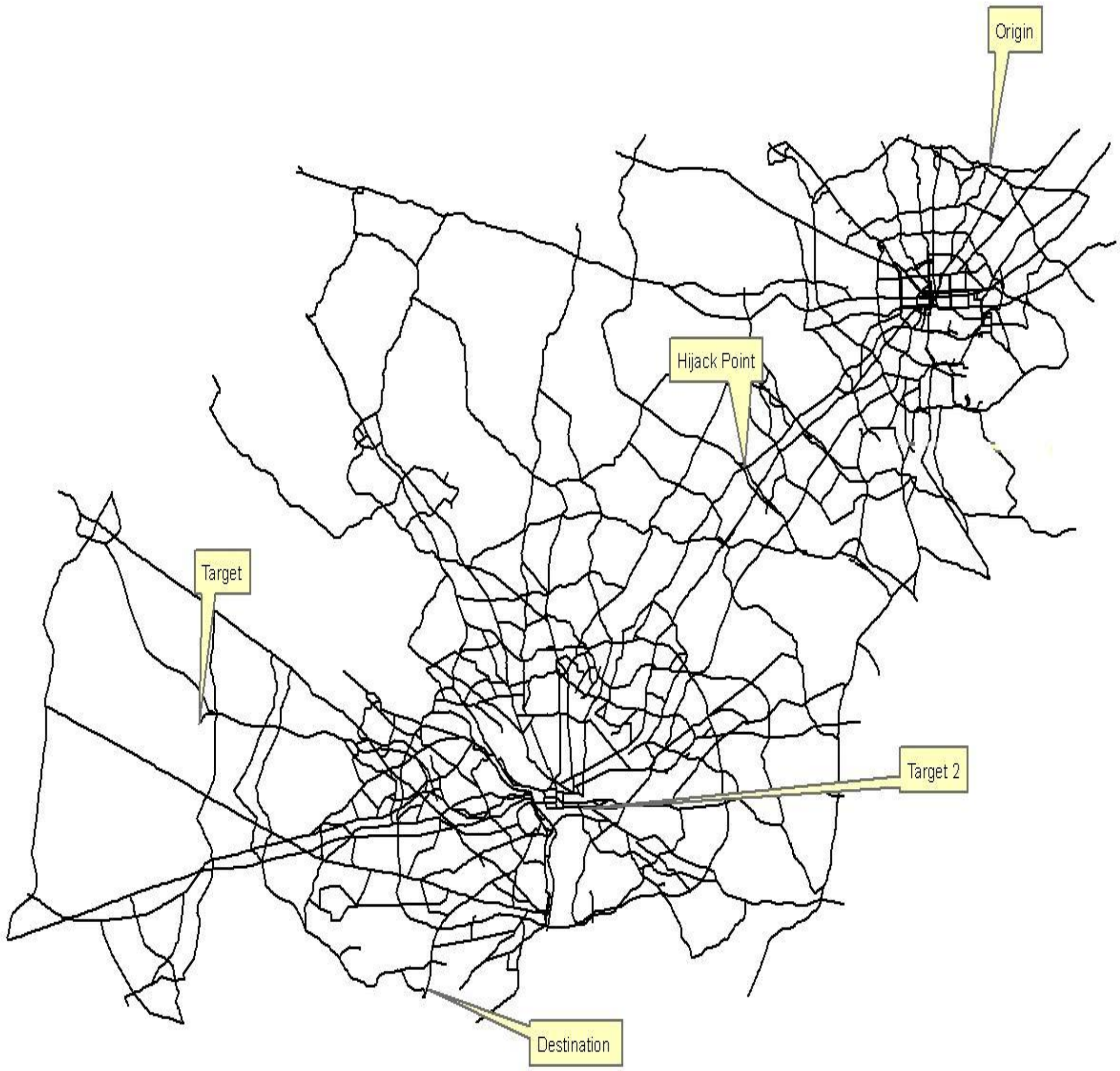



**Figure 9. Network showing the origin and destination for the hazmat truck and possible targets for malicious entities.**

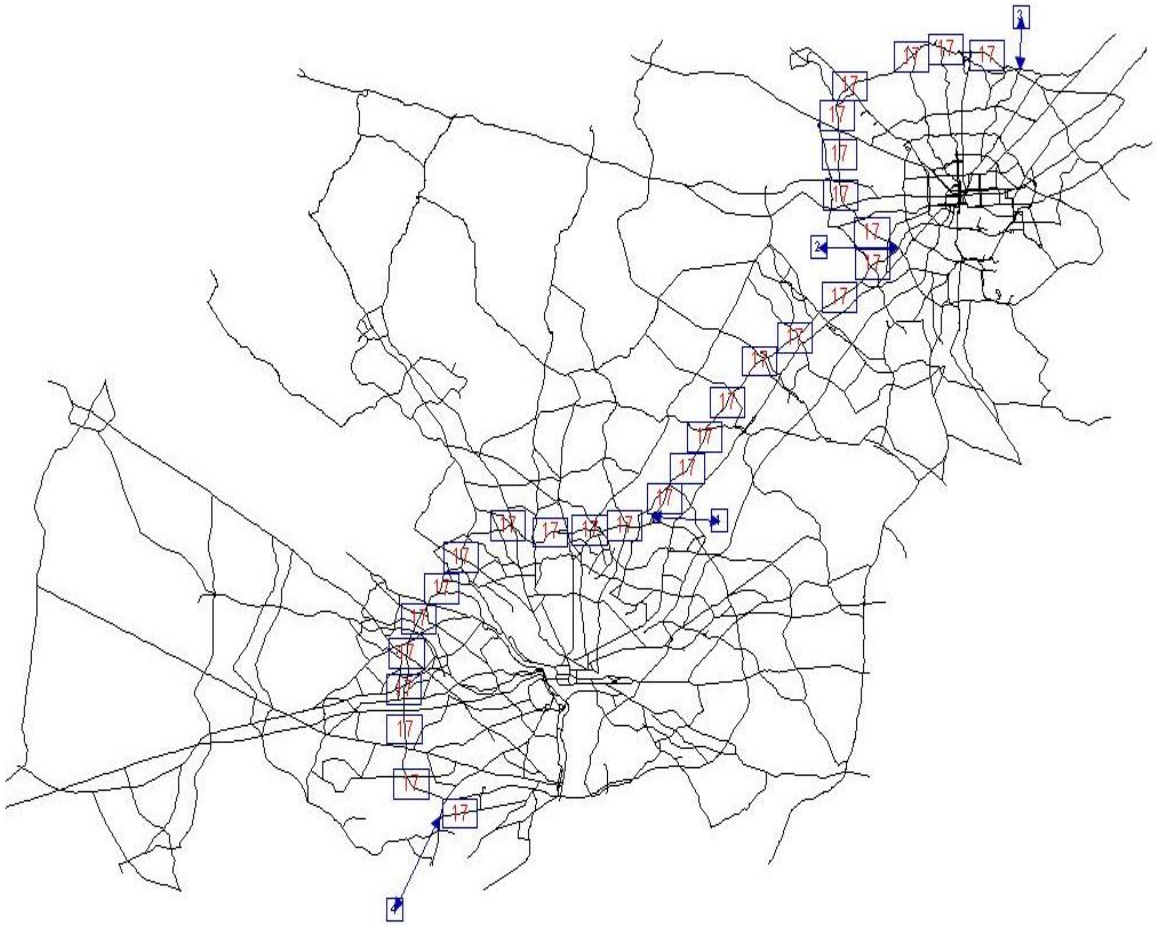


**Figure 10. Transportation network showing the pre-specified route for hazmat truck in the present scenario between Baltimore and Washington, DC**

The hijack point is chosen on Interstate 95 to see the behavior of the set paths *P*. Implementing the algorithm and methodology described in the above sections, consequence is calculated and then probable paths of malicious entities are plotted. In the described methodology, the population of an entire city/town is assumed to be concentrated at a point; this cannot be justified in the case of Washington, DC due to the large amount of population. Hence the location of Washington DC's population has been split into four centroids of approximately equal population and the maps showing the four areas of Washington DC are shown in appendix B.

The probable paths of the hijacker to the International Dulles Airport and National Mall were computed. The first target (International Dulles Airport) is located in the suburbs of Washington, DC and has a good connectivity with interstates and national highways. This target's location yields distinguishable routes due to different combinations of travel time and consequence. The second target chosen is the National Mall in Washington DC which is an urban network like most of the cities where population is dispersed and each point is accessible at lower travel speeds.

The Lambda value indicates the percentage importance given to the consequence term i.e. the degree to which the malicious entity prefers to cover secondary locations compared to travel time. For example, a value of 0.2 for lambda indicates 20% weight age to consequence and 80% to travel time, which means that the malicious entity prefers to reach his target faster rather than high consequence locations. The Figures 11-18 show the paths taken by malicious entity for various values of lambda for the first target.


**Figure 11 The path taken by the malicious entity for value of λ =0.1 to λ =0.4 towards IAD.**

The paths obtained with values of λ ranging from 0.1 to 0.4 are similar; this indicates that the travel time is the key dominating factor until this range.

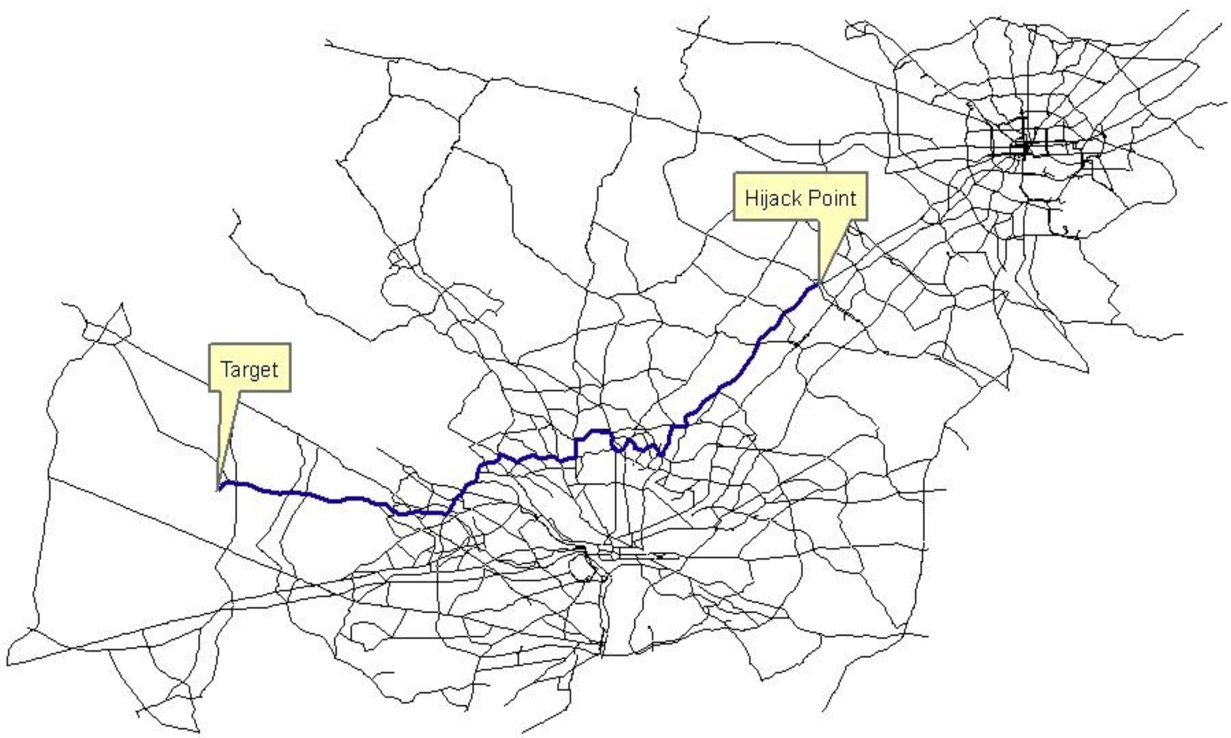



**Figure 12. The path taken by malicious entity for value of λ =0.5 towards IAD.**

A small change in path is observed at $\lambda = 0.5$ at the point where I 95 and I 495 meets. The change is not really considerable, yet the consequence term is increasing the dominance in the objective function.

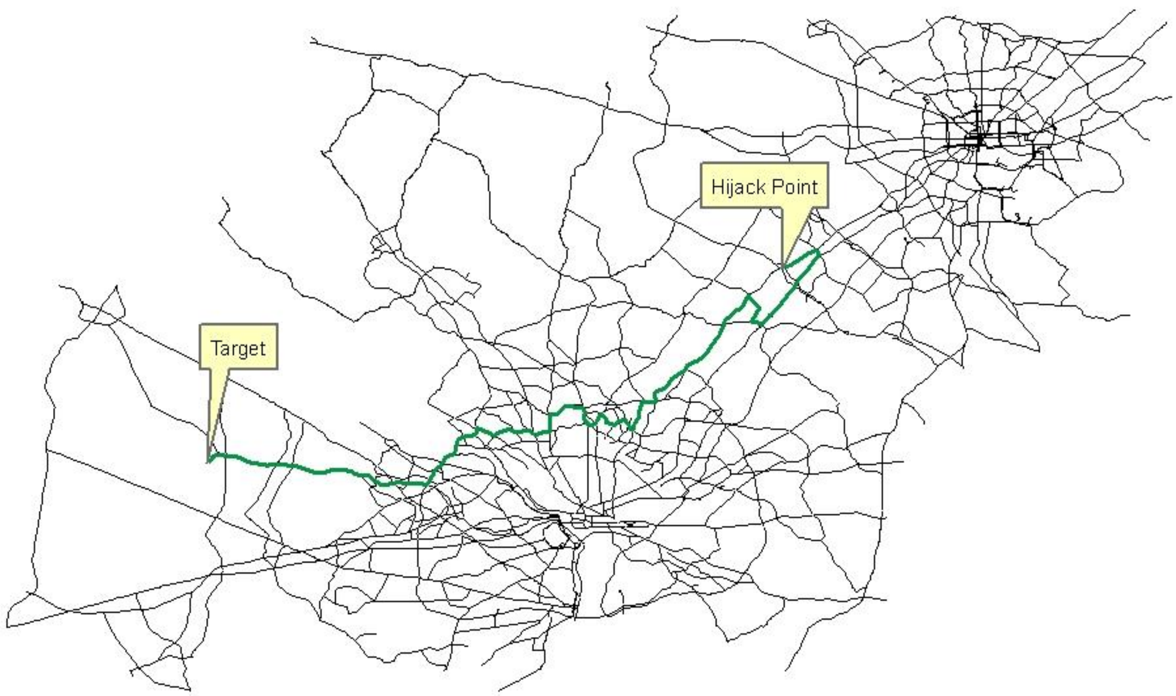



**Figure 13. The path taken by malicious entity for value of $\lambda$=0.6 towards IAD.**

At $\lambda$= 0.6 the path of the vehicle moves towards the centre of the Washington, DC. It can be observed that the consequence term is more dominant over travel time from this particular value of $\lambda$.

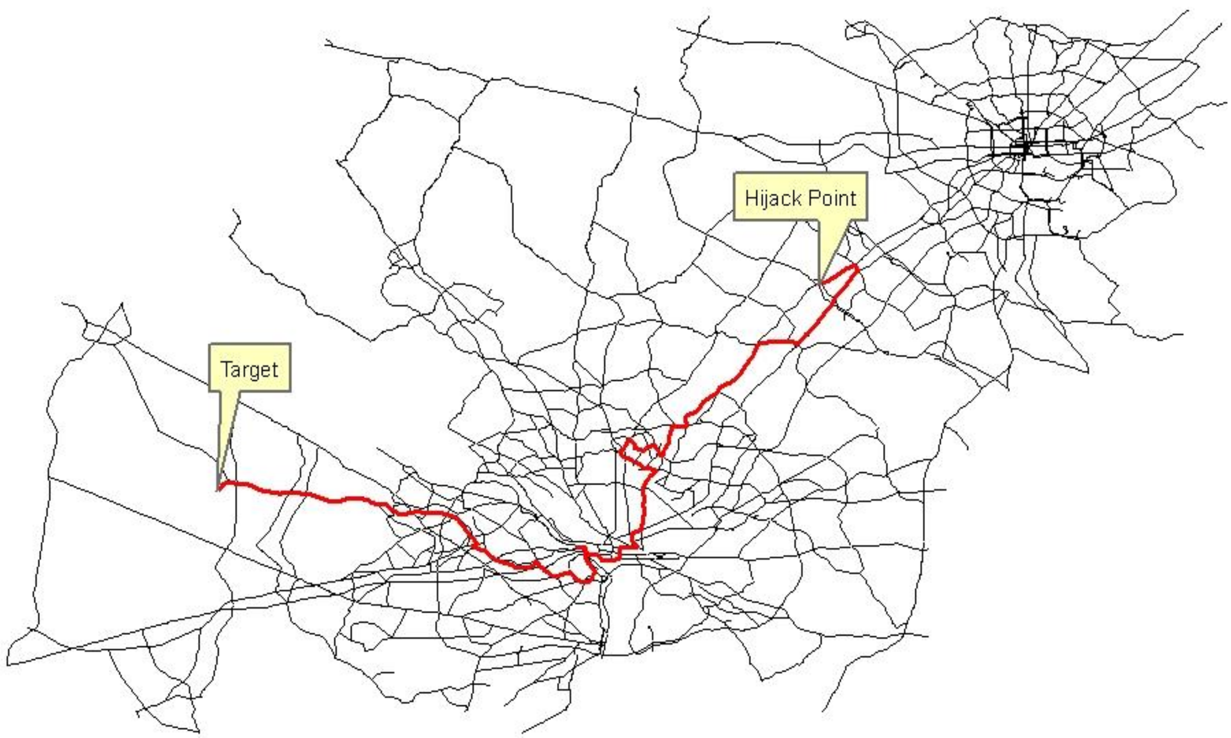



**Figure 14. The path taken by malicious entity for value of $\lambda$ =0.7 towards IAD.**

From $\lambda = 0.7$, the paths spans the centre of Washington DC which is a very high consequence area.While the travel time on this path is higher than the paths obtained for $\lambda = 0.1$-$0.6$, the malicious entity is willing to trade this cost for additional consequences.

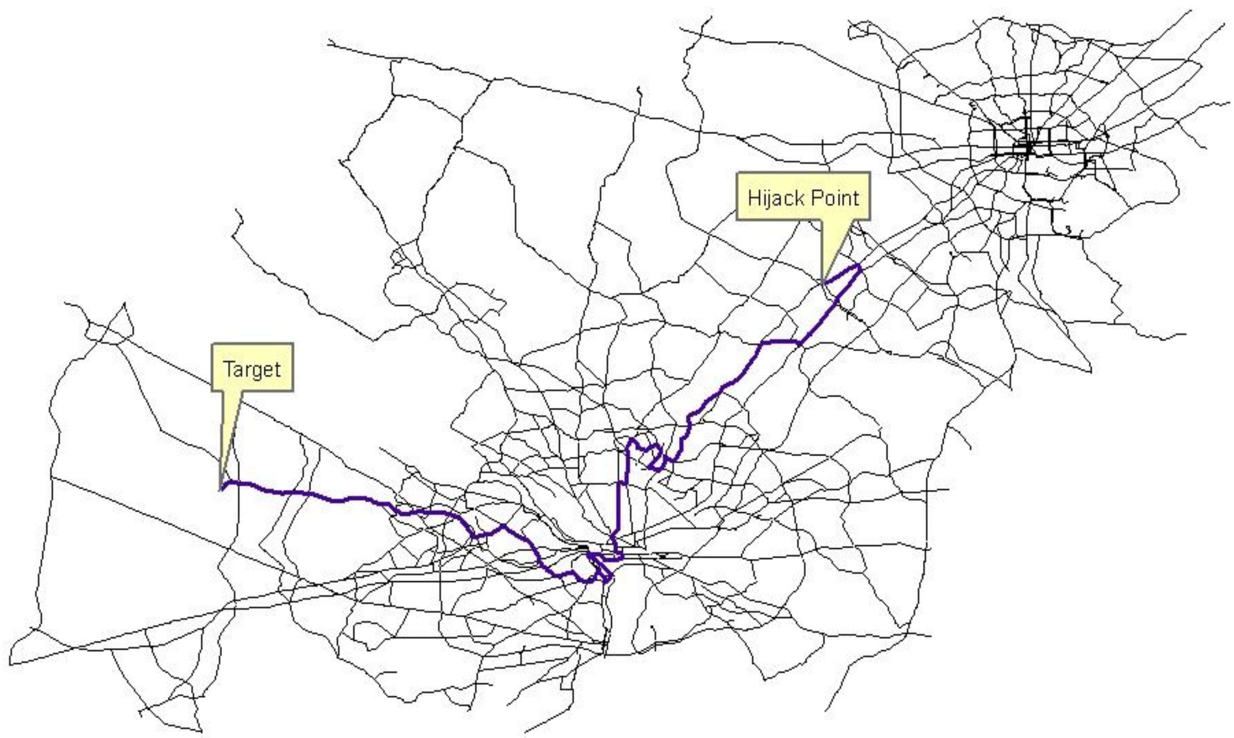



**Figure 15. The path taken by malicious entity for value of λ =0.8 towards IAD.**

The same behavior is observed for the next figures; as the value of λ increases the paths span more of Washington, DC area.

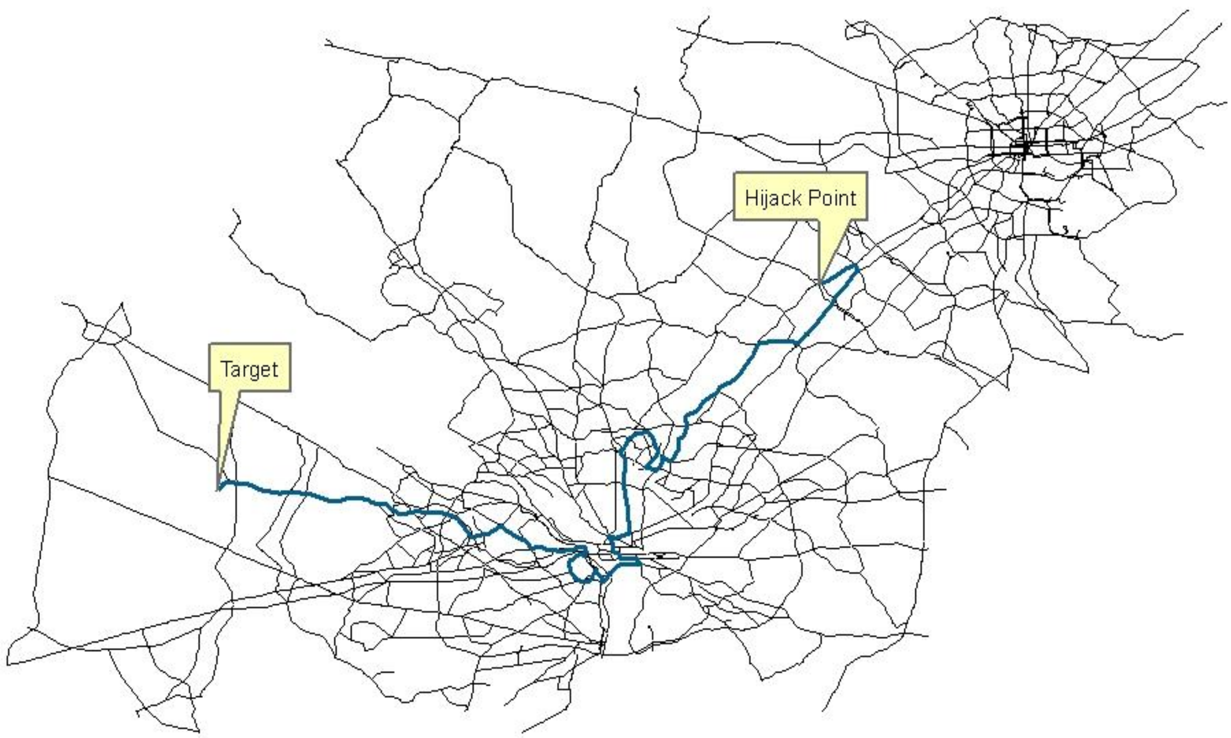



**Figure 16. The path taken by malicious entity for value of λ =0.9 towards IAD.**

The present value of λ reduces the weight of travel time to 0.1, hence the path obtained gives a trip with a very high consequence which can seen in the Figure 16 as it spans high consequence areas of Washington, DC.


**Figure 17. The path taken by malicious entity for value of λ =1.0 towards IAD.**

The paths taken by the hijacker when λ reaches 1.0 might not be practical since they overemphasize the consequence term and do not consider the travel time. This is also not feasible since there is no consideration of travel time and this might lead to failure of the plan itself, as the malicious entity might give a very high time for the law authorities to stop him moving further.

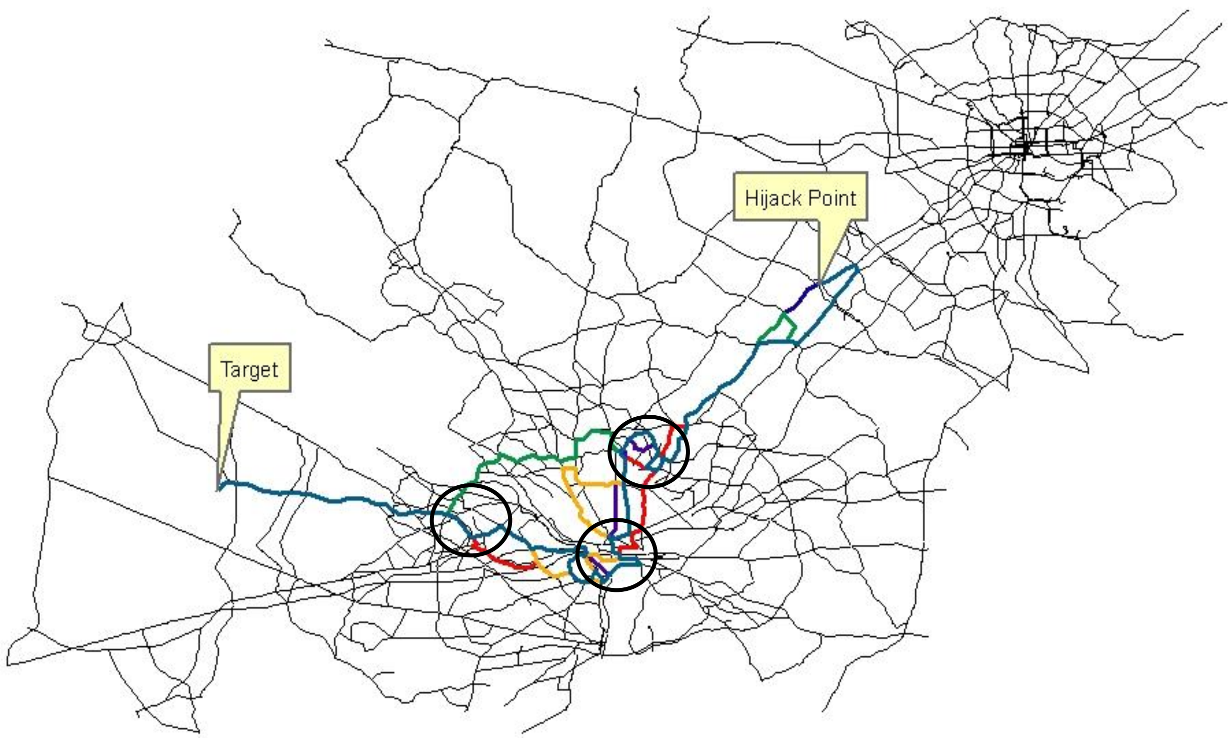



**Figure 18. The path taken by malicious entity for various values of λ towards IAD.**

Figure 18 shows all the paths overlapped Black circles indicate overlap areas where the probability of the hijacker passing is very high.  Extra measures can be taken at these places to prevent the hijacker from moving further.

Contrary to previous target which had routes as a combination of Interstates and national highways. Washington DC is an urban area with access roads with low speeds and many routes to reach the same place. The next target is the National Mall and the most probable paths obtained to the target are shown in Figures 19-25.

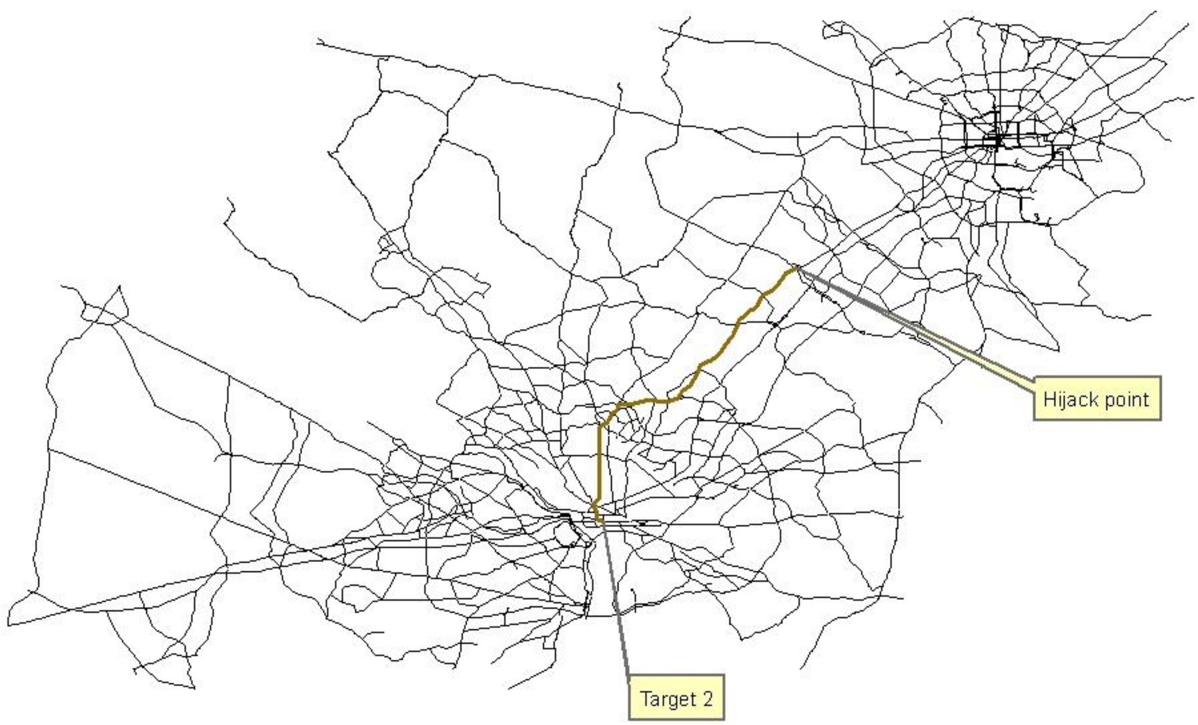



**Figure 19. The path taken towards the National Mall by a malicious entity for λ =0.1 to λ=0.4.**

Figure 19 shows the path obtained when λ ranges from 0.1 to 0.4. Similar to the previous target, travel time dominates the consequence term hence, no change can be observed within this range.


**Figure 20. The path taken towards the National Mall by a malicious entity for λ =0.5.**

A different route (shown in Figure 20) is observed at λ=0.5, though it is not much different from the previous route. Even λ = 0.6 gives the same route as λ =0.5. The variation created by the consequence term in paths is not as high as observed with the previous target (IAD).

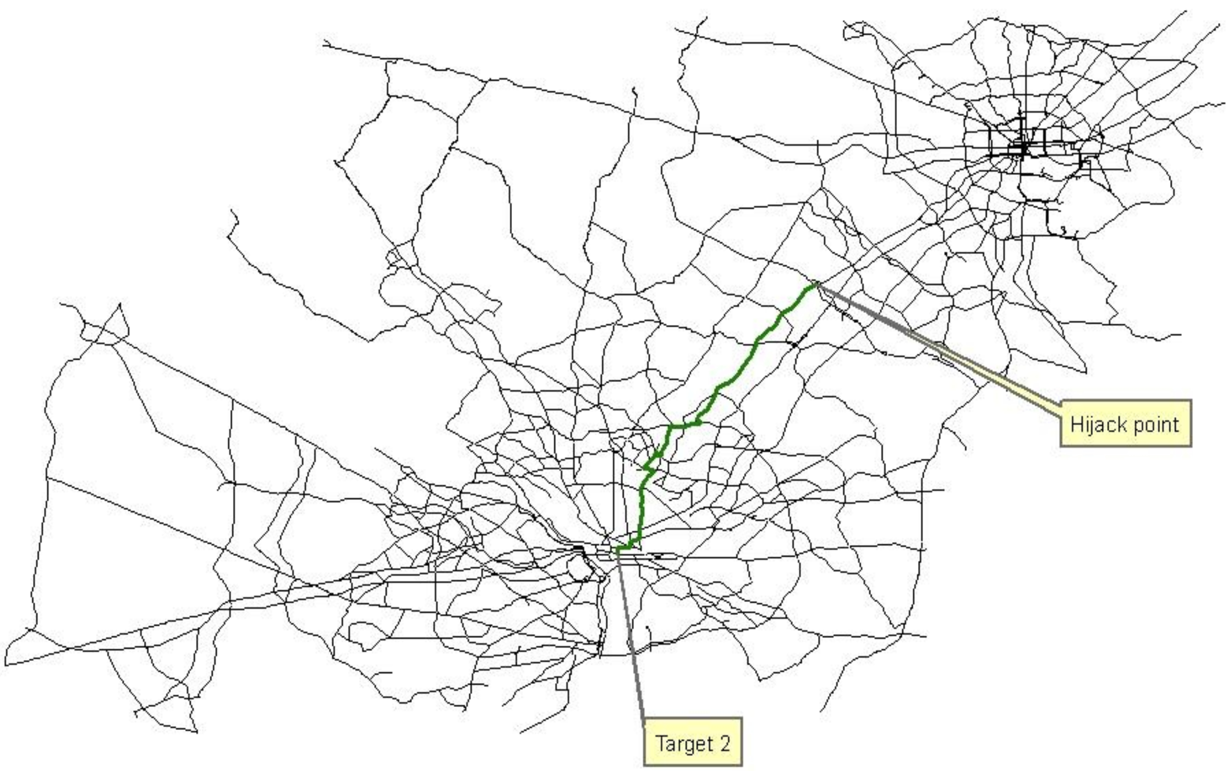



**Figure 21. The path taken towards the National Mall by a malicious entity for λ =0.7.**

The path obtained with λ =0 .7 spans more of Washington, DC.  A small variation can be observed in the route from I 495 towards “target 2” which is not found in Figure 20.

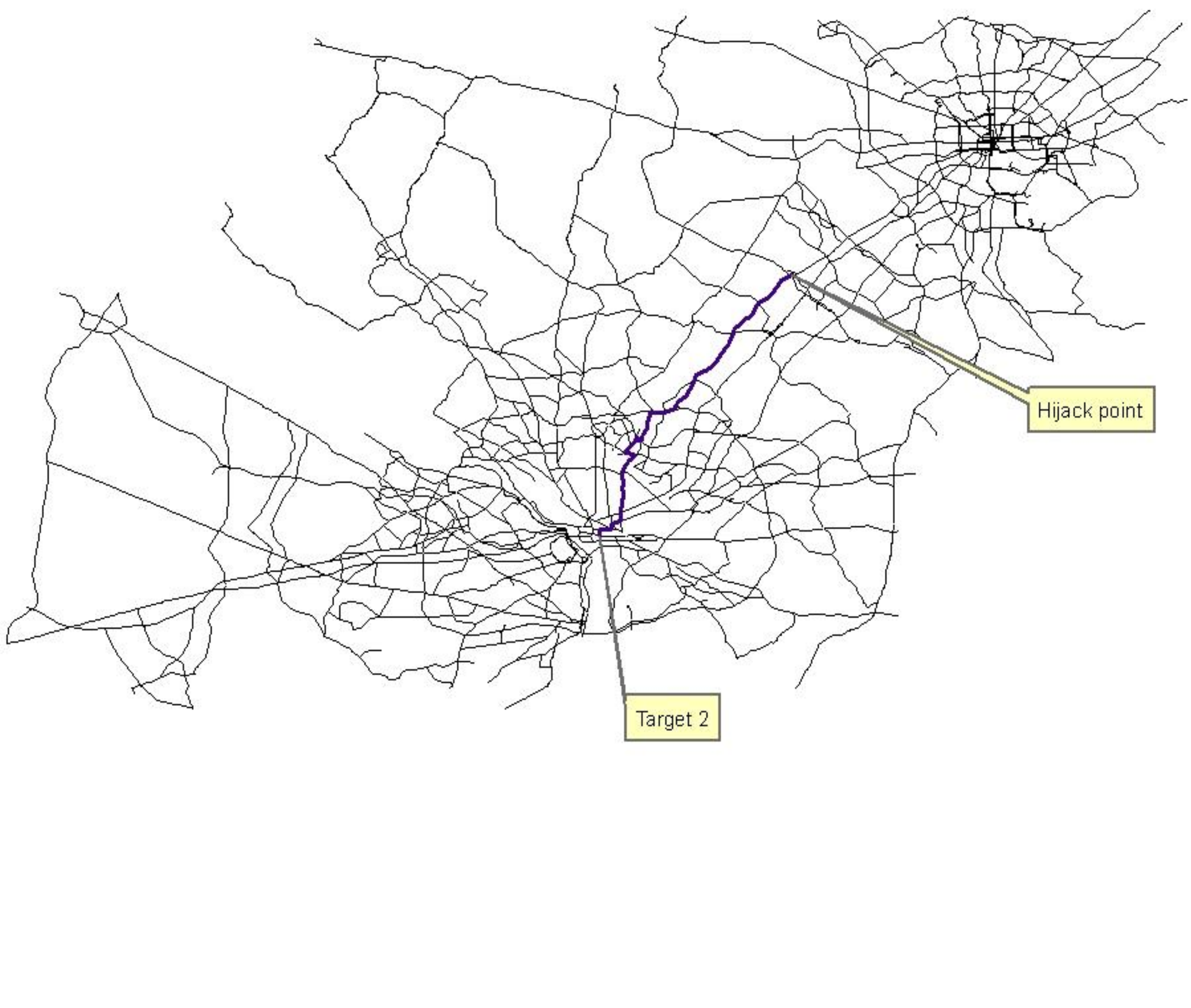



**Figure 22.  The path taken towards the National Mall by a malicious entity for λ =0.8.**

A small variation is observed in the path of the hazmat carrier at λ = 0.8 in Figure 22. The difference is observed at the end of the path near the target. It can be seen that the variation is not really considerable until λ =0.8.

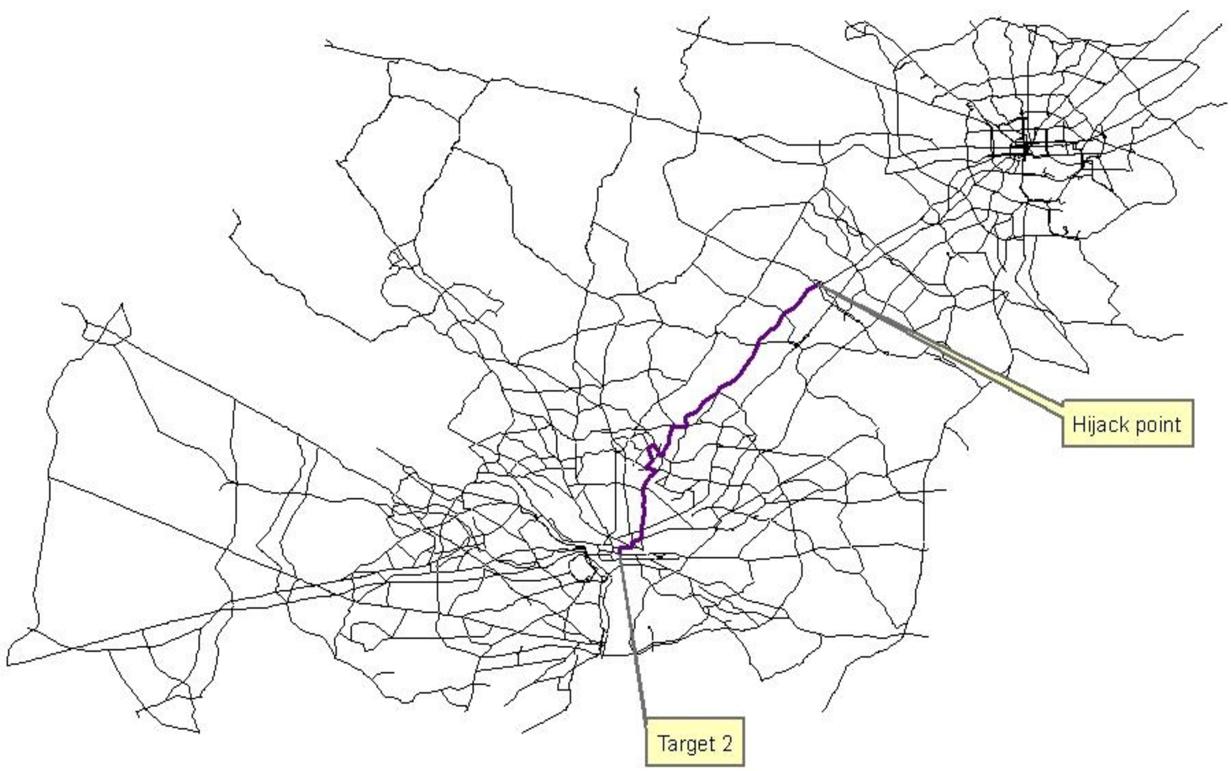



**Figure 23. The path taken towards the National Mall by a malicious entity for λ =0.9**

A variation can be clearly observed when λ =0.9, which implies that consequence term is dominating over travel time. The variation, as mentioned earlier, is evident at the point of contact with I-495.


**Figure 24. The path taken towards the National Mall by a malicious entity for λ =0.95.**

The value of λ is increased by 0.05 to see the behavior of the paths, as the consequence term slowly dominated the travel time. When $\lambda = 0.95$, a completely different path can be seen with hijacker moving away from I- 95 initially and joining back later.

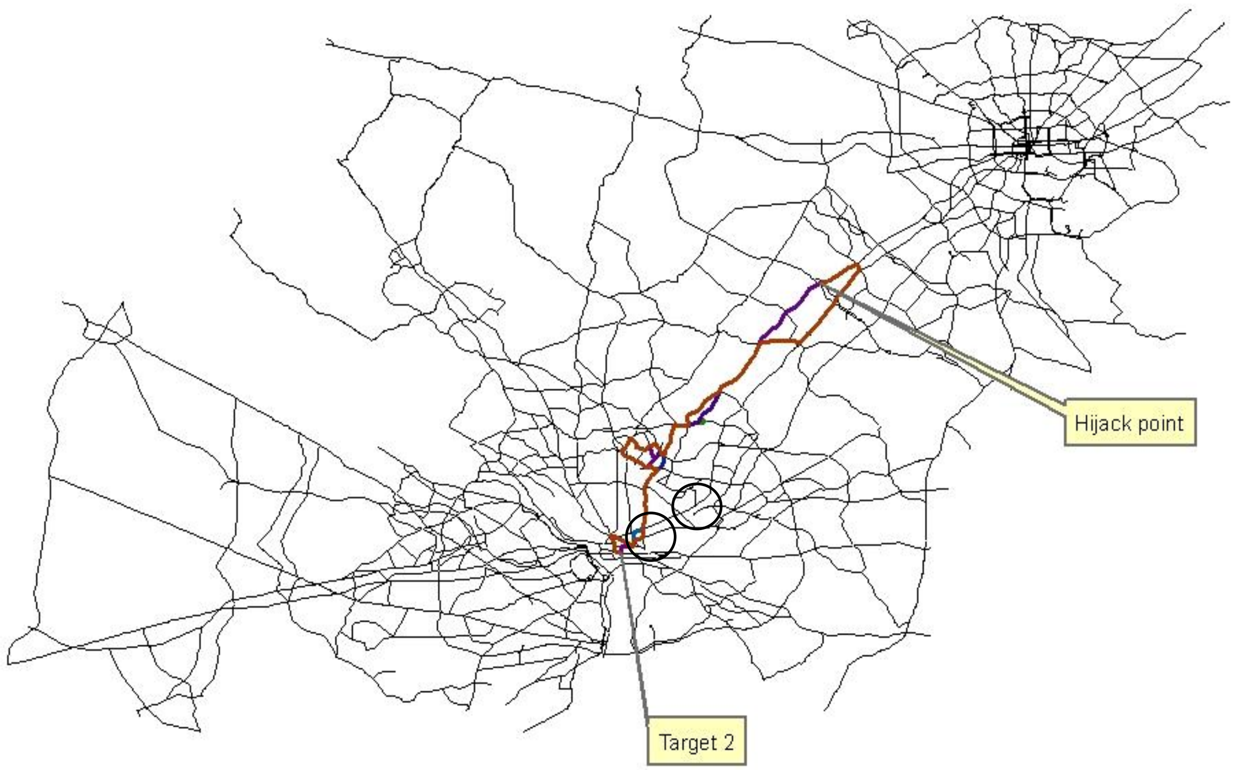



**Figure 25. The path taken towards the National Mall by a malicious entity for various values of λ**

The black circles in Figure 25 show the critical points where extra forces should be introduced. Figure 25 and Figure 18 show the critical points for the corresponding targets. It can be observed that there is not much variation in paths to the National Mall due it its urban location where consequence is equally distributed through the city compared to IAD which is located outside heavily populated area and thus giving a choice between travel time and consequence for the hijacker. An expert panel is consulted for determining the most likely path taken by the hijacker. The results are described in section 5.6.

## 5.2 Neural network training, simulation and creating boundaries for Targets.

The PNN is trained with data obtained from the transportation network. The classification of each node is based on the distance and deviation from the normal route which is shown in Figure 10 with square blocks. The classification of the nodes is shown in Figure 26. The green triangles show the nodes which are marked as safe and either driving on those links connecting them poses no immediate threat or the paths might have been taken due to incidents (The paths are supported by simulation results described in next section). The orange stars indicate nodes that are suspicious i.e. they act as a transition between safe and dangerous areas. The black cross marks indicate dangerous areas or places where once reached by the hazmat truck, the hijack is certain. Few outliers were introduced to consider possible human error in judging a location threat level (safe, suspicious or dangerous).

Hazmat carrier trucks are allowed to change their routes only in the case of incidents. The algorithm for identifying the diversion should take into consideration these special routes incident related. The final assessment of the threat level should take care of three scenarios, first the prediction from neural network results, incident related paths, and finally the allowed distance to the target. The final consideration gives law enforcement agencies buffer time required to stop and prevent the hazmat truck from reaching the target.

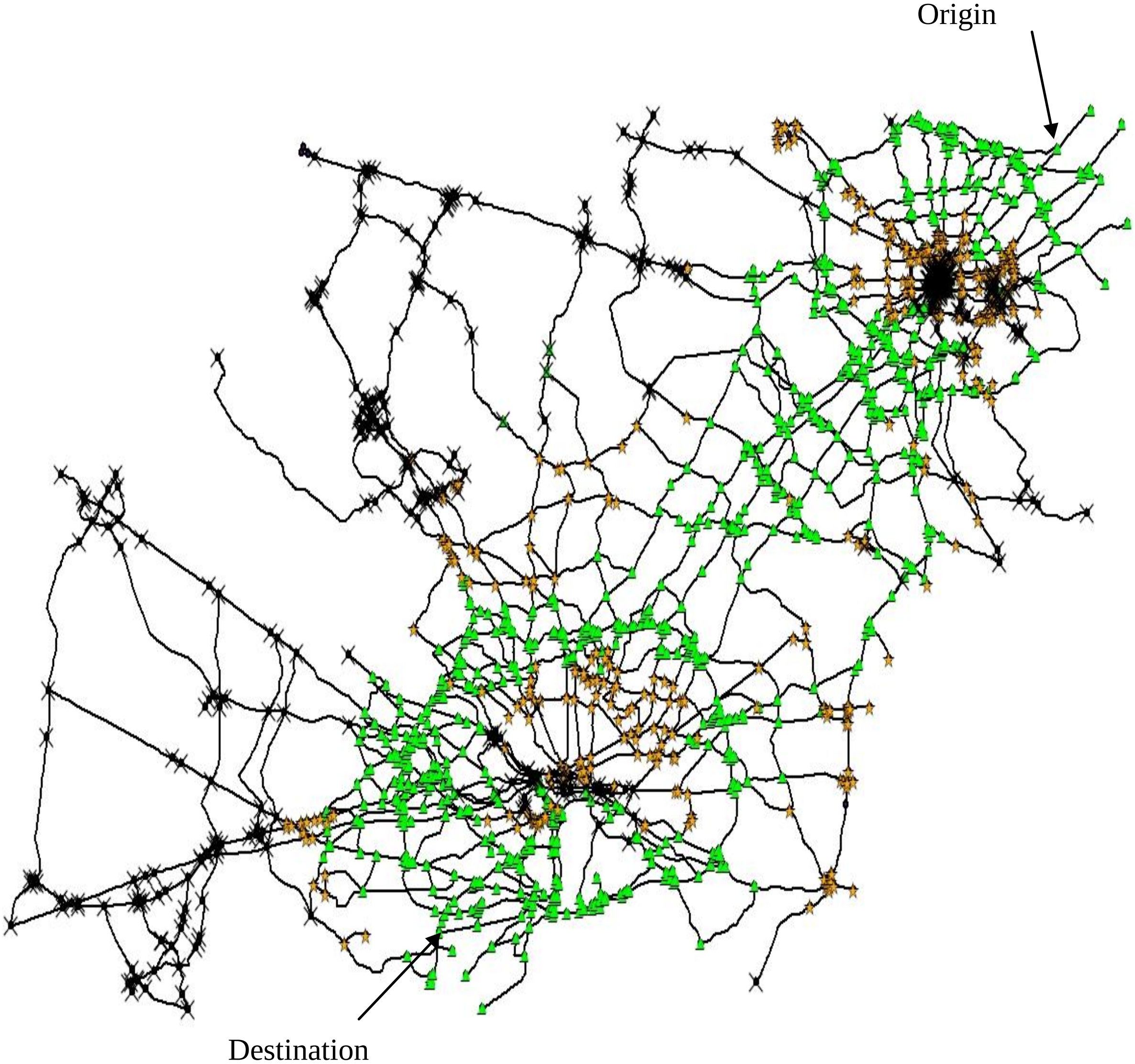



**Figure 26. Network showing classification of network nodes in the network between Baltimore and Washington, DC.**

### 5.2.1 Simulation

As mentioned above, the paths that would be taken by the hazmat driver in the case of incidents need to be taken into consideration for predicting the threat level, and this is achieved by simulating the incidents. The simulation of the network was performed in VISSUM 4.10 software which is macro simulation software available company named PTV in Germany[51]. Federal regulations indicate that congestion (without an incident) cannot be a reason for rerouting. Thus, rerouting is simulated only in the case of incidents where a link is blocked. The new links/routes taken are added as safe/suspicious routes in the data used for training the neural network. The complete path shown in figure 10 was blocked link by link and simulated to see the alternate paths taken by the Hazmat carrier.

### 5.2.2 Neural network training and simulation

A total of 1400 nodes were randomly selected for training the neural network and 370 nodes were used for testing the trained neural network. The shape of the probabilistic neural network can be observed from Figure 27.

From the Figure 26 of the neural network, a 1400 neuron layer is created in the middle in Figure 27. The neural network created is trained with various values of SPREAD (described in chapter 2). Various SPREAD values are selected 0.01 to 1.0. The resultant neural network was tested using 370 points from the transportation network. The predicted threat groups are compared with the actual group values which were initially assigned and were represented in Figure 26. Errors obtained from different values of SPREAD value are shown in Table 2.

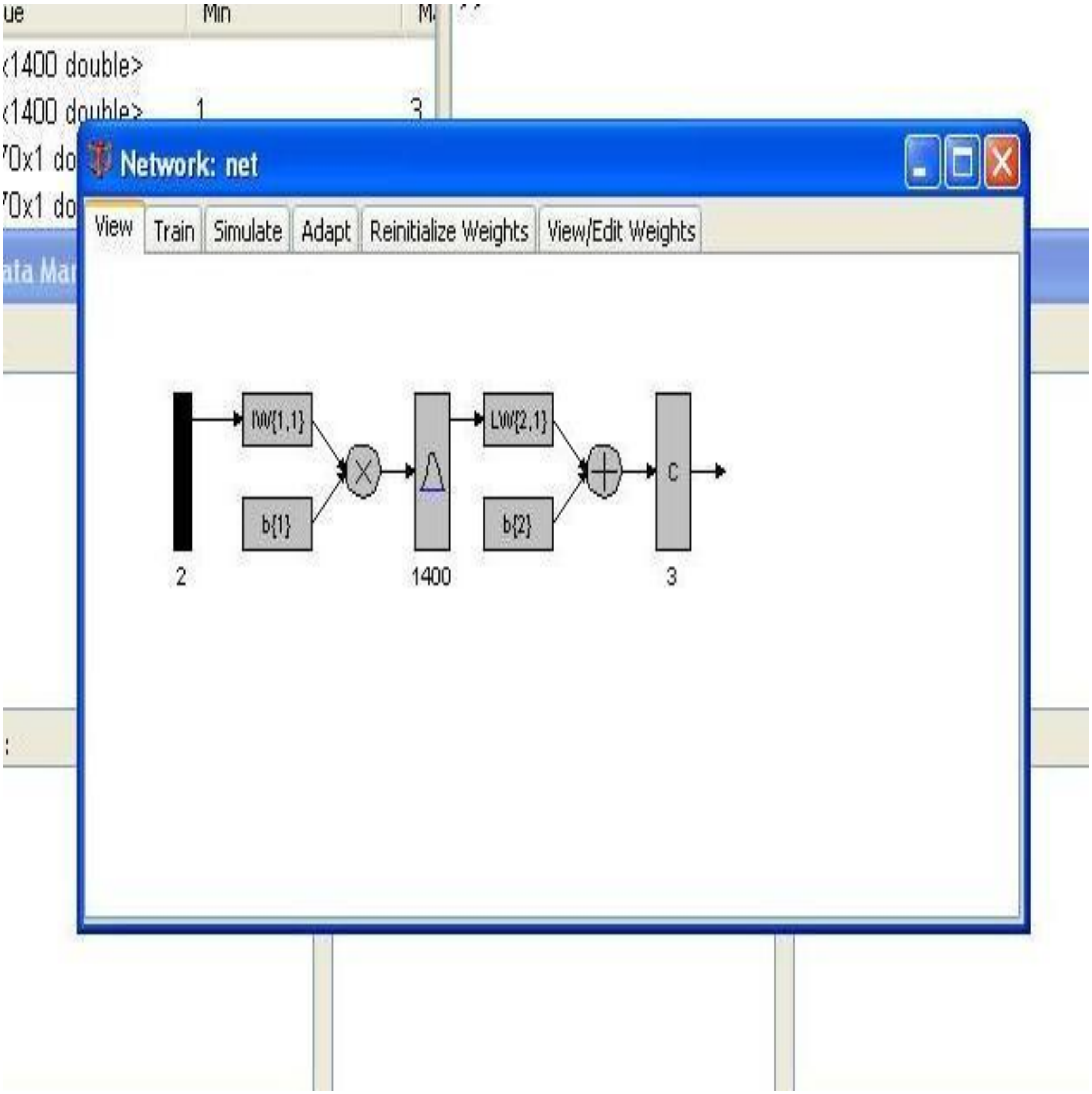



**Figure 27. MATLAB platform for training the PNN**

Table 2. SPREAD value and corresponding error percentages

| SPREAD | Error % |
|---|---|
| **0.01** | 6.486486 |
| **0.1** | 7.297297 |
| **0.2** | 8.648649 |
| **0.3** | 10.54054 |
| **0.4** | 10.81081 |
| **0.5** | 10.81081 |
| **0.6** | 10.81081 |
| **0.7** | 12.7027 |
| **0.8** | 13.78378 |
| **0.9** | 14.05405 |
| **1.0** | 14.59459 |

As the network with a SPREAD value of 0.01 has lowest error percentage, it was chosen for prediction of threat levels. The neural network prediction of the threat levels should then be verified with links which are prohibited around the target in order to provide law enforcement with buffer time for law enforcing authorities and a final threat level should be delivered.

## *5.3 Testing the most probable paths with PNN*

The path(s) obtained from the path prediction methodology section is tested with the trained neural network model to understand the behavior of the developed model in alerting the authorities. Table 3 and Table 4 show the results of the neural network when a path of the malicious entity towards IAD and the National Mall are given as input to the PNN. The paths shown here are for $\lambda = 0.6$(IAD) and $\lambda = 0.7$(National Mall) and rest of paths are shown in Appendix C. The tables show the node number on which the vehicle GPS information is obtained, its corresponding *($r_i$, $\theta_i$)* and the output obtained from the neural network. The nodes are arranged in order from the hijack point to the target.

Category 1 indicates safe nodes, 2 indicates suspicious and 3 indicates dangerous. . Table 3 corresponds to paths in Figure 13 and Table 4 corresponds to Figure 21.

Table 3. The Threat level predicted from the trained neural network for the route shown in Figure 13

| Node Number | **Diversion angle** | **Radial Distance** | **Category** |
|---|---|---|---|
| 7951(Hijack origin) | 11.4783 | 0.7371 | 1 |
| 7982 | 11.4783 | 0.7094 | 1 |
| 8063 | 11.4783 | 0.6605 | 1 |
| 8086 | 11.4783 | 0.6472 | 1 |
| 8097 | 11.4783 | 0.6644 | 1 |
| 8011 | 8.1096 | 0.7612 | 1 |
| 8001 | 8.1096 | 0.7763 | 1 |
| 7984 | 8.1096 | 0.7929 | 1 |
| 7927 | 8.1096 | 0.8838 | 1 |
| 7904 | 8.1096 | 0.9116 | 1 |
| 7884 | 8.1096 | 0.9424 | 1 |
| 7851 | 11.4783 | 0.9268 | 1 |
| 9622 | 11.4783 | 0.8399 | 1 |
| 7781 | 11.4783 | 0.885 | 1 |
| 7699 | 11.4783 | 0.9272 | 1 |
| 7688 | 11.4783 | 0.9442 | 1 |
| 7659 | 11.4783 | 0.9812 | 1 |
| 9620 | 8.1096 | 1.1286 | 1 |
| 7448 | 8.1096 | 1.3056 | 1 |
| 7427 | 8.1096 | 1.3367 | 1 |

| | | | |
|---|---|---|---|
| 7410 | 8.1096 | 1.3559 | 1 |
| 7407 | 11.4783 | 1.353 | 1 |
| 7400 | 14.0699 | 1.3457 | 1 |
| 7384 | 16.2602 | 1.3873 | 2 |
| 7375 | 16.2602 | 1.4315 | 2 |
| 9658 | 16.2602 | 1.4455 | 2 |
| 7363 | 14.0699 | 1.4604 | 1 |
| 7332 | 16.2602 | 1.4731 | 2 |
| 7308 | 16.2602 | 1.4999 | 2 |
| 9617 | 11.4783 | 1.5245 | 1 |
| 7241 | 11.4783 | 1.588 | 1 |
| 7228 | 14.0699 | 1.5822 | 1 |
| 7218 | 14.0699 | 1.6281 | 1 |
| 7213 | 11.4783 | 1.6538 | 1 |
| 7204 | 8.1096 | 1.6628 | 1 |
| 7202 | 8.1096 | 1.6657 | 1 |
| 7186 | 8.1096 | 1.68 | 1 |
| 7071 | 8.1096 | 1.849 | 1 |
| 7042 | 8.1096 | 1.8773 | 1 |
| 7052 | 8.1096 | 1.8707 | 1 |
| 7050 | 11.4783 | 1.8397 | 1 |
| 6995 | 14.0699 | 2.0417 | 1 |
| 6984 | 11.4783 | 2.1308 | 1 |
| 9605 | 11.4783 | 2.2858 | 1 |
| 9607 | 11.4783 | 2.5653 | 1 |
| 6849 | 8.1096 | 2.6532 | 1 |
| 6838 | 8.1096 | 2.6736 | 1 |
| 6833 | 8.1096 | 2.7458 | 1 |
| 6817 | 8.1096 | 2.7999 | 1 |
| 6766 | 8.1096 | 2.9778 | 1 |

| | | | |
|---|---|---|---|
| 6765 | 8.1096 | 3.028 | 1 |
| 6762 | 8.1096 | 3.0634 | 1 |
| 6759 | 8.1096 | 3.0971 | 1 |
| 6757 | 8.1096 | 3.1349 | 1 |
| 6751 | 8.1096 | 3.194 | 1 |
| 6733 | 8.1096 | 3.2685 | 1 |
| 9609 | 0 | 3.4596 | 1 |
| 9642 | 0 | 3.5795 | 1 |
| 6673 | 0 | 3.762 | 1 |
| 6667 | 0 | 3.8846 | 1 |
| 6642 | 8.1096 | 3.7958 | 1 |
| 9641 | 8.1096 | 3.6517 | 1 |
| 6562 | 11.4783 | 3.7474 | 1 |
| 6317 | 19.9484 | 2.8772 | 3 |
| 6276 | 19.9484 | 2.7902 | 3 |
| 6129 | 23.0739 | 2.5818 | 3 |
| 6098 | 23.0739 | 2.5634 | 3 |
| 6027 | 23.0739 | 2.5152 | 3 |
| 6007(IAD) | 23.0739 | 2.4902 | 3 |

Table 4. The Threat level predicted from the trained neural network for the route shown in National Mall Figure 21

| Node Number | Diversion angle | Radial Distance | Category |
|---|---|---|---|
| 7951(hijack origin) | 11.4783 | 0.7371 | 1 |
| 7922 | 11.4783 | 0.7576 | 1 |
| 9622 | 11.4783 | 0.8399 | 1 |
| 7781 | 11.4783 | 0.885 | 1 |

| | | | |
|---|---|---|---|
| 7699 | 11.4783 | 0.9272 | 1 |
| 7688 | 11.4783 | 0.9442 | 1 |
| 7659 | 11.4783 | 0.9812 | 1 |
| 9620 | 8.1096 | 1.1286 | 1 |
| 7501 | 8.1096 | 1.2075 | 1 |
| 7493 | 8.1096 | 1.2253 | 1 |
| 7488 | 8.1096 | 1.2402 | 1 |
| 7498 | 11.4783 | 1.2505 | 1 |
| 7497 | 11.4783 | 1.251 | 1 |
| 7479 | 8.1096 | 1.2647 | 1 |
| 7459 | 8.1096 | 1.294 | 1 |
| 7448 | 8.1096 | 1.3056 | 1 |
| 7427 | 8.1096 | 1.3367 | 1 |
| 7410 | 8.1096 | 1.3559 | 1 |
| 7407 | 11.4783 | 1.353 | 1 |
| 7400 | 14.0699 | 1.3457 | 1 |
| 7384 | 16.2602 | 1.3873 | 2 |
| 7375 | 16.2602 | 1.4315 | 2 |
| 9658 | 16.2602 | 1.4455 | 2 |
| 7356 | 18.1949 | 1.4432 | 1 |
| 7324 | 19.9484 | 1.4338 | 2 |
| 7337 | 21.5652 | 1.4272 | 2 |
| 7366 | 21.5652 | 1.3671 | 2 |
| 7331 | 21.5652 | 1.4385 | 2 |
| 7306 | 21.5652 | 1.5339 | 2 |
| 9615 | 18.1949 | 1.9738 | 2 |
| 7305 | 18.1949 | 2.0911 | 2 |
| 7273 | 18.1949 | 2.2182 | 2 |
| 7259 | 18.1949 | 2.2556 | 2 |
| 7257 | 16.2602 | 2.3385 | 2 |

| | | | |
|---|---|---|---|
| 7231 | 16.2602 | 2.5448 | 3 |
| 7233(National Mall) | 16.2602 | 2.6999 | 3 |

Interpreting the results of the neural network is critical to identify the diversion of the hazmat vehicle. The yellow shades in Table 3 and Table 4 show the transition of the vehicle from a safe zone to dangerous zone.

An alarm to law enforcement agencies can be given as soon as a suspicious zone is detected in the path of the vehicles like that shown in the above table or a few more observations could be made about the truck threat level to deal with the diversion on a conservative side, since the accuracy percentage is 94% rather than perfect. It can be also seen that in Table 3 the vehicle moves into a suspicious zone and then comes back to a safe zone. In such cases, waiting for a few more observation to confirm diversion might be a better choice. Thus final decision by the authorities on diversion is dependent on tolerance level of the law agency using the path diversion detecting methodology.

## *5.4 Validation of assumption in the malicious entity path prediction methodology.*

To validate the assumptions which were made during the development of the methodology for obtaining the probable paths, expert opinion was solicited. The questions that were asked to expert are listed in appendix B.

The questions were given to a Transportation Research Board Critical Transportation Infrastructure Protection Committee member and two officials from the Department of Homeland Security who requested anonymity. A total of three experts have presented their opinion about the situation. Their responses are summarized in Tables 5-8.

The base scenario of the situation was explained to the experts i.e. the hijacking scenario and the few questions were communicated to the panel in written form. The expert panel was asked to answer the questions with positioning themselves as terrorists involved in the hijacking scenario.

**Table 5. Response of the expert panel to the Question 1**

| Questions | Expert 1 | Expert 2 | Expert 3 | Range |
|---|---|---|---|---|
| 1.Importance for target selection<br>(Population size\| Historic importance \| Huge infrastructure) | 1\|2\|1 | 1\|4\|3 | 1\|3\|3 | 1-5<br>1 –high<br>2-low |
| 2.Importance for Secondary location<br>Population\|Accessibility\| | 2\|2 | 2\|1 | 2\|1 | 1-5<br>1 –high<br>2-low |
| Preferred path | 3 | 2 | 2 | 1-10<br>1-shortest path<br>10- high consequence path |

Additional comments: Include likeliness of detected and stopped

All the experts gave highest importance to population in choosing both the main target and secondary locations. Accessibility was also given a very high rating by the experts. The same factors were taken into consideration for determining malicious routes in this work. Though the likelihood of getting caught has been used in the formulation, due to the unavailability of data, it was assumed to be constant.

As far as preference of paths is concerned, the experts preferred a path rated with either 2 or 3 for question 3.As seen earlier the paths for IAD did not change for $\lambda$ values between = 0.1 and 0.4. Between $\lambda$ values of 0.5 and 1.0 changes were observed. Similar behavior was observed for the National Mall. Thus from the experts opion about preferred path , Figure 13 or 14 is the preferred path for the malicious entity moving towards IAD and Figure 21 is preferred by the malicious entity moving towards the National Mall. The paths obtained with help of expert panel gave higher preference to travel time rather than consequence.

# CHAPTER 6: Summary, Conclusions, and Future Work

The chapter is organized as follows Section 6.1 gives a summary of the methodologies used for route diversion detection and estimating the most probable paths taken by malicious entities. Section 6.2 provides conclusions for the methodologies used and results obtained. Section 6.3 details the possible future work possible.

## *6.1 Summary*

Previous research on hazmat routing has concentrated on two major aspects. First are the different procedures for calculating the consequence value based on the possible population exposed, kind of hazard, and other factors. The second are the various methodologies for obtaining the safe route for hazmat carriers. Very few researchers examined the possibility of hijack and subsequent decisions taken by the malicious entity. The present work extends the hazmat routing and analyzes the possibility of hijack and possible routing decisions made by the malicious entity after the hijack.

### 6.1.1 Route Diversion Identification Methodology

Identifying the diversion of a hazmat carrier is crucial and was done previously using various methods such as geo fencing. In the present work, neural networks were used for predicting the diversion in the hazmat vehicle paths. PNN are good classification tools, which can handle large amounts of data and are well suited for real world non linear boundary problems. To use PNN, a methodology was developed to represent the nodes in the road network using between Baltimore and Washington, DC. A total of 1400 data points were used for training the network and 370 points were used for testing. A classification error of 6.486 % was obtained, indicating a good classification performance for real world applications.

It is the first work towards using Artificial Intelligence in the hazmat field and would be useful to implement in case authorities have to track a large number of hazmat shipments per day[48].

### 6.1.2 Prediction of Probable Paths Taken by Malicious Entity

The second part of the thesis focused on developing a procedure for predicting paths taken by the hijacker once the hijacking point and possible target are known. The work developed a new consequence estimation methodology based on perceived consequence rather than the actual consequence caused when a hijacker uses a hazmat truck as a weapon. The perceived consequence is the actual factor that determines hijacker route selection for two reasons. First, the hijacker is not informed with the actual consequence values since it is difficult to assess and also varies considerably with time of day, day of the week, and season. Secondly, it is uncertain that the complete population in the city would be affected by one hazmat truck, but the hijacker would try to maximize the destruction.

This consequence methodology is developed to assess the perceived consequence and correspondingly assign the consequence to the links passing nearby the populated centers. The methodology was applied on the network between Washington, DC and Baltimore. Around 300 populated areas span this network. The Washington, DC population centre was split into four areas and diagrams corresponding to them can be seen in the Appendix B.

A new algorithm based on the breath first search was developed to handle fractional programming[49]. Different routes were obtained by varying the weights to consequence and travel time to two targets, one at IAD and other being National Mall. The assumptions that were made for developing the methodology were validated with the help of an expert panel. The panel has given a high rating for population when choosing a target and, in case of secondary locations; population and accessibility distance were given the same importance. The panel recommended the inclusion of the factor "likeliness of getting caught" in their path

## *6.2 Conclusions*

### 6.2.1 Path diversion methodology

Probabilistic neural networks had very good classification ability in the present research especially involving spatial components. The methodology identifies the movement of vehicle from safe to suspicious and then dangerous zones and alert the authorities.

### 6.2.2 Path prediction methodology

The critical points of intersection were obtained on the map when the routes to a particular target are overlapped. These critical points can be places where extra security measures can be installed for effective stopping of malicious entity with hazmat truck. A wide variation of routes was obtained for IAD being target when compared with National Mall.

The path prediction methodology is an extension to the work done until now in the hazmat security area. The work is a very initial methodology for predicting the paths taken by the hijacker.

## *6. 3 Future work*

The present work is an initial methodology and can be improved in different ways. The section 6.3.1 explains three different directions in which the neural network methodology can be improved and applied in other fields and section 6.3.2 explains three future directions for improving the path prediction methodology.

### 6. 3.1 Route Diversion Identification Methodology

In the present work, the methodology for training the neural network is focused on a single origin and destination. The parameters were developed taking into consideration the pre specified origin and destination, thus, the procedure might not be suited for multiple destinations. A general procedure should be developed by using some standard acceptable deviant distances from the main standard route, supplemented with a look up table developed to verify with the exceptions such as route blocking for special situations. A methodology should also be developed to determine when to initiate the alarm for the law enforcement agencies in case a suspicious zone is detected in the path of the vehicle, depending on the kind of hazmat and acceptable error for the agency.

This problem can also be addressed by shifting the origin every time the hazmat truck moves towards a new node and then localizing the decision there. As PNN is trained very quickly, implementation is feasible in the real world. Various different types of neural networks have been developed for faster and efficient computation, which should be investigated.

The procedure with some variations can also be used for detecting diversion in planes which have three dimensional movements where extra inputs like altitude, speed angle should be considered. This work would require a modified methodology to fit three dimensional coordinates along with the speed of the plane.

### 6.3.2 Prediction of Probable Paths Taken by Malicious Entity

The path prediction methodology developed in the present work has taken travel time corresponding to free flow speed. As the second target is located in an urban arena, congestion is an integral part of travel time and should be considered in the future. The congestion data can be obtained either from real world data or through simulation. This would increase the accuracy of the paths obtained as the hijacker would not prefer the hazmat carrier to be stuck in traffic for a long time.

The likelihood of getting caught is an important factor for determining the hijacker routes. Though the factor was initially included in this work, it was treated as a constant due to non availability of data.  This data should be pursued in future work. As the hijacking events are relatively rare, data pertaining to events are generally not available Hence, a full survey of security officials would justify and improve the methodology.

The work belongs to the area of hazmat transportation security. This study addresses a gap in the literature by considering the aspect of hijacking. The work addresses two aspects of hijacking: first, detection of diversion from the standard route and second, finding the most probable paths taken by the hijacker to reach one of the likely targets. Neural networks were used for predicting the diversion, and a new methodology based on perceived consequence is developed to identify the probable paths of hijacker. Both the methodologies gave satisfactory results when tested on the network between Baltimore and Washington, DC.

# REFERENCES

1. Bureau of Transportation Statistics. *National Transportation Statistics*. 2007 [cited 2007 4 November]; Available from: http://www.bts.gov/publications/national_transportation_statistics/.

2. Taboada, J., et al., *Risk Communication : Around the world Neural Networks Models for assessing Road suitability for dangerous Goods transport* Human and Ecological Risk Assessment, Taylor and Francis Group, 2006. **12**: p. 17.

3. Office of Hazardous Material Safety *Emergency Response Guide*. 2004 [cited 2007 July 4]; Available from: http://hazmat.dot.gov/pubs/erg/erg2004.pdf.

4. Brown,D.F. et al., *A Risk Assessment for National Transportation of selected Hazardous Materials.* Transportation Research Board, 2001.

5. Bureau of Transportation Statistics. *2002 Commodity Flow Survey Hazardous Materials*. 2007 [cited 2007 August 7]; Available from: http://www.bts.gov/publications/commodity_flow_survey/2002/hazardous_materials/.

6. Jesse McKinley and C. Marshall., *Overpass Near San Francisco Collapses After Fire*, in *New York Times*. 2007: San Francisco.

7. United States Department Of Transportation. *Enhancing Security of Hazardous Material* 2002 [cited; Available from: http://hazmat.dot.gov/riskmgmt/rmsef/rmsef.htm.

8. National Highways Institute, *Highway Routing of Hazardous Materials Guidelines for Applying Criteria*, NHI, Editor. 1996. p. 134.

9. Federal Motor Carrier Safety Association, F.M.C.S.A. *Transportation of Hazardous Materials; Driving and Parking Rules*. 2007 [cited 2007 30 July]; Available from: http://www.fmcsa.dot.gov/rules-regulations/administration/fmcsr/fmcsrruletext.asp?section=397.67#.

10. Harwood, D.W., J.G. Viner, and E.R. Russell, *Procedure for Developing Truck Accident and Release Rates for Hazmat Routing.* Journal of Transportation Engineering-Asce, 1993. **119**(2): p. 189-199.

11. Garrido, R.A. and A. Bronfman., *Road Pricing in Urban Hazardous Materials Transportation: A Practical Assessment*, in *PIARC Seminar on Road Pricing with Emphasis on Financing, Regulation and Equity*. 2005: Cancun, Mexico.

12. Kara, B.Y., E. Erkut, and V. Verter., *Accurate calculation of hazardous materials transport risks.* Operations Research Letters, 2003. **31**: p. 285-292.

13. Frank, W.C., J.C. Thill, and R. Batta, *Spatial decision support system for hazardous material truck routing.* Transportation Research Part C-Emerging Technologies, 2000. **8**(1-6): p. 337-359.

14. Carotenutoa, P., S. Giordanib, and S. Ricciardellib., *Finding minimum and equitable risk routes for hazmat shipments.* Computers and operation research, 2005. **34**: p. 1304-1327.

15. Nemhard, D.A. and C. C.White, *Application of Non-Order-Preserving Path Selection to Hazmat Routing.* Transportation Science, 1997. **31**(3): p. 262-271.

16. Erkut., E. and A. Ingolfsson, *Catastrophic avoidance models for hazardous material route planning* Transportation science, 2000. **34**(2): p. 165-179.

17. Erkut, E. and A. Ingolfsson., *Transport risk models for hazardous materials: revisited.* Operations Research Letters, 2005. **33**: p. 81-89.

18. Huang, B. and P. Fery, *Aiding Route Decision for Hazardeous Material Transportation* Transportation Research Board 2005, 2005.

19. Erkut, E. and O. Alp, *Integrated routing and scheduling of hazmat trucks with stops en route.* Transportation Science, 2007. **41**(1): p. 107-122.

20. Chang, T.S., L.K. Nozick, and M.A. Turnquist, *Multiobjective path finding in stochastic dynamic networks, with application to routing hazardous materials shipments.* Transportation Science, 2005. **39**(3): p. 383-399.

21. Carotenuto, P., et al., *A tabu search approach for scheduling hazmat shipments.* Computers & Operations Research, 2007. **34**(5): p. 1328-1350.

22. Meng, Q., D.H. Lee, and R.L. Cheu, *Multiobjective vehicle routing and scheduling problem with time window constraints in hazardous material transportation.* Journal of Transportation Engineering-Asce, 2005. **131**(9): p. 699-707.

23. Alumur, S. and B.Y. Kara, *A new model for the hazardous waste location-routing problem.* Computers & Operations Research, 2007. **34**(5): p. 1406-1423.

24. Jacobs, T.L. and J.M. Warmerdam, *Simultaneous Routing and Siting for Hazardous-Waste Operations.* Journal of Urban Planning and Development-Asce, 1994. **120**(3): p. 115-131.

25. Zhang, J.J., J. Hodgson, and E. Erkut, *Using GIS to assess the risks of hazardous materials transport in networks.* European Journal of Operational Research, 2000. **121**(2): p. 316-329.

26. Boulmakoul, A., *Fuzzy graphs modelling for HazMat telegeomonitoring.* European Journal of Operational Research, 2006. **175**(3): p. 1514-1525.

27. Sherali, H.D., S. Subramanian, and P. kachroo. *Incident response: crew scheduling and rerouting of hazmat carriers*. in *Proceedings of SPIE, the international society for optical engineering*. 1998.

28. Beroggi, G.E.G., *A Real-Time Routing Model for Hazardous Materials.* European Journal of Operational Research, 1994. **75**(3): p. 508-520.

29. Huang, B., C.R. Long, and Y.S. Liew., *GIS and genetic algorithms for HAZMAT route planning with security considerations.* International Journal of Geographical Information Science,Taylor and francis 2004. **18**(8): p. 769-787.

30. Murray-Tuite, P., R.H. Garrido, and R. Nune., *Path Prediction Methodology for Hazardous Materials Transported by Malicious Entities*, in *11th World Conference on Transport Research and publication*. 2006: UCB.

31. Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern Classification* 2 ed. 2004 Wiley-Interscience.

32. Sun, P., *Outlier Detection in High Dimensional. Spatial and Sequential Data Sets*, in *Computer Science* 2006, The University of Sydney: Sydney. p. 118.

33. Kriegel., H.-P. *Density-Based Cluster- and Outlier Analysis*. 2004 [cited 2007 October 12]; Available from: http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/Clustering/.

34. Sander, J., et al., *Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications.* Data mining and knowledge discovery, 1998. **2**(2).

35. J.A.Benediktsson, P.H.Swain, and O.K.Ersoy., *Neural Network Approaches versus Statistical Methods in Classification of Mutlisource Remote Sensing Data* IEEE transactions on geosciences and remote sensing. **28**(4): p. 540-552.

36. Hardin, P.J., *Neural networks vs. nonparametric neighbor-based classifiers for semisupervised classification of Landsat imagery*, in *Part of the SPIE Conference on Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation II*. 1999: Denver, Colorado.

37. *Artificial neural network*. 2007 [cited 2007 14 July]; Available from: http://en.wikipedia.org/wiki/Artificial_neural_network.

38. MathWorks. *Probabilistic Neural Networks*. 2006 [cited 2006 29 December]; Documentation of neural networks tool box]. Available from: http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/nnet.shtml.

39. Specht, D.F., *Probablistic Neural Networks* Neural Networks 1989. **3**: p. 109-118.

40. Sherrod, P.H. *Probabilistic and General Regression Neural Networks*. 2007 [cited 2007 1 August]; Available from: http://www.dtreg.com/pnn.htm.

41. K.Z.Mao, K.C.Tan, and W.Ser., *Probablistic Neural Network Structure Determination for Pattern Classification.* IEEE Transactions on Neural Networks 2000. **11**(4): p. 8.

42. Nune, R., et al., *Artificial Neural networks-Genetic algorithm based model for backcalculation of pavement layer moduli* International Journal of Pavement Engineering 2006**7**(3): p. 221-230.

43. Sayed., T., W. Abdelwahab., and F. Navin., *Identifying Accident -Prone Locations Using Fuzzy Pattern Recognition* Journal of Transportation Engineering, 1995. **121**(4): p. 352-358.

44. Zhang, H. and D. Luo, *A New Method for Traffic Signs Classification Using Probablistic Neural Networks*, in *Advances in Neural Networks-ISNN 2006*. 2006, Springer-Verlag: Chengdu, China. p. 33-39.

45. Marwala, T., *Fault Classification Using Pseudomodal Energies and Probabilistic Neural Networks.* Journal of Engineering Mechanics, 2004. **13**(11): p. 1346–1355.

46. Sweeney, W.P.J. and M.T. Musavi., *Application Of Neural Networks For Chromosome Classification*, in *Annual Conference on Engineering in Medicine and Biology*. 1993, IEEE, Piscataway, NJ, USA: San Diego, CA, USA.

47. Jin, X., D. Srinivasan, and R.L. Cheu, *Classification of Freeway Traffic Patterns for Incident Detection Using Constructive Probabilistic Neural Networks.* IEEE TRANSACTIONS ON NEURAL NETWORKS, 2001. **12**(5): p. 1173-1187.

48. Gates, K. and M. Filiaggi, R. Nune, Editor. 2007: Falls Church *Personal Communication*.

49. Wang, Y. *Dijkstra Algorithm Consistent with Cyclic Paths* 2006 [cited 2006; Available from: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=7869&objectType=file.

50. Bureau of Transportation Statistics, B.o.T.S. *National Transportation Atlas Database*. 2006 [cited 2007 October 16]; Available from: http://www.bts.gov/publications/national_transportation_atlas_database/2006/.

51. PTV. *VISUM*. 2007 [cited 2007 25 october]; Available from: http://www.ptv-ag.com/cgi-bin/traffic/traf_visum.pl.

52. United States Census 2000. *Your gateway to census 2000*. 2000 [cited 2007 December 12]; Available fromhttp://www.census.gov/main/www/cen2000.html.

# APPENDIX

# Appendix A-MATLAB CODE

## *Code used for paths prediction methodology*

### Code used for generating sequential node numbers and other data matrix required for path prediction

```
% this script is for sorting the nodes and then replaces them with a
% contious numbers so that it is easy for algorithm
clc;
%data contains the matrix described in the methodology section
use_matrix = data(:,2);
use_matrix = [use_matrix' data(:,3)'];
use_matrix = use_matrix' ;
use_matrix =sort(use_matrix);
rep_indices= [];
for i = 1: size(use_matrix)
   if(i~=1)
      if (use_matrix(i)~= use_matrix(i-1))
         rep_indices = [rep_indices use_matrix(i)];
      end
   end
   if(i==1)
      rep_indices(1)= use_matrix(i);
   end



end
```

```
%rep_indices lists all the matrices without repetitions in a single matrix

rep_indices= rep_indices';
i = 1:size(rep_indices);
i = i'; % we need to replace the actual node matrix with the matrix nodes in sequential
manner
%hence are creating a second row each node number is associated with a
%different value this is done for easy computation in dikstra algorithm
rep_indices = [rep_indices i];



data_trial1 = data(:,2); % creations of new matrices
data_trial2 = data(:,3);
data_new1 = zeros(size(data_trial1),1);
data_new2 = zeros(size(data_trial2),1);
for i = 1:size(rep_indices)
  m = find(data_trial1 == rep_indices(i,1)); % find the value to be replaced
  % replacing the vlaue
  data_new1(m) = rep_indices(i,2);
  n = find(data_trial2 == rep_indices(i,1)); % find the value to be replaced
  % replacing the vlaue
  data_new2(n) = rep_indices(i,2);
end
%data_new2(6622) = 6687;
data_new = [data_new1 data_new2]; % final matrix is created using the replaced values
now this is used to create a transmat matrix
%required for optimization algorithm
```

```
% We need to asscociate arc number so that we can easily track it back
% hence the following steps are run
i = 1:size(data_new1);
i = i';
data_new = [i data_new ];
%clear i data _trail1 data_trail2 rep_indices;
% Now i need to from an transmat  matrix
transmat =Inf(1776);
% capture_law = Inf(2212);
consequence =Inf(1776);
 for i = 1:length(data_new1)
%      if distance(i)~= 6000000
%      transmat(data_new(i,2),data_new(i,3)) = distance(i);
%      end
      consequence(data_new(i,2),data_new(i,3)) =  cost_damage(i);
    % capture(data_new(i,2),data_new(i,3))= 1;

 end
 % capture = capture_law;
```

**Algorithm developed to calculate maximum and minimum consequence value**

The algorithm and coding is developed using the file published in mat lab exchange server[49]

```
%This is the primary file which is used for maximum and minimum of the
%parameters in the objectives i.e. minimum and maximum.this code is used
%only for consequence part the distance part is implemeted using the normal
%function [r_path, r_cost] = dijkstra(pathS, pathE, transmat)
% The Dijkstra's algorithm, Implemented by Yi Wang, 2005
% This version support detecting _cyclic-paths_
%
% USAGE:
%  [path, cost]= dijkstra(pathStart, pathEnd,  transMatrix)
%
% PARAMETERS:
%   pathS : the index of start node, indexing from 1
%   pathE : the index of end node, indexing from 1
% transmat: the transition matrix, or adjacent matrix
%

% Ensure the transition matrix is square
%
global num_conseq;
global den_conseq;
if ( size(transmat,1) ~= size(transmat,2) )
    error( 'detect_cycles:Dijkstra_SC', ...
        'transmat has different width and heights' );
end
```

```
tic
% Initialization:
%  noOfNode     : nodes in the graph
%  parent(i)    : record the parent of node i
%  distance(i)  : the shortest distance from i to pathS
%  queue        : for width-first traveling of the graph
noOfNode = size(transmat, 1);
visit = zeros(noOfNode,1);
temp_num_conseq =Inf;
temp_den_conseq =Inf;
temp_dist = Inf;
for i = 1:noOfNode

   parent(i) = 0;
   distance(i) = Inf;
   num_conseq(i) = Inf;
   den_conseq(i) = Inf;
   dist(i) = Inf;
   visit(i)= 0;
end

queue = [];



% Start from pathS


for i=1:noOfNode
   if transmat(pathS, i)~=Inf
      i
      visit(pathS)=1;
```

```
        num_conseq(i)= consequence(pathS,i)*capture(pathS,i);
        den_conseq(i)= capture(pathS,i);
        dist(i) = transmat(pathS,i);
        distance(i) = (0.3)*(consequence(pathS,i)*capture(pathS,i)/capture(pathS,i));
        %+0*transmat(pathS,i);
        distance(i)
        parent(i)   = pathS;
        queue       = [queue i];

    end
end

size(queue)

% Width-first exploring the whole graph
%
while length(queue) ~= 0

    hopS  = queue(1);
length(queue);
    queue = queue(2:end);

    for hopE = 1:noOfNode

        if(transmat(pathS,hopE)==Inf&& transmat(hopS,hopE)~=Inf)

            if (num_conseq(hopS)+consequence(hopS,hopE)*capture(hopS,hopE)~=Inf)

                if( num_conseq(hopE)~=inf)
                    temp_num_conseq = num_conseq(hopE);
```

```
        end
        num_conseq(hopE)
=num_conseq(hopS)+consequence(hopS,hopE)*capture(hopS,hopE);
        %          den_conseq(hopE)= den_conseq(hopS)+capture(hopS,hopE);
        %          dist(hopE)= dist(hopS)+transmat(hopS,hopE);
      end
      if(den_conseq(hopS)+capture(hopS,hopE)~=Inf)
        if( den_conseq(hopE)~=inf)
          temp_den_conseq = den_conseq(hopE);
        end
        den_conseq(hopE)= den_conseq(hopS)+capture(hopS,hopE);
      end
      %        if(dist(hopS)+transmat(hopS,hopE)~=Inf)
      %          if( dist(hopE)~=inf)
      %            temp_dist = dist(hopE);
      %          end
      %          dist(hopE)= dist(hopS)+transmat(hopS,hopE);
      %        end
    end
    k = (0.3)*(num_conseq(hopE)/den_conseq(hopE));

    if(num_conseq(hopE)&&den_conseq(hopE)==Inf)
      k=Inf;

    end

    k = k;

    %0.0*dist(hopE);
    if distance(hopE)< k
```

```
    num_conseq(hopE)=temp_num_conseq ;
    den_conseq(hopE)=temp_den_conseq ;
    dist(hopE)=temp_dist;

end

if distance(hopE)>k &&transmat(hopS,hopE)~=Inf

    % to check whether particular node has been visited or not
    %store in temporary variables to get bac once cycle proved
    temp1=distance(hopE);
    temp2=parent(hopE) ;

    distance(hopE) = (0.3)*(num_conseq(hopE)/den_conseq(hopE));

    %+0* dist(hopE);
    parent(hopE)   = hopS;
    check = parent(hopE);
    for i = 1:2212
        if(check~=pathS)
            check = parent(check);

        end

    end



    if(check~=pathS)

        distance(hopE)=temp1;
```

```
            parent(hopE) = temp2;

          end

          if(check==pathS)

            queue    = [queue hopE];

          end


        end
    end
visit(hopS) = 1;
end

'done'
%
r_path = [pathE];
i = parent(pathE);

while i~=pathS && i~=0
  r_path = [i r_path];
  i     = parent(i);
end

if i==pathS
  r_path = [i r_path];
else
  r_path = []
end
```

```
r_cost = distance(pathE);
 toc
```

## Code for the path prediction methodology

The algorithm and coding is developed using the file published in mat lab exchange server[49]

```
% This version support detecting _cyclic-paths_
%
%
% PARAMETERS:
%   pathS : the index of start node, indexing from 1
%   pathE : the index of end node, indexing from 1
% transmat: the transition matrix, or adjacent matrix
%

% Ensure the transition matrix is square
%
global num_conseq;
global den_conseq;
if ( size(transmat,1) ~= size(transmat,2) )
    error( 'detect_cycles:Dijkstra_SC', ...
        'transmat has different width and heights' );
end

tic
% Initialization:
%  noOfNode     : nodes in the graph
%  parent(i)    : record the parent of node i
```

```
%  distance(i)  : the shortest distance from i to pathS
%  queue        : for width-first traveling of the graph
noOfNode = size(transmat, 1);
visit = zeros(noOfNode,1);
temp_num_conseq =Inf;
temp_den_conseq =Inf;
temp_dist = Inf;
ld = 1.0;
%-----------------for IAD--------------------------%
% Min_distance  49.336
% Max_dist  128.2728
%
% min_consequence   1.15E+05
% max_consequence   1.83E+06


% for national mall
% Min_time  31.9461
% Max_time  44.1092
%
% min_consequence   1.51E+05
% max_consequence   1.80E+06
%-----------------------------------------------------
d_min =31.9461 ;
d_max = 44.1092;
c_max= 2.00E+06;
c_min =2.01E+05;

for i = 1:noOfNode
```

```
    parent(i) = 0;
    distance(i) = Inf;
    num_conseq(i) = Inf;
    den_conseq(i) = Inf;
    dist(i) = Inf;
     visit(i)= 0;
end

queue = [];



% Start from pathS

for i=1:noOfNode
    if transmat(pathS, i)~=Inf
        i
        %  visit(pathS)=1;
        num_conseq(i)= consequence(pathS,i)*capture(pathS,i);
        den_conseq(i)= capture(pathS,i);
       dist(i) = transmat(pathS,i);
        distance(i) = (-ld/(c_max-
c_min))*((consequence(pathS,i)*capture(pathS,i)/capture(pathS,i))-c_min)+((1-
ld)/(d_max-d_min))*(transmat(pathS,i)-d_min);
       % distance(i)
      (ld/(c_max-c_min))*((consequence(pathS,i)*capture(pathS,i)/capture(pathS,i))-
c_min)
      ((1-ld)/(d_max-d_min))*(transmat(pathS,i)-d_min)
        parent(i)   = pathS;
        queue       = [queue i];
```

```
    end
end




% Width-first exploring the whole graph
%
while length(queue) ~= 0

    hopS  = queue(1);
    queue = queue(2:end);
%length(queue)
    for hopE = 1:noOfNode

        if(transmat(pathS,hopE)==Inf)

            if (num_conseq(hopS)+consequence(hopS,hopE)*capture(hopS,hopE)~=Inf)

                if( num_conseq(hopE)~=inf)
                    temp_num_conseq = num_conseq(hopE);
                end

                num_conseq(hopE)
=num_conseq(hopS)+consequence(hopS,hopE)*capture(hopS,hopE);

            end
            if(den_conseq(hopS)+capture(hopS,hopE)~=Inf)
                if( den_conseq(hopE)~=inf)
                    temp_den_conseq = den_conseq(hopE);
                end
```

```
            den_conseq(hopE)= den_conseq(hopS)+capture(hopS,hopE);
        end
        if(dist(hopS)+transmat(hopS,hopE)~=Inf)
            if( dist(hopE)~=inf)
                temp_dist = dist(hopE);
            end
            dist(hopE)= dist(hopS)+transmat(hopS,hopE);
        end
    end
   k = (-ld/(c_max-c_min))*(((num_conseq(hopE)/den_conseq(hopE))-c_min));
%

 if(num_conseq(hopE)&&den_conseq(hopE)==Inf)
        k=Inf;


    end


     k =k+((1-ld)/(d_max-d_min))*(dist(hopE)-d_min) ;



    if(num_conseq(hopE)&&den_conseq(hopE)==Inf)
        k=Inf;


    end




     if distance(hopE)< k&&k~=Inf
      num_conseq(hopE)=temp_num_conseq ;
            den_conseq(hopE)=temp_den_conseq ;
            dist(hopE)=temp_dist;
```

```
end

if distance(hopE) > k
  % &&transmat(hopS,hopE)~=inf

   % to check whether particular node has been visited or not
   %store in temporary variables to get bac once cycle proved
   temp1=distance(hopE);
   temp2=parent(hopE) ;

   distance(hopE) = k;
   parent(hopE)   = hopS;
   check = parent(hopE);
   for i = 1:2212
       if(check~=pathS)
           check = parent(check);
       end
       if(check ==pathS)
           break;
       end
   end
   if(check~=pathS)
       distance(hopE)=temp1;
       parent(hopE) = temp2;
       num_conseq(hopE)=temp_num_conseq ;
       den_conseq(hopE)=temp_den_conseq ;
       dist(hopE)=temp_dist;

   end
```

```
            if(check==pathS)
                queue     = [queue hopE];

            end



        end
    end

end

'done'
% Back-trace the shortest-path
%
r_path = [pathE];
i = parent(pathE);

while i~=pathS && i~=0
    r_path = [i r_path];
    i      = parent(i);
end

if i==pathS
    r_path = [i r_path];
else
    r_path = []
end
```

```
% Return cost
%
r_cost = distance(pathE);

%
%
toc
r_path = r_path';
```

**Code used for creating the required format of GIS to display results**

```
 fid = fopen('exp1.txt','wt');
for i = 1:length(r_path)-1
   li =rep_indices( r_path(i));
   li1= rep_indices(r_path(i+1));
fprintf(fid,' "TONODENO" = %d AND "FROMNODENO"  = %d OR
"FROMNODENO"  = %d AND "TONODENO" = %d OR ',li,li1,li,li1);
end
fclose(fid);
```

***Code used for estimation of consequence***

Code used for calculating distance between population centres and links

```
for i = 1 :length(data)
   index_two= data(i,2); % assigning the starting of link to a variable
    index_three = data(i,3);   % assigning the ending of link to a variable
    a = find(rep_indices == index_two);  % Equivalent number in conversion , starting
point
     b = find(rep_indices == index_three);   % Equivalent number in conversion , ending
point

   data_represented(i,1) =a ;
   data_represented(i,2) = b;

   del_lat = (pi/180)*(nodes_network(a,3) - nodes_network(b,3));
del_long = (pi/180)*(nodes_network(a,2)- nodes_network(b,2));
a_degree = sin(del_lat/2) *sin(del_lat/2) +
cos((pi/180)*nodes_network(a,3))*cos((pi/180)*nodes_network(b,3))*sin(del_long/2)*si
n(del_long/2);
c = 2*asin(min(1,sqrt(a_degree)));
d = 6371*c;
  data_represented(i,3) = d;
  i
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %%% Finding distance between population nodes and links.
   for i = 1: 304
   for j = 1:length(data)
     side_a = dist_population_nodes(i,(data_represented(j,2)));
```

```
      side_c = dist_population_nodes(i,data_represented(j,1));
      side_b = data_represented(j,3);
      length_median = 0.5*sqrt(((2* side_a* side_a)+(2* side_c* side_c))- (side_b*
side_b)); % Formula for length for median
    distance_links_population_nodes(j,i) = length_median;

   end
   i
   end
```

## Code used for assigning consequence to links

```
consequence_links_population =zeros(4654,1);
tic
for i = 1: 1:length(distance_links_population_nodes)
    for j = 1:304
        if(distance_links_population_nodes(i,j)<16.00)
            consequence_links_population(i)= population_perception(j)+
consequence_links_population(i);
        end
    end
end
toc
```

# Appendix B – Questionnaire

## *Question communicated to expert panel*

Questions

1. As a terrorist what factors would you give importance to for selecting your target? Rate the following factors from value of 1 to 5 where 1 being highly important and 5 being very low

| | High | | | | Low | Answer |
|---|---|---|---|---|---|---|
| **Population size** | 1 | 2 | 3 | 4 | 5 | |
| **Historic importance of the target** | 1 | 2 | 3 | 4 | 5 | |
| **Huge infrastructure like<br>Presence of metro stations, sky scrapers** | 1 | 2 | 3 | 4 | 5 | |
| **Any important factor** | | | | | | |

2. As a terrorist, once you hijack the hazmat truck, suppose you would choose routes based on some logical reasoning like presence of secondary locations, travel time etc. Secondary location is defined as a location where you would carry out your plan, if you could not reach your original target which is close to be on your original path.

| **How important is each of the following factors in considering your route based on potential secondary location?** | **High** | | | | **Low** | **Answer** |
|---|---|---|---|---|---|---|
| **Population size** | 1 | 2 | 3 | 4 | 5 | |
| **Accessibility to the location from Your path** | 1 | 2 | 3 | 4 | 5 | |
| **Any other important parameter** | | | | | | |

3. As a terrorist, if you have to choose between the route with the shortest travel time and the route passing through most of the potential secondary location. Please locate the strength of your preference between the two options.

| **Shortest route** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **High consequence path** |
|---|---|---|---|---|---|---|---|---|---|---|---|

Answer:

4. As a hijacker, what would what factors would you consider when you need to plan for your route in above situation. (Optional)

# Appendix C – Data Used

## *Basic data of the network*

### Data of the network

**Table 6. Information about the network in form data matrix**

| Link Number | Starting node | Ending node | Length |
|---|---|---|---|
| 5763 | 5618 | 5581 | 3.508 |
| 5764 | 5563 | 5615 | 15.098 |
| 5764 | 5615 | 5563 | 15.098 |
| 5765 | 5563 | 5851 | 16.411 |
| 5765 | 5851 | 5563 | 16.411 |
| 5770 | 5601 | 5613 | 0.862 |
| 5770 | 5613 | 5601 | 0.862 |
| 5771 | 5601 | 5614 | 0.922 |
| 5771 | 5614 | 5601 | 0.922 |
| 5788 | 5511 | 5586 | 6.658 |
| 5788 | 5586 | 5511 | 6.658 |
| 5789 | 5511 | 5541 | 4.854 |
| 5789 | 5541 | 5511 | 4.854 |
| 6732 | 5547 | 5560 | 12.713 |
| 6732 | 5560 | 5547 | 12.713 |
| 6741 | 5549 | 5583 | 3.334 |
| 6741 | 5583 | 5549 | 3.334 |
| 6742 | 5539 | 5544 | 0.526 |
| 6742 | 5544 | 5539 | 0.526 |
| 6743 | 5544 | 5549 | 0.401 |

| | | | |
|---|---|---|---|
| 6743 | 5549 | 5544 | 0.401 |
| 6744 | 5541 | 5544 | 0.422 |
| 6744 | 5544 | 5541 | 0.422 |
| 6745 | 5544 | 5547 | 0.241 |
| 6745 | 5547 | 5544 | 0.241 |
| 6746 | 5583 | 5601 | 1.276 |
| 6746 | 5601 | 5583 | 1.276 |
| 6747 | 5586 | 5601 | 1.049 |
| 6747 | 5601 | 5586 | 1.049 |
| 6748 | 5560 | 5563 | 4.112 |
| 6748 | 5563 | 5560 | 4.112 |
| 8217 | 9187 | 9298 | 2.643 |

The above matrix gives data about the link number, node numbers on the both sides of the link and distance of the link.

## Population data and location of town centroid

**Table 7. A subset of data used showing the population and location of cities**

| ID | CONTROLTYPE | NAME | XCOORD(lon) | YCOORD(lat) | Perceived Population |
|---|---|---|---|---|---|
| 2 | 0 | Timonium | -76.6197 | 39.437 | 15000 |
| 3 | 0 | Hampton | -76.5725 | 39.4184 | 5000 |
| 4 | 0 | Mays Chapel | -76.6494 | 39.4331 | 10000 |
| 5 | 0 | Carney | -76.5224 | 39.4045 | 30000 |
| 6 | 0 | White Marsh | -76.4313 | 39.3841 | 10000 |
| 7 | 0 | Lutherville | -76.6261 | 39.4212 | 15000 |

| | | | | | |
|---|---|---|---|---|---|
| 8 | 0 | Towson | -76.6013 | 39.4017 | 50000 |
| 10 | 0 | Parkville | -76.5391 | 39.3777 | 30000 |
| 11 | 0 | Rossville | -76.4805 | 39.3549 | 10000 |
| 12 | 0 | Overlea | -76.5196 | 39.3635 | 10000 |
| 13 | 0 | Owings Mills | -76.7797 | 39.4197 | 20000 |
| 14 | 0 | Middle River | -76.4386 | 39.3343 | 20000 |
| 15 | 0 | Garrison | -76.749 | 39.4015 | 5000 |
| 16 | 0 | Pikesville | -76.7219 | 39.3744 | 30000 |
| 17 | 0 | Oakland | -76.9016 | 39.4125 | 2000 |
| 18 | 0 | Rosedale | -76.5149 | 39.3204 | 20000 |
| 19 | 0 | Essex | -76.4746 | 39.3097 | 35000 |
| 21 | 0 | Eldersburg | -76.9429 | 39.4085 | 20000 |
| 22 | 0 | Randallstown | -76.7945 | 39.3676 | 30000 |
| 24 | 0 | Lochearn | -76.7305 | 39.3474 | 20000 |
| 25 | 0 | Milford | -76.7406 | 39.3483 | 20000 |
| 26 | 0 | Milford Mill | -76.77 | 39.3478 | 20000 |
| 27 | 0 | Hebbville | -76.7621 | 39.3416 | 0 |
| 28 | 0 | Woodlawn | -76.7272 | 39.3233 | 35000 |
| 29 | 0 | Sykesville | -76.9678 | 39.3737 | 5000 |
| 30 | 0 | Baltimore | -76.6122 | 39.2904 | 800000 |
| 31 | 0 | Granite | -76.8549 | 39.343 | 0 |
| 32 | 0 | Edgemere | -76.4476 | 39.2423 | 10000 |
| 33 | 0 | Dundalk | -76.52 | 39.2511 | 50000 |
| 34 | 0 | Woodstock | -76.871 | 39.329 | 5000 |

| | | | | | |
|---|---|---|---|---|---|
| 35 | 0 | Mount Airy | -77.1547 | 39.3762 | 5000 |
| 36 | 0 | Catonsville | -76.7313 | 39.2721 | 35000 |
| 37 | 0 | Frederick | -77.4105 | 39.4143 | 50000 |
| 38 | 0 | New Market | -77.2694 | 39.3826 | 300 |
| 39 | 0 | Lansdowne | -76.6605 | 39.2451 | 15000 |
| 40 | 0 | Bartonsville | -77.358 | 39.3926 | 10000 |
| 41 | 0 | Brooklyn Park | -76.6159 | 39.2287 | 10000 |

The above table is a subset obtained from the census.gov website[52] and then modified to insert the latitude, longitude of the location of centroid and perceived population.

## Location of nodes in the network

**Table 8. Subset of data showing the location of nodes**

| Node Numbers | Longitude | Latitude |
|---|---|---|
| 5539 | -77.644 | 38.8194 |
| 5541 | -77.6425 | 38.8158 |
| 5544 | -77.6394 | 38.818 |
| 5547 | -77.6376 | 38.8193 |
| 5549 | -77.6361 | 38.8165 |
| 5560 | -77.6251 | 38.9318 |
| 5563 | -77.6229 | 38.969 |

| | | |
|---|---|---|
| 5581 | -77.6095 | 39.14 |
| 5583 | -77.6087 | 38.8032 |
| 5586 | -77.6069 | 38.7974 |
| 5601 | -77.5976 | 38.7998 |
| 5613 | -77.59 | 38.8017 |
| 5614 | -77.5894 | 38.7983 |
| 5615 | -77.5885 | 39.086 |
| 5618 | -77.5845 | 39.1234 |
| 5620 | -77.583 | 39.1182 |
| 5623 | -77.5815 | 39.1206 |
| 5636 | -77.574 | 39.1024 |
| 5648 | -77.5645 | 39.1155 |
| 5653 | -77.5605 | 38.7985 |
| 5660 | -77.5563 | 39.0986 |
| 5661 | -77.5555 | 39.1281 |
| 5674 | -77.5468 | 39.1403 |
| 5678 | -77.5452 | 39.1044 |
| 5682 | -77.5446 | 39.1301 |
| 5686 | -77.542 | 39.0817 |
| 5703 | -77.533 | 38.7374 |
| 5713 | -77.5292 | 38.7385 |
| 5717 | -77.5285 | 38.7478 |
| 5719 | -77.5282 | 38.7308 |
| 5721 | -77.5268 | 38.8183 |
| 5725 | -77.5249 | 38.8013 |
| 5729 | -77.523 | 38.7406 |
| 5732 | -77.522 | 38.7393 |
| 5736 | -77.5211 | 38.8045 |

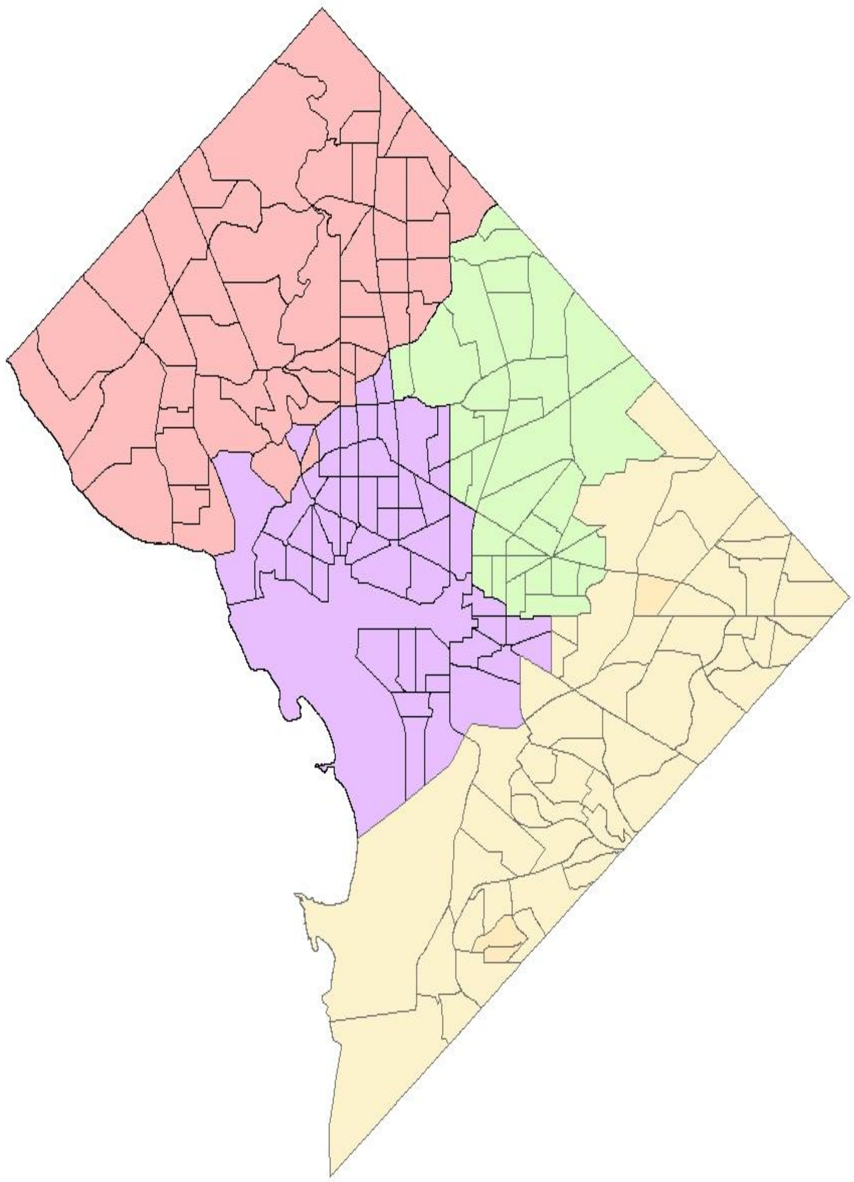


**Figure 28 Map of Washington, DC divided approximately in to four equal parts (with different colors).**

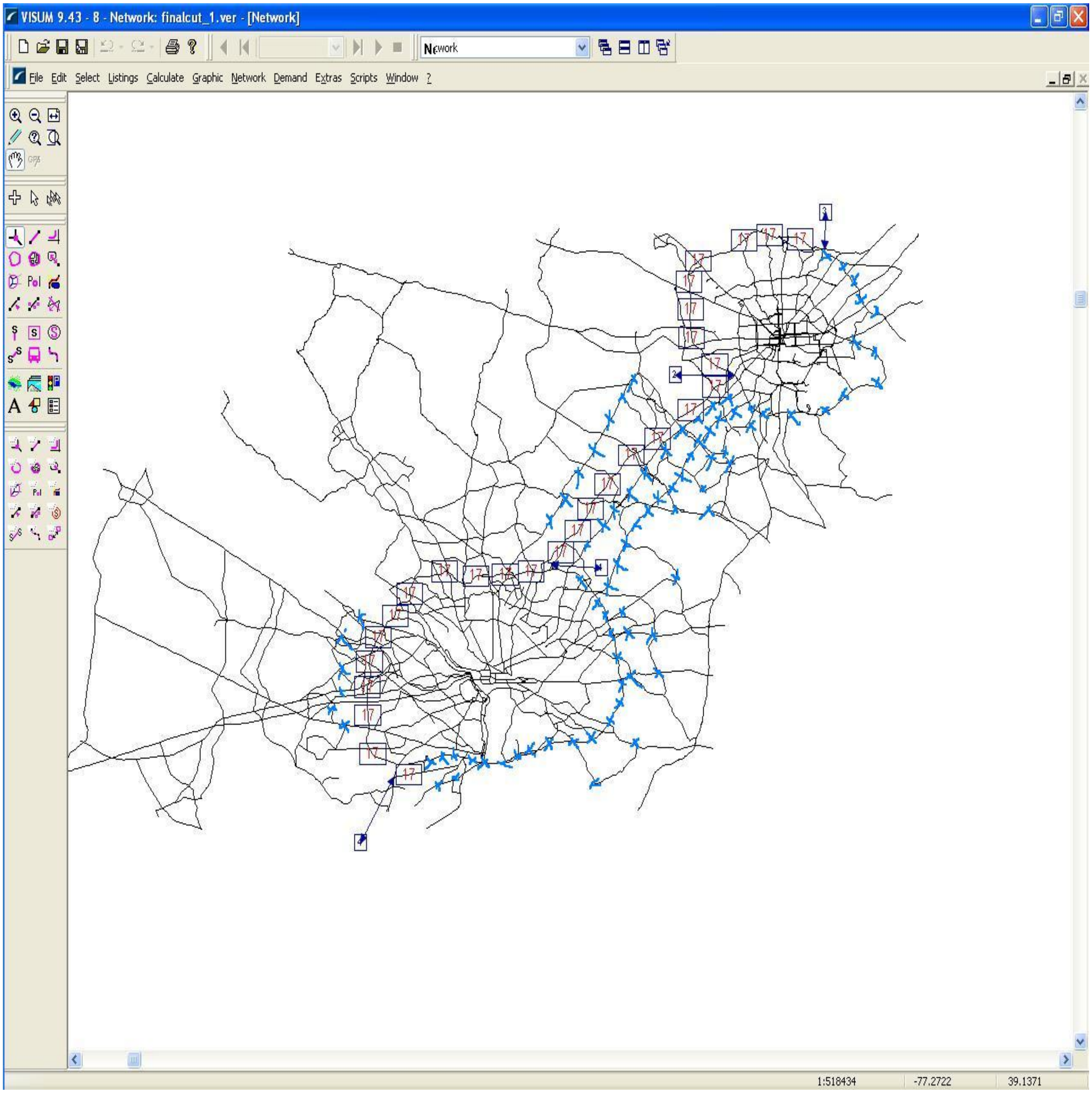



**Figure 29. Figure showing paths taken by hazmat carrier when incidents take place links in the network (indicated blue color X mark)**

# Appendix C Results

## *Results from path prediction methodology*

**Table 9. Parameters required for calculating the probable paths of malicious entity**

| Target | IAD | National Mall |
|---|---|---|
| Minimum Consequence | 2.23E+05 | 2.01E+05 |
| Maximum Consequence | 1.99E+06 | 2.00E+06 |
| Minimum Travel time | 49.336 | 31.9461 |
| Maximum Travel time | 128.2728 | 44.1092 |

**Table 10. Routes obtained from the MATLAB program for various values of λ(subset) towards National Mall**

| The node Value of the routes obtained for a λ | | | | |
|---|---|---|---|---|
| **0.5** | **0.6** | **0.7** | **0.8** | **0.85** |
| 904 | 904 | 904 | 904 | 904 |
| 895 | 895 | 895 | 895 | 895 |
| 1737 | 1737 | 1737 | 1737 | 1737 |
| 850 | 850 | 850 | 850 | 850 |
| 836 | 836 | 836 | 836 | 836 |
| 829 | 829 | 829 | 829 | 829 |
| 819 | 819 | 819 | 819 | 819 |
| 1735 | 1735 | 1735 | 1735 | 1735 |
| 750 | 750 | 750 | 750 | 723 |
| 745 | 745 | 745 | 745 | 711 |
| 742 | 742 | 742 | 742 | 705 |
| 748 | 748 | 748 | 737 | 703 |
| 747 | 747 | 747 | 728 | 699 |
| 737 | 737 | 737 | 723 | 693 |
| 728 | 728 | 728 | 711 | 688 |
| 723 | 723 | 723 | 705 | 1772 |
| 711 | 711 | 711 | 703 | 682 |
| 705 | 705 | 705 | 699 | 669 |
| 703 | 703 | 703 | 693 | 681 |
| 699 | 699 | 699 | 688 | 663 |
| 693 | 693 | 693 | 1772 | 672 |
| 683 | 683 | 688 | 681 | 683 |
| 668 | 668 | 1772 | 663 | 668 |
| 655 | 655 | 681 | 672 | 655 |
| 1730 | 1730 | 663 | 683 | 1730 |
| 654 | 654 | 672 | 668 | 631 |
| 637 | 637 | 683 | 655 | 632 |

| | | | | |
|---|---|---|---|---|
| 632 | 632 | 668 | 1730 | 630 |
| 630 | 630 | 655 | 654 | 619 |
| 619 | 619 | 1730 | 637 | 620 |
| 620 | 620 | 654 | 632 | |
| | | 637 | 630 | |
| | | 632 | 619 | |
| | | 630 | 620 | |
| | | 619 | | |
| | | 620 | | |

**Table 11. Routes obtained from the MATLAB program for various values of λ(subset) towards IAD**

| Node Values obtained for values of λ | | | |
|---|---|---|---|
| **0.5** | **0.6** | **0.7** | **0.8** |
| 904 | 904 | 904 | 904 |
| 895 | 1773 | 1773 | 1773 |
| 1737 | 917 | 917 | 917 |
| 850 | 938 | 938 | 938 |
| 836 | 942 | 942 | 942 |
| 829 | 944 | 944 | 944 |
| 819 | 922 | 922 | 922 |
| 1735 | 921 | 921 | 921 |
| 723 | 919 | 919 | 919 |
| 711 | 897 | 897 | 897 |
| 705 | 890 | 890 | 890 |
| 703 | 879 | 879 | 879 |
| 699 | 864 | 864 | 864 |
| 693 | 1737 | 845 | 845 |
| 688 | 850 | 829 | 829 |

| | | | |
|---|---|---|---|
| 1772 | 836 | 819 | 819 |
| 682 | 829 | 1735 | 1735 |
| 669 | 819 | 723 | 723 |
| 656 | 1735 | 711 | 1734 |
| 1732 | 723 | 705 | 693 |
| 621 | 711 | 703 | 683 |
| 617 | 705 | 699 | 672 |
| 611 | 703 | 693 | 663 |
| 610 | 699 | 688 | 681 |
| 603 | 693 | 1772 | 1772 |
| 601 | 688 | 682 | 682 |
| 594 | 1772 | 669 | 669 |
| 508 | 682 | 656 | 656 |
| 483 | 669 | 1732 | 1732 |
| 491 | 656 | 621 | 621 |
| 489 | 1732 | 617 | 623 |
| 452 | 621 | 623 | 624 |
| 444 | 617 | 1731 | 622 |
| 1720 | 611 | 661 | 615 |
| 1722 | 610 | 663 | 597 |
| 361 | 603 | 672 | 608 |
| 354 | 601 | 683 | 607 |
| 350 | 594 | 668 | 620 |
| 341 | 508 | 655 | 618 |
| 301 | 483 | 1730 | 605 |
| 300 | 491 | 631 | 556 |
| 299 | 489 | 632 | 537 |
| 297 | 452 | 637 | 531 |
| 295 | 444 | 647 | 526 |
| 292 | 1720 | 630 | 524 |

| | | | |
|---|---|---|---|
| 282 | 1722 | 619 | 538 |
| 1724 | 361 | 620 | 590 |
| 1757 | 354 | 618 | 596 |
| 245 | 350 | 605 | 592 |
| 242 | 341 | 556 | 595 |
| 219 | 301 | 537 | 572 |
| 1756 | 300 | 531 | 574 |
| 181 | 299 | 529 | 573 |
| 95 | 297 | 527 | 565 |
| 91 | 295 | 522 | 549 |
| 83 | 292 | 514 | 544 |
| 81 | 282 | 498 | 542 |
| 69 | 1724 | 495 | 533 |
| 66 | 1757 | 513 | 525 |
| | 245 | 523 | 510 |
| | 242 | 526 | 509 |
| | 219 | 524 | 504 |
| | 1756 | 549 | 501 |
| | 181 | 544 | 472 |
| | 95 | 542 | 467 |
| | 91 | 533 | 461 |
| | 83 | 525 | 443 |
| | 81 | 510 | 426 |
| | 69 | 509 | 422 |
| | 66 | 504 | 419 |
| | | 501 | 417 |
| | | 472 | 410 |
| | | 467 | 406 |
| | | 461 | 359 |
| | | 443 | 313 |

| | | | |
|---|---|---|---|
| | | 426 | 291 |
| | | 422 | 1741 |
| | | 419 | 253 |
| | | 416 | 247 |
| | | 320 | 242 |
| | | 288 | 219 |
| | | 291 | 1756 |
| | | 1741 | 181 |
| | | 253 | 95 |
| | | 247 | 91 |
| | | 242 | 83 |
| | | 219 | 81 |
| | | 1756 | 69 |
| | | 181 | 66 |
| | | 95 | |
| | | 91 | |
| | | 83 | |
| | | 81 | |
| | | 69 | |
| | | 66 | |

s

## *Results from the trained neural network*

**Table 12. Validation data used for testing PNN using various values of SPREAD (subset)**

| Node Number | Diversion angle | Vector Distance | Threat level(actual) | spread = 0.1 | spread = 0.2 | Spread =0.3 | Spread = 0.4 | Spread =0.5 |
|---|---|---|---|---|---|---|---|---|
| 6881 | 37.8145 | 1.1848 | 3 | 3 | 3 | 3 | 3 | 3 |
| 9218 | 32.8599 | 0.1408 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8559 | 16.2602 | 0.4547 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7661 | 16.2602 | 2.1998 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8870 | 67.6663 | 0.0856 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9612 | 11.4783 | 4.7055 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8832 | 34.9152 | 0.2236 | 3 | 3 | 3 | 3 | 3 | 3 |
| 6540 | 14.0699 | 3.0251 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9625 | 14.0699 | 1.8398 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8613 | 60 | 0.2028 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9059 | 54.5495 | 0.1707 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8426 | 21.5652 | 0.5289 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8521 | 19.9484 | 0.1192 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8399 | 19.9484 | 0.5865 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8967 | 43.1136 | 0.2163 | 2 | 2 | 2 | 2 | 2 | 2 |
| 5686 | 28.3576 | 2.1164 | 3 | 3 | 3 | 3 | 3 | 3 |
| 7609 | 14.0699 | 1.4337 | 1 | 1 | 1 | 1 | 1 | 1 |