REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
10/6/2015	Software user menu	1-June 12 - 30-September 2015

4. TITLE AND SUBTITLE
MECH: Algorithms and Tools for Automated Assessment of Potential Attack Locations

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-12-1-0531

5c. PROGRAM ELEMENT NUMBER
12PR07044-00

6. AUTHOR(S)
S. George, X. Wang, J. Lin, B. Qu, J.C. Liu

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Computer Science and Engineering
Texas A&M University
H. R. Bright Building, Room 502B
3112 TAMU
College Station, TX 77843-3112

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
875 North Randolph Street
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; distribution if Unlimited

13. SUPPLEMENTARY NOTES

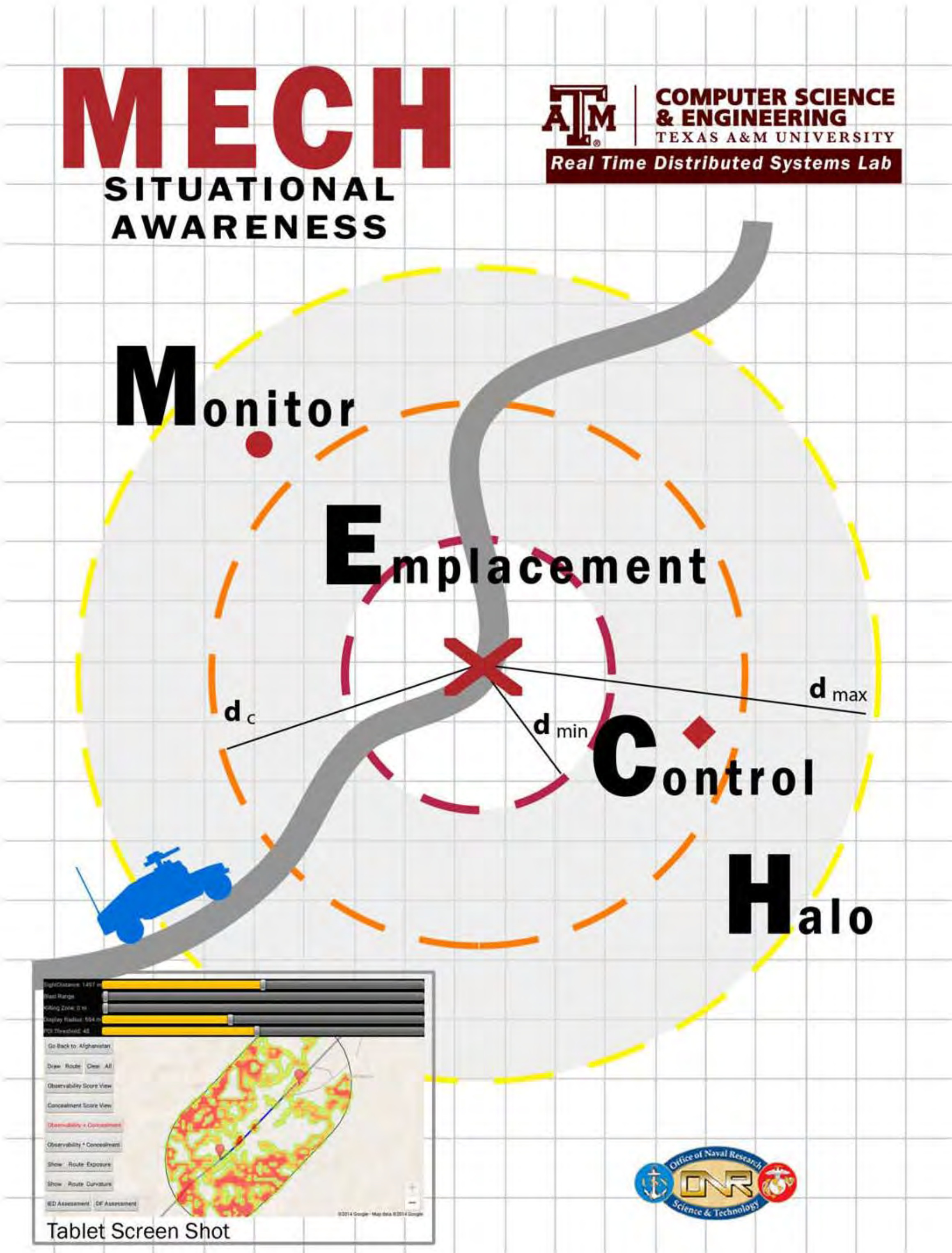
14. ABSTRACT
This report presents the user menu for a prototype of the Monitor, Emplacement, and Control in a Halo (MECH) model for modeling and simulation of behaviors associated with insurgent attacks, and their relationship with geographic locations and temporal windows. The software system is consisted of an Android device to make query to a server, which can make tactical analysis using the MECH model based statistical pattern analysis and simulation tools. The statistical pattern analysis method is cross checked with tactical simulations based on the MECH behavior model. The prototype demonstrates the effectiveness of fusion of statistical pattern analysis, simulation, and human interpretation of military doctrines within the context of the two modeling approaches. It shows the feasibility of self-guided situational analysis informed by MECH-based situational awareness analytics.

15. SUBJECT TERMS
Situational awareness, Geomorphometric Features, Visibility-related Features, Statistics, Asymmetric Attack

16. SECURITY CLASSIFICATION OF: unclassified			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** J.C. Liu
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	none	24	**19b. TELEPHONE NUMBER** *(include area code)* 979-845-8739

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

MECH Situational Awareness System



User Guide

October 2, 2015

Point of Contact

Dr. Jyh-Charn (Steve) Liu

Department of Computer Science and Engineering

Texas A&M University

liu@cse.tamu.edu

979-845-8739

1 Overview

"**M**onitoring of target movements, **E**mplacement of a device, and **C**ontrol of the device in a **H**alo," or **MECH,** is an analytical abstraction to model the locational relationships between a target and its attackers (around the route) in asymmetric conflicts[1]. A conceptual diagram of the MECH model is illustrated on the cover of this document. The MECH V0.1 software system ranks and predicts the relative likelihood of locations that may be used for asymmetric conflicts based on probabilistic reasoning of attackers' behaviors, utility of inter-visibility for actions and protection, and machine learning (ML) based pattern classification. The behavior modeling subsystem computes and compares cumulative utility of locations within an area of interest (ROI) based on the combination of behavior constraints defined in a Halo shape, visibility models, and risk aversion criteria. The user can also customize an ensemble of ML classifiers, preprocessing of features of historical data, as well as training algorithms. The two subsystems make independent predictions for cross assessments of the tactical situations.

The system architecture of MECH v0.1 is shown in Figure 1. The Android *MECH-App* shown on the left side of the figure is for end users to request tactical risk assessments for a ROI, which can be a *point*, a *route* between two points, or a rectangle defined by 2 diagonal points. The *MECH web portal server* (MECH-WPS) shown on the right side of the figure is responsible for performing studies created by the users. The MECH *classifier training subsystem* (MECH-CTS) supports import of raw data and feature generation, configuration of classifier training experiments and performance reports, and creation of ensembles. As needed, MECH-CTS can run on a different computer to share the computing workload of the MECH-WPS.

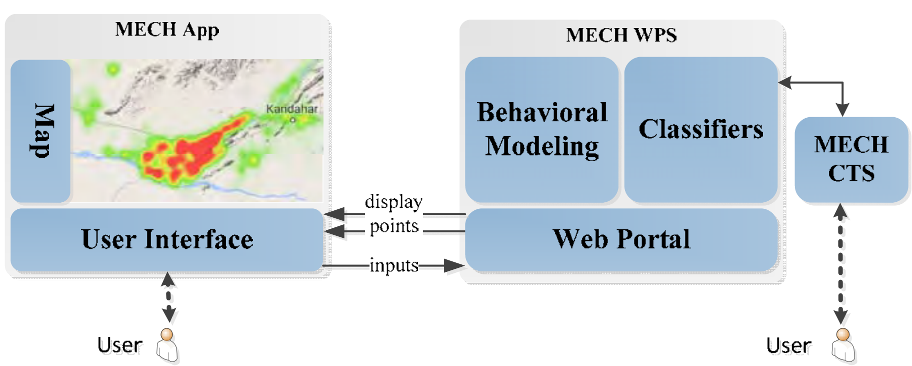


Figure 1. Organization of Primary Functional Components for MECH v1.0

---

[1] MECH offers important insights on utility of locations for certain types of asymmetric conflicts, but it does not know when or where the next attack will take place.

2 MECH-App

MECH-App runs on a touch screen based Android device for end users to access the analytics tool. MECH is designed for users to gain situational awareness of tactical risks in a region of interest R, and the Proximity (P) of R. Each point of R is associated with a Halo annulus, whose parameters are defined with respect to the tactical behaviors. P is created by taking the union of the Halo annuluses of each point in R. Any point on R can be an E point (for device emplacement), and any point *p* in P can be an M/C (monitor of the target and control of the device) location. MECH-App accesses the Google Map service for map operations, and the MECH-WPS for the tactical analytics, respectively. MECH v 0.1 supports five major studies:

(CA1) Basic Measurements: Several measurements derived from Line of Sight (LOS) on P and R locations within a study area.

(CA2) Behavioral Modeling: Probabilistic reasoning of locations for M, E, and C activities. Tactical parameters can be defined based on a risk averse/seeking Halo model.

(CA3) Machine Learning (ML) Classifier: Classification of R points by using one of the trained ML classifiers stored on the MECH-WPS.

(CA4) Past Events: Past events in the displayed map area of the APP.

The CA1-CA4 tabs are organized as a group in the Android *navigation drawer* (Figure 2.1 (a)), which is initially hidden when MECH-App is first launched. It can be brought to the screen by swiping the left up edge of the main display. CA4 is permanently displayed on the *action bar* (Figure 2.1 (b)) on the top of the screen so that historical incident sites can be superimposed on the currently displayed map.

(a) (b)

Figure 2.1. The main menu for MECH v0.1. (a) The navigation drawer, and (b) the action bar.

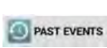
Main modules in the MECH-App are summarized as follows:

Modules	Description	Affected Studies

M0	(1) Reset the map. (2) Homing and acquiring of the base map from the Google map server.	CA1-CA3, CA4
M1	Set the study area, either as a point or a route between two points.	CA1-CA3
M2	Set up the Halo model parameters (i.e., sight distance, fire Range, and blast Range, and the distance curve modes, etc.) for M3. This module does not consider the risk averse/seeking tactics.	CA1
M3	Select a Line-of-Sight (LOS) based measurement and the Halo model. User defined Halo parameters are used if M2 is run prior to M3. Otherwise, default Halo parameters will be used for the study if M2 was not run.	CA1
M4	Set up the risk averse/seeking Halo parameters in attackers' tactical parameters.	CA2
M5	Select one of the assessment functions to be sent to the MECH-WPS for study, and pass the study results to M8 for display.	CA2
M6	Overlay display of the historical sites.	CA4
M7	Adjust the display range of a heat-map display, and the cut-off threshold of points to be displayed.	CA1, CA2
M8	Draw the heat map for a study result.	CA1, CA2, CA3
M9	Select a trained classifier from the MECH-WPS for study of a route.	Step 1 of CA3.
M10	Contact the MECH-WPS for classification of the chosen route points. The prediction results are sent to M8 for display.	Step 2 of CA3

Action Bar

The appearance of the action bar on the top does not change in most scenarios.

Control Button	Actions
	It selects a route "*R*"for study. A route is selected by tap-and-hold one point on the map, and then tap-and-hold another point. The APP will highlight the route that has been selected.
	It brings the map to the Afghanistan region. (As of writing this document, the APP only works for the Afghanistan region.)
	It erases all overlay points on map and resets all parameters to their default values.
	Historical incident sites in the display area can be displayed anytime by tapping

Navigation Drawer

The navigation drawer will expand to show operations available in (CA1-CA4) by tapping its button.

CA1: Basic Measurement

CA1 produces LOS based measurements without considering behavior aspects. The main menu is shown in Figure 2.2. The *route curvature* and *route exposure* produce R points as outputs, and all other analyses produce P points as outputs. While all P and R points are analyzed, only points that have the highest values are returned to the App. (The same rule applies to all studies.)

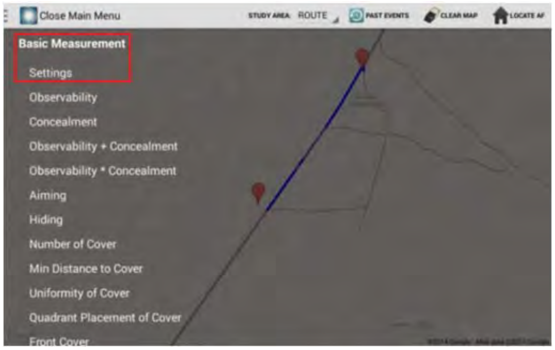


Figure 2.2. Main Menu

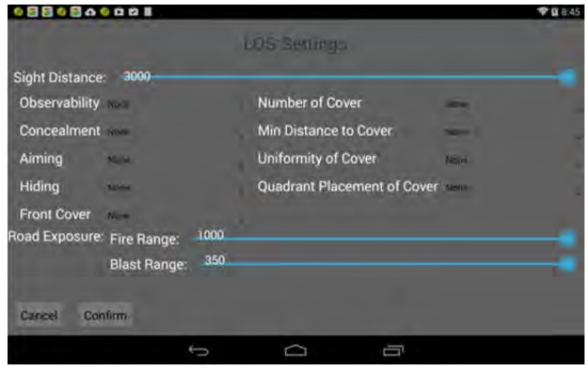
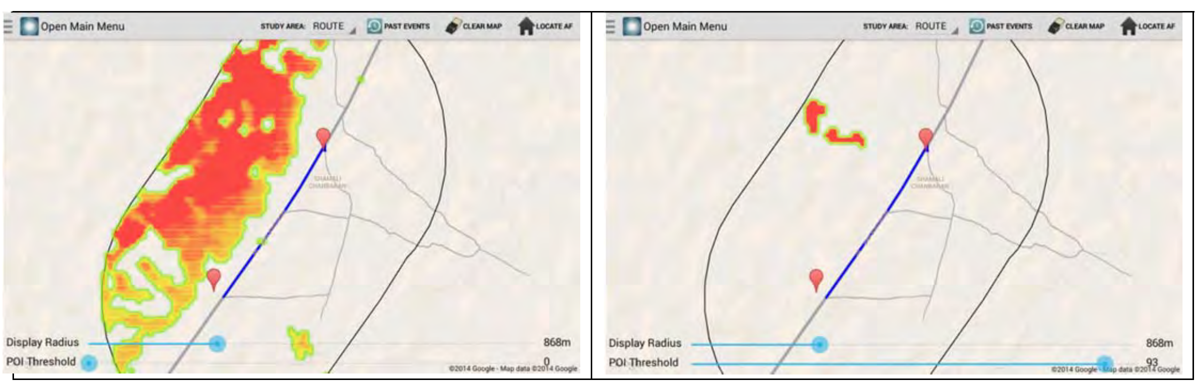


Figure 2.3. Parameters setting

Workflow: M1 → M2 → M3 → M7 → M8

The user selects a route (M1), sets parameters (M2, in Figure 2.3), and then chooses the type of study (M3, in Figure 2.2). The Halo model can be accessed by tapping the "Halo Parameter Setting" button, and the user can adjust tactical parameters listed below using the sliding bars. A set of default values will be used for a study if the user skips this step. After the result is returned, the user can manually adjust the display range (M7, in Figure 2.4), as well as the points of interest (POI) (M7) after the results are displayed. The study can end here, or the displayed results on P points can be used for behavioral modeling in CA2.



(a) (b)

Figure 2.4. Result after adjusting display radius and POI threshold. (a) Before the POI is adjusted, and (b) after the POI is adjusted.

Halo Parameter Setting

The following notations represent the utility of the physical measurement based on user defined utility curves.

2.1 **Observability U_V(Observability):** The cumulative LOS values between a *p* point to all R points. The score of a point *p* is proportional to the number of R points that it can see. The computation is done for all *p* points within the Halo radius, along each point on R, but only points with the highest scores are returned to the MECH-App for display.

2.2 **Concealment U_C (Concealment):** The number of points around *p* with no visibility to R. This tab produces *p* points with highest U_C scores.

2.3 **Concealed observabilityU_{CV}:** A combined measure of $U_V$and U_C. U_{CV} can be "additive": $U_C + U_V$ (Observability + Concealment), or "multiplicative": $U_C \times U_V$ (Observability * Concealment), as specified by the user.

2.4 **Aiming timing U_T(Aiming):** The ability for a *p* point to see a target's continuous movements for 150 meters before reaching a particular location on R.

2.5 **Hiding U_H(Hiding):** The sum of U_{nc},U_{mc}, U_{uc}, U_{pc} and $U_{fc}$for a *p* point. The radius of the usable hiding area, or escape path from the *p* point is fixed at 350 meters.

2.6 **Number of Covers (U_{nc})(Number of Cover):** The number of covers around a *p* point.

2.7 **Shortest Distance to a Cover (U_{mc})(Min Distance to Cover):** The shortest distance for a *p* point to reach a cover. The score U_{mc} is inversely proportional to the distance.

2.8 **Uniformity of Covers (U_{uc})(Uniformity of Cover):** The standard deviation of distances between a *p* point and its covers.

2.9 **Quadrant Placement of Cover (U_{pc}) (Quadrant Placement of Cover):** The number of (the 4) quadrants around *p* that have covers.

2.10 **Front Cover (U_{fc}) (Front Cover):** The number of invisible points around a *p* point, along the side facing R.

2.11 **Route Exposure (Route Exposure):** For each point *r* on R, the number of *p* points within the study area defined on R that can see *r*. That is, which points on R have the highest visibility from P.

2.12 **Route Curvature (Route Curvature):** The curvature of road segments on R. Or, straightness of the road segments.

CA2: Behavioral Modeling (Strategic(R level) Assessment)

Three types of studies are supported based on the risk averse/seeking behavioral model defined in the Halo model.

Types 1 & 2: E locations for IED& DF Attacks

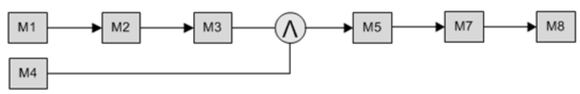
The IED and DF studies follow identical operational steps, but use different control buttons to access different tactical models on the MECH-WPS. They both use a set of P locations produced in CA1 as the potential M/C locations, and then the study produces the E locations with the highest risk scores.

Step 1: Tap the "Halo Parameter (IED/DF based)" button to open a new menu for adjustment of tactical parameters in the Halo model. Default values will be used if this step is not taken.

Step 2: Tap the "Start IED/DF Risk Assessment" button to request the MECH-WPS to run the study.

The "^" symbol in the workflow chart below means that both M4 (setting of Halo parameters) and the chain of M1-M2-M3 need to be run, but the two branches need not be in a specific order. In comparison to the Halo model used in CA1, M4 is a more elaborate version that supports characterization of risk averse/seeking behaviors.

IED&DF Workflow:

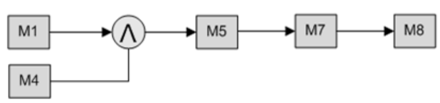


Type 3: R2P Study

This study uses all R points to study their proximity points within the radius of the Halo model. The study produces a list of points that have the highest M/C scores.

Step 1: Tap the "Halo Parameter Setting (R2P)" to set up the Halo parameters.
Step 2: Tap the "Start R2P Assessment" to produce the heat maps of P points with the highest scores with respect to R.

R2P Workflow:



Halo MECH Parameter Setting (Figure 2.5)

A screen shot of the Halo model setup page is shown in Figure 2.5 (Note, the full length menu is cut and then fit into the horizontal space of the document.) The display is organized into 5 views. In view 2, the center of the Halo is the victim location, and the M/C location can be anywhere within the sight distance (blue circle) but outside of the blast range (red circle). View 1 is dedicated to setting the geometric parameters of the Halo model, which are listed in the following table. They are adjusted by swiping the sliding bars.

Parameters	Description

Sight Range (blue circle)	The outer radius of the Halo based on the human sight range.
Blast Range (red circle)	The blast range of an improvised explosive device (IED) (The inner radius of Halo annulus). Attackers do not stay in this range.
Aiming Range (light blue line)	A range for *M*/*C* points to see the target continuously move along the route to the attack engagement location *E*.
Device Triggering Range (yellow line)	IED attacks: the maximum range to trigger an IED device. DF attacks: the shooting range of the attackers.
Return Fire Range (black line)	The range of return fire by the victims from *E* to the *M*/*C* location.
Retreat Distance to a cover (green circle)	The reachable distance to a cover. The choice can be the nearest or a randomly chosen one based on the behavior model.

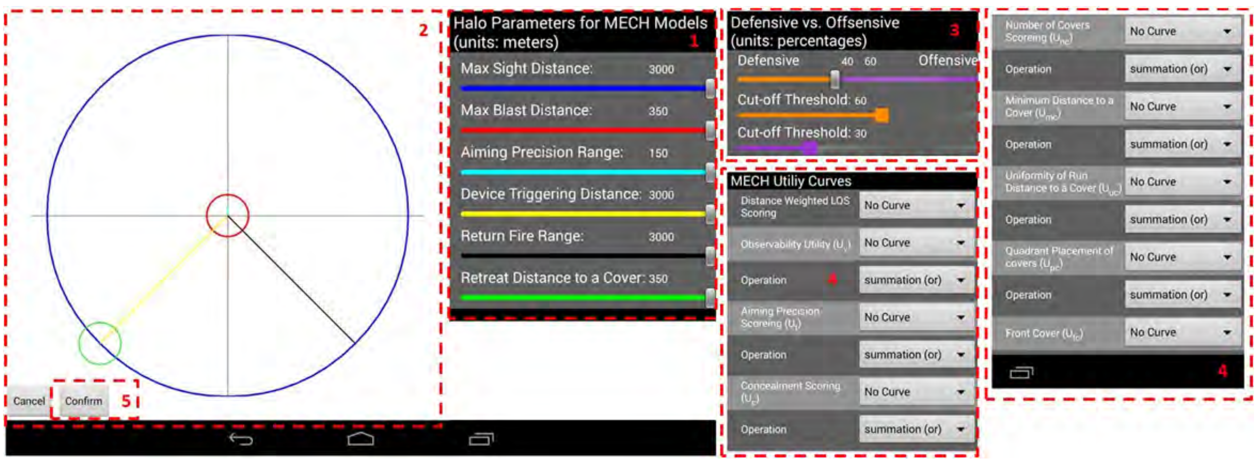
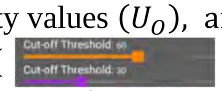


Figure 2.5: Halo Parameters for MECH Models

Risk Averse Behaviors in Tactical Planning (Views 3,4 of Figure 2.5)

Attackers go through formal or informal optimization (i.e., prioritization, trade off choices) of their *offense* (risk seeking) and *defense* (risk averse) measures in planning tactical operations. MECH offers a generalized expression to incorporate a few offense and defense measures. Let the symbol "Δ" represent a multiplication "× " or addition "+" operation between two variables. A generalized utility function that represents a combination of the offensive and defensive measures can be expressed as follows:

$$f(U_O, U_D) = s_O \cdot s_D[\omega_O(U_O)\Delta\, \omega_D(U_D)].$$

s_O and s_D represent two threshold switches of the offense and defense utilities. They overwrite the value of $f(U_O, U_D)$ to zero if either of the utility values (U_O), and (U_D) is smaller than the thresholds set by the user using the sliding bars (). The two coefficients $\omega_O$and ω_D define the relative weights of the offense vs. defense utilities. The ratio between them is adjusted by the sliding bar at top of view 3. By proper adjustment of these parameters, the user can study a wide range of risk averse-seeking behaviors.

Every utility in view 4 is derived using LOS as the building block. The standard definition of the LOS between two points does not consider their distance, but this may not be the case in the real world environment. In MECH, the user can use the distance to weight the tactical value of LOS using one of the built-in *distance curve* functions (No curve, step function, linear decrease, and rapid decrease). They are organized into a pull down menu for the user to pick the preferred mode from the four different distance curve functions:

(1) No curve: The distance is irrelevant in the tactical study.
(2) Step function: The tactical value of LOS becomes zero when the distance is greater than 2000 meters.
(3) Linear decrease: The tactical value of the LOS decreases at the rate of 1/distance.
(4) Rapid: The tactical value of the LOS decreases rapidly at the rate of ${}^{1}/_{\exp(d)}$, where *exp* is the exponential function and *d* the distance.

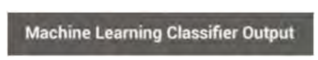
The remaining buttons in view 4 are meant to describe details of the utilities U_O and U_D of the offensive and defensive measures:

$U_O = \ (U_V \Delta\, U_T),$

where U_V and U_T represent observability and aiming for the offensive measure, respectively.

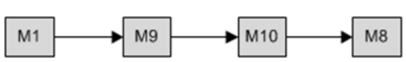
$U_D = \omega_D \ \left[U_C\, \Delta \left(U_{nc} \Delta\, U_{mc} \Delta\, U_{uc} \Delta\, U_{pc} \Delta\, U_{fc}\right)\right],$

where U_V, U_T, U_C, U_{nc}, U_{mc},U_{uc}, U_{pc}, and U_{fc} are defined in CA1. By adjusting a few sliding bars, the user can easily select which of these factors, together with their weights to be used for a study.

CA3: Machine Learning Classifier ()

MECH supports a 2-label ("incident" (Y) vs. "non-incident" (N)) classification of R locations. A common interpretation of the "Y" ("N") output for a location X is that the features associated with X are more similar to that of the incidents (non-incidents) used in the training process. The user can run a study by simply following the three steps of (1) selecting a route, (2) selecting a classifier, and then (3) sending the request. The prediction will be made to all points on R using the chosen classifier, and the result is displayed on R in the heat-map format.

Workflow:



3 MECH-CTS

MECH-CTS is responsible for importing raw data and their transformation into useable forms for all algorithms in MECH. It is also responsible for training and performance evaluation of classifiers. As of writing this report, the whole application runs as one Matlab program by a command line statement:

> "**Matlab –r CTS.m**"

Main User Interface

The main user interface of MECH-CTS is shown in Figure 3.1. The navigation bar on the top (view 1) allows the user to run one of the three functions, Feature Generation (Feature Generation), Classifier Training (Classifier Training), and Classifiers Ensemble (Classifier Ensemble). Details of a selected function are displayed below the navigation bar. For instance, the snapshot illustrated in Figure 3.1 is the display after (Feature Generation) is clicked.

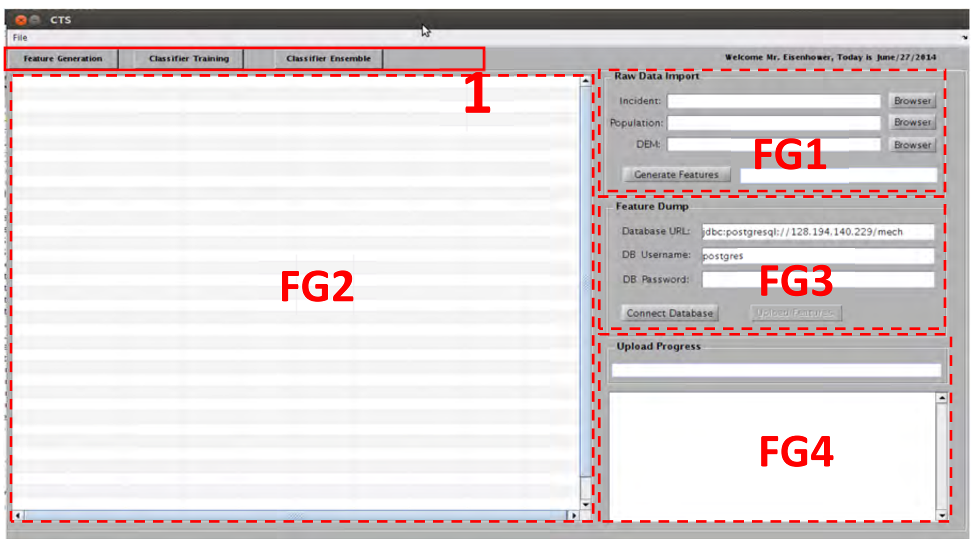


Figure 3.1. The main user interface for MECH-CTS

Feature Generation (Feature Generation)

This module is responsible for (1) importing raw data (historical data file, road points file, DEM, population file), (2) generation of features and uploading them to the database. The display space for this module is organized into FG1-FG4.

FG1. The user uses the (Raw Data Import) button to select files that contain incident sites, population and DEM data.

FG2. After clicking the (Generate Features) button, the feature generating process will start to run, while the features being generated are streamed on view FG1.

FG3. The user can specify the database to store the features.

FG4. Operation log and the progress bar of the process.

After they are loaded into the database, the input data and their transformed features can be used by all applications, including the MECH-CTS.

Classifier Training (Classifier Training)

This module is responsible for feature preprocessing (feature normalization and feature reduction), running the training algorithms of classifiers, evaluation of their performance on different data sets, and upload of the trained classifiers to the MECH-WPS. Once uploaded to MECH-WPS, trained classifiers can be further grouped by the MECH-CTS to create *ensembles*. The end user of MECH-APP can select an individual classifier or an ensemble to perform the classification of a study area. The main GUI for the classifier training is shown in Figure 3.2, where the display area is organized into views TC1-TC5.

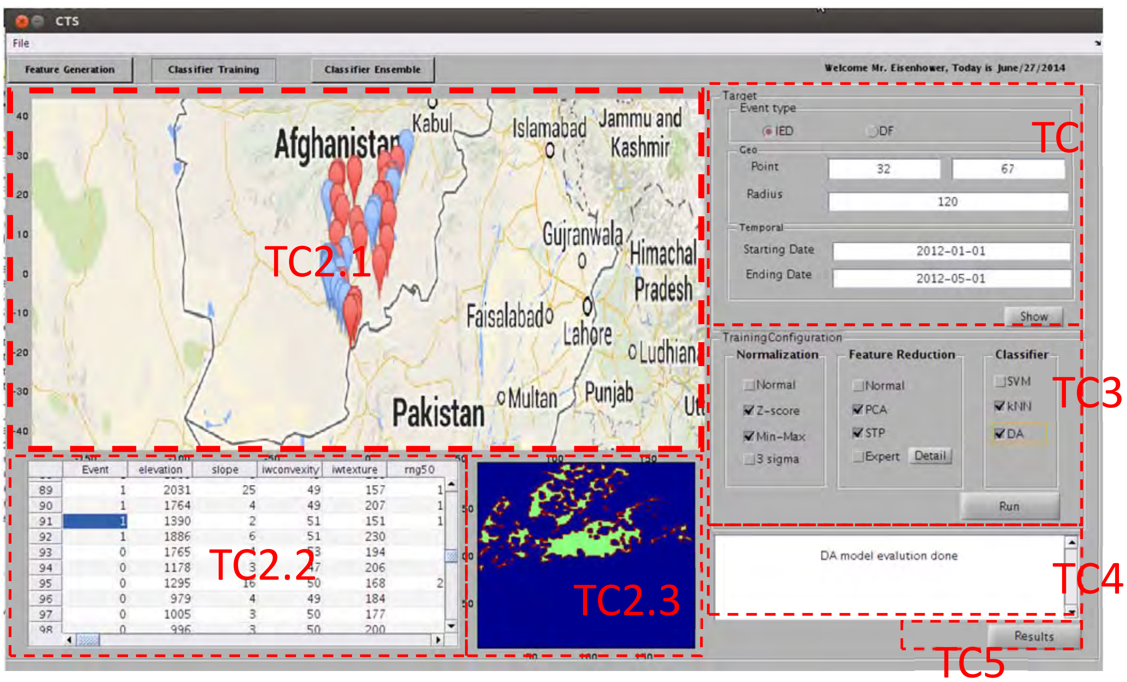


Figure 3.2. Classifier training interface

TC1, TC2.1-TC2.3: Data Selection and Viewing

TC1. It specifies the event type, focal point of the study location and its radius, and the time period. MECH-CTS then automatically generates the training and testing data based on the specification. By clicking the (Show) button, the training data will be shown in TC2.1-TC2.3:

TC2.1. It displays the location of non-event points (blue) and event points (red) as an overlay on the Googlemap.

TC2.2. Each row represents one location, and its associated features.
TC2.3. It displays the viewshed of the point marked in TC2.2.

When the user specifies the time period and area to be used in a training study, all incident locations that satisfy the temporal-spatial constraint will be retrieved as the incident data. Then, the same number of non-incident data will be randomly generated. Combined together, 2/3 of the combined data points are used for training of the classifiers, and the remaining 1/3 for evaluation of the performance of the trained classifiers. To reflect the time sequence of events faithfully, the times of all the data points used in the training phase are ahead of those used in the testing phase.

TC3: Classifier Training Configuration

The classifier training process is configured based on the combinations of the *normalization* methods, *feature reduction* methods, and the *machine learning* methods. Each particular combination produces one classifier at end of the training process. The user can upload one or more classifiers (each of which represents one of the combinations) to the MECH-WPS for online use. Except for the "Expert" option, which requires more actions by clicking the ( ) b Detail ll other options can be chosen by simple clicking their checkboxes. The training process starts by clicking the (Run) button.

Feature reduction

The user can choose one or more feature reduction methods marked in TC3 to manage the 77 features for training.

- The (Normal) mode uses the full set of features (with zero reduction).
- The (PCA) and (STP) modes are based on mathematical transforms.
- In the (Expert) mode, the user manually chooses the features to be used in the training.

To enter the expert mode feature selection, the user clicks the (Detail) button next to the (Expert) checkbox to see a popup user interface for management of groups of features. Figure 3.3 illustrates a snapshot of feature group 21 (marked in SFG2) and its list of features (marked in SFG3) when the user clicks the (Select Groups) button (in SFG1) in the expert mode. By clicking the plaintext title of a group, features of one group can be displayed in SFG3. All feature groups with marked "select" checkboxes will be used in classifier training.

When the user clicks the (Feature Group Clean Up) button (marked in SFG4), a new display (Figure 3.5) will pop up with the list of feature groups already uploaded to the database. The user can remove any obsolete feature group as needed.

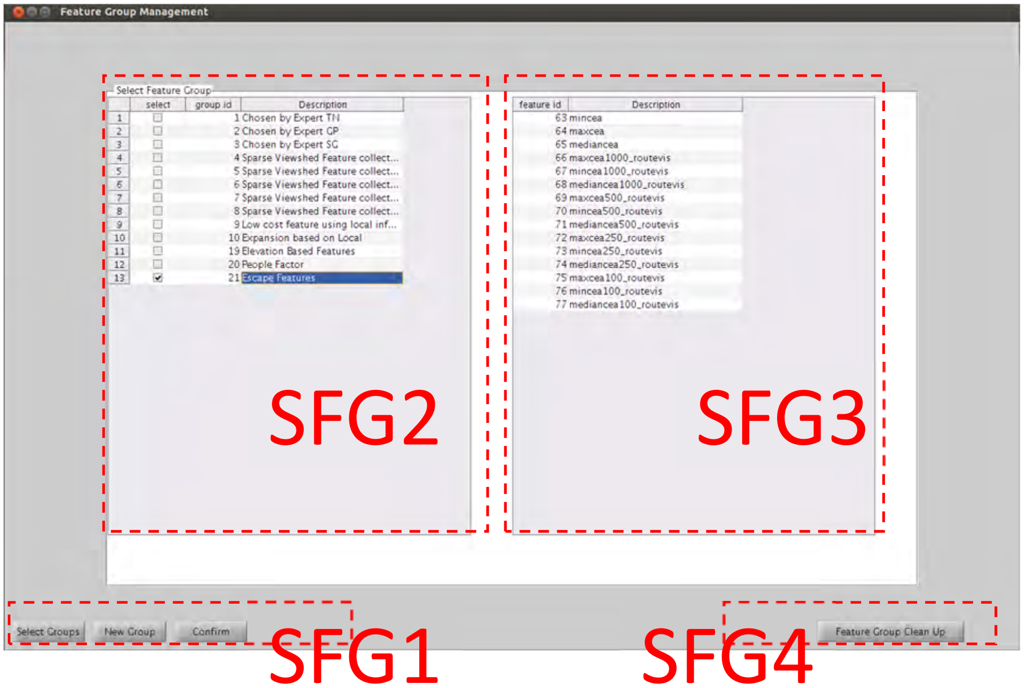


Figure 3.3. Feature group selection in the expert mode.

The user can create a new group by clicking the (Add Group) button in SFG1 to switch to a new display shown in Figure 3.4, whose display area is organized into four areas NG1-NG4. NG1 gives the full list of 77 features. The user can view the *discriminant ability* of a feature (to separate the two Y/N classes in this feature space) in NG2 by clicking the feature's plaintext label displayed in NG1. The user can include an arbitrary number of features into the training process by clicking their checkboxes. NG3 is used to enter brief remarks (rationale, why certain features are being used or removed, etc.) about this feature group. By clicking the (Add The Group) button, the newly configure feature group is added to the database for future use.

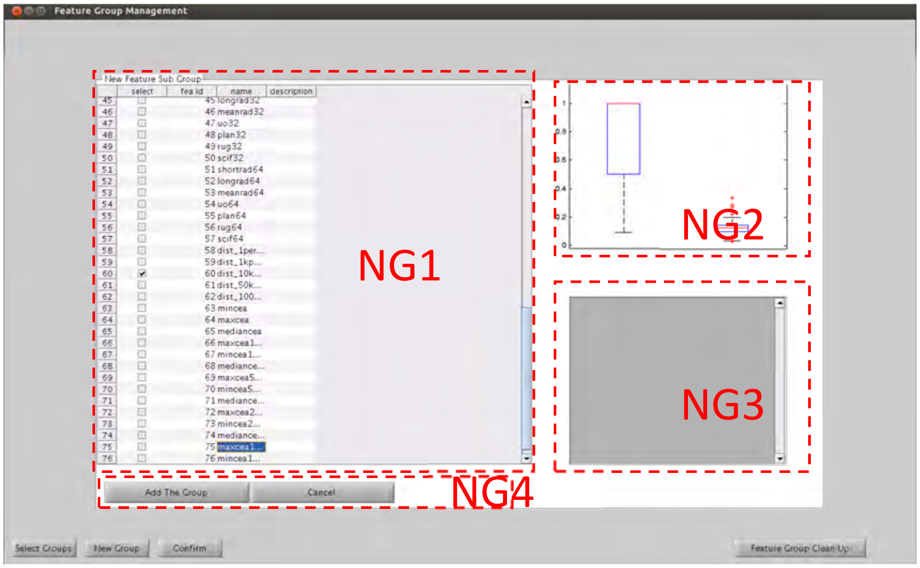


Figure 3.4. Add a feature group and its features.

After the creation and selection process is completed, the user should have a list of feature groups marked, and by clicking the (Confirm) button in SFG1, the selected feature groups will be passed to the training modules. The display window then returns to the classifier training interface shown in Figure 3.2.

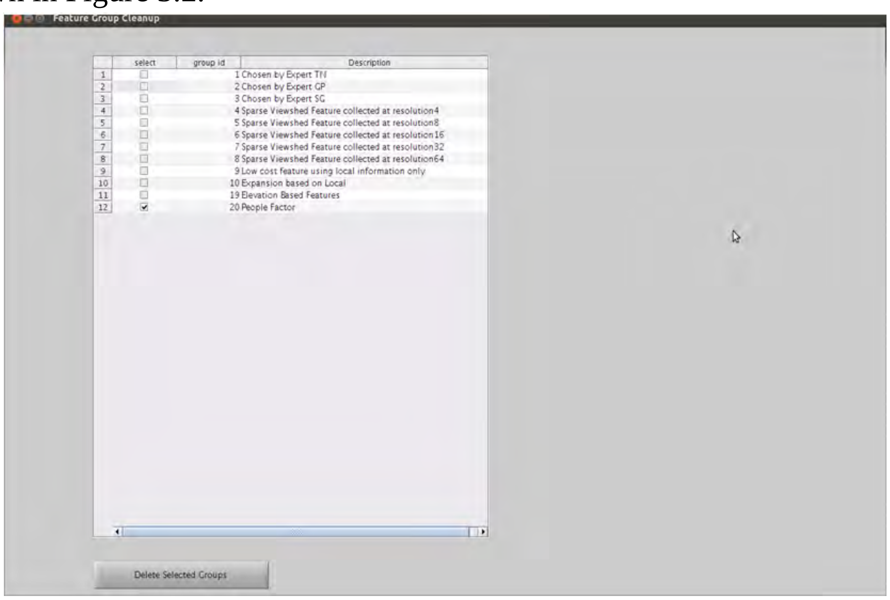


Figure 3.5. Delete feature groups

TC4: Display of the Operation Log

TC5: Display of the Performance Results

By clicking the (Results) button, the display will switch to the performance result reporting page in Figure 3.6. Each combination of the training configuration and its classification performance is shown as one row in view R1. The user can reserve one or more of the displayed configurations to be uploaded to the MECH-WPS by clicking their "reserve" checkboxes. By clicking the (Upload) button, the reserved classifiers will be uploaded to the MECH-WPS, but the configuration still stays in the MECH-CTS. The user has the option to run more experiments by clicking (More Experiments) button, and the display switches to the training main menu (Figure 3.2). A unique internal ID will be assigned to each uploaded classifier, together with a concise descriptor of the experiment attributes, which will be displayed when the end user requests to view available classifiers from their MECH-App.

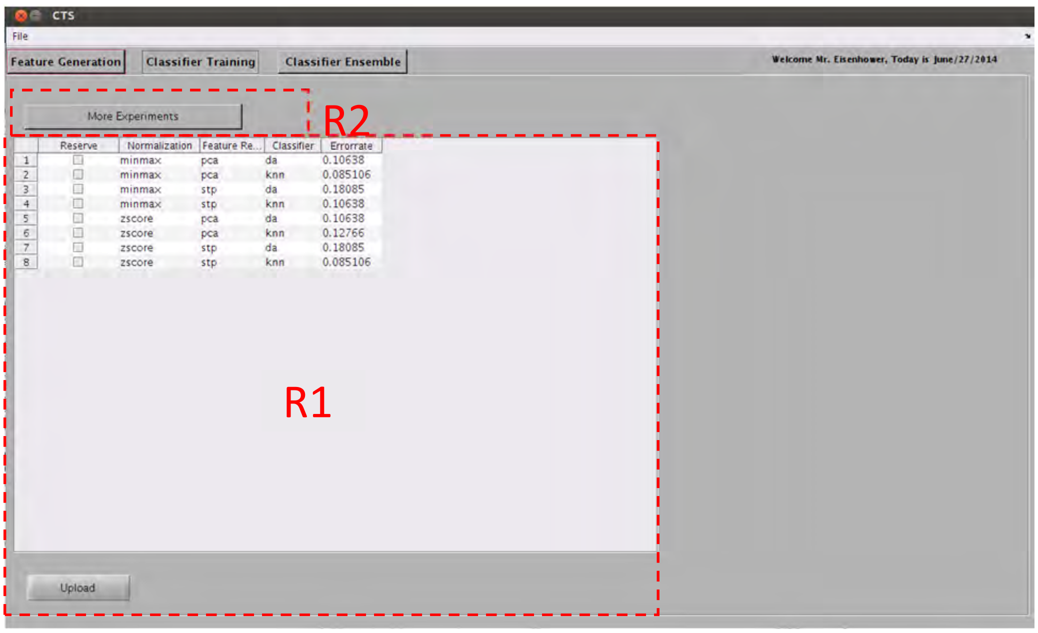


Figure 3.6. Performance reporting and saving of classifiers.

Classifier Ensemble (Classifier Ensemble)

This module is responsible for the creation of ensembles of trained classifiers, which have been uploaded to the MECH-WPS. The user interface is illustrated in Figure 3.7 where the display is organized into three views CE1-CE3.

CE1. Selection of Study Location

This view is for the selection of a particular location of interest to create an ensemble from a list of available classifiers, each of which has been trained around a location. The list of classifiers are displayed in the order of their distances to the location of the current study. All checked classifiers will be used in the ensemble.

CE2. Selection of the ensemble architecture

a. Voter based ensemble: Classifiers' outputs are fed to a voter, which uses a

i. (Majority rule): the ensemble output is the majority of the Y/N outcomes of the classifiers.

ii. (Any rule): the ensemble output is “Y” if any of the classifiers produces a “Y”.

b. Stacking ensemble: The ensemble is modeled as yet another classifier, whose inputs are the outputs of selected classifiers. The ensemble in this option goes through the training process similar to earlier ones to optimize its parameters.

CE3. Performance Validation

By clicking the (Run Eval) button, the performance of the selected ensemble architectures is tested using a newly generated data set around the study area to assess the performance of the ensemble. By clicking the (Save Selected) button, marked ensembles are uploaded into the MECH-WPS for on-line prediction.

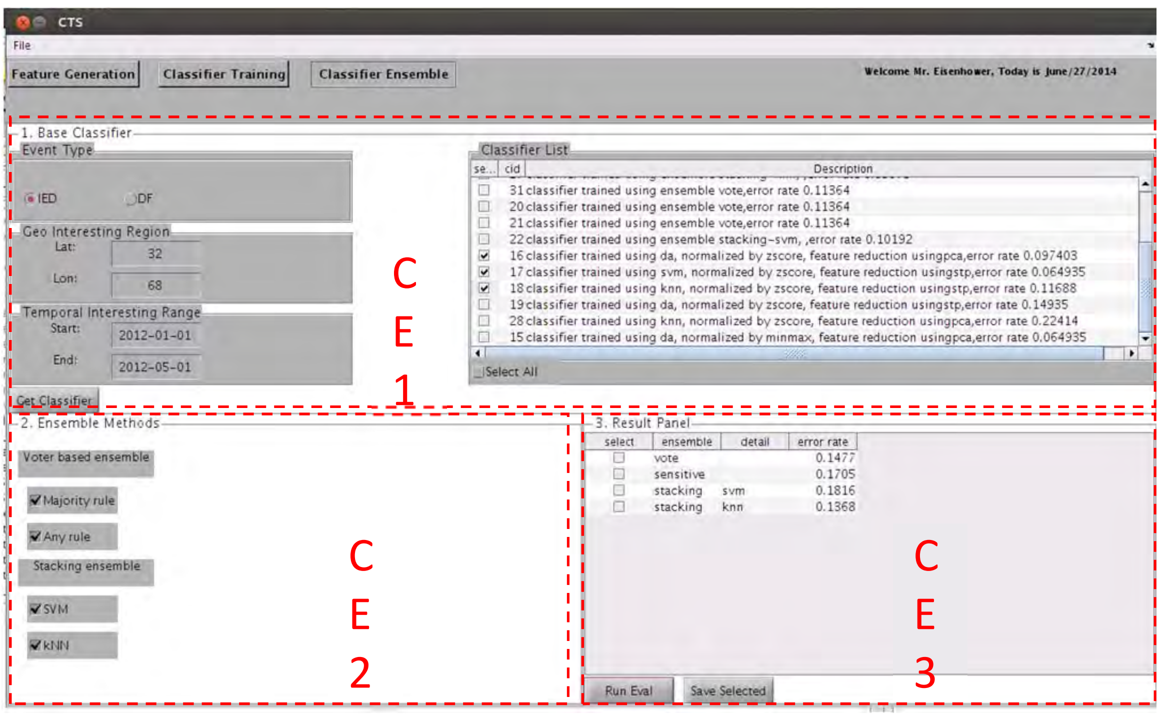


Figure 3.7. Classifiers ensemble interface

4 MECH-WPS

MECH-WPS provides a web portal for end users to access the services provided by the computing engines. Installation of the MECH-WPS package is the only operation most relevant to the user.

Appendix 1. Features Used in the MECH Classifiers

Emplacement Related Features	
Type	Descriptors and Descriptions (unit: 1 pixel = 33.4 meters)
Elevation	Elevation
	The height above or below sea level.
Slope	Slope
	The absolute value of the change rate in elevation along steepest path.
Shape	IW_convexity
	The surface *curvature* of a circle area (radius =10 pixels). (Smaller values imply smoother areas.)
Shape	IW_texture
	The number of pits divided by the number of pits and peaks in a circle area (radius =10 pixels, or 334 m).
Visibility	RtVisMin_100, RtVisMed100, RtVisMax100
	Minimal (Min), Medium (Med), and Maximum (Max) of visibility at the distance of 100 meters to the route.
Elevation	Elv_rng50
	The difference between largest and smallest elevations with 50 meters.
Shape	Rough_50
	The standard deviation of elevations with 50 meters at a location.
Shape	Local_op_4, Local_op_8, Local_op_16, Local_op_32, Local_op_64
	Sparse view shed, the view shed along *n* (*n*=4, 8, 16, 32, 64) equally spaced directions. Viewshed: an indicator on flatness of the terrain. Smaller values imply flatter terrain.
Distance	Dist_pop_1, Dist_pop_1k, Dist_pop_10k, Dist_pop_50k, Dist_pop_100k
	The nearest distance to a city/village with the population size of *n*, *n* = 1/ 1k/10k/50k/100k.

Monitor/Control Related Features	
Type	Descriptors and description
Visibility	Visidx100-350, Visidx_350, Visidx_500, Visidx_1000
	The number of visible points within the view shed of a point (e.g., view TC2.3 in Figure 3.2). About the suffixes: 100-350: the study area is a halo annulus with inner and outer radiuses 100 and 350 meters, respectively. 350/500/1000: the study area is a full circle with the radius of 350/500/1000 meters.
Shape	SCID100-350, SCID_350, SCID_500, SCID_1000

	A discrete *shape complexity* index to characterize the evenness of radii along different directions in a (full) viewshed. About the suffixes: same as above.
Elevation	Elv_rng100, Elv_rng350, Elv_rng500, Elv_rng1000
	The difference between largest elevation and smallest elevation with *n* (*n* = 100, 350, 500, 1000) meters.
Shape	Rough_100, Rough_350, Rough_500, Rough_1000
	Same definition as Rough_50, with the range *n* = 100/350/500/1000 meters.
Distance	Short_rad_4, Short_rad_8, Short_rad_16, Short_rad_32, Short_rad_64
	Short_rad_*n*, *n* =4,/8/16/32/64: The shortest distance from the center to an invisible point along the *n* directions.
Distance	Long_rad_4, Long_rad_8, Long_rad_16, Long_rad_32, Long_rad_64
	The longest distance from the center to an invisible point along the *n* (*n* =4,/8/16/32/64) directions.
Distance	Mean_rad_4, Mean_rad_8, Mean_rad_16, Mean_rad_32, Mean_rad_64
	Mean_rad_*n*, *n* =4,/8/16/32/64: The average distance from the center to an invisible point along the *n* directions.
Area	Planimtrc_4, Planimtrc_8, Planimtrc_16, Planimtrc_32, Planimtrc_64
	The area of a sparse viewshed based on its pixel count along its *n* (*n*=4/8/16/32/64) directions.
surface	Rugosity_4, Rugosity_8, Rugosity_16, Rugosity_32, Rugosity_64
	The surface area (which considers the elevations of points) of a view shed divided by its planimetric area along its *n* (*n*=4/8/16/32/64) directions.
Shape	SCIF_4, SCIF_8, SCIF_16, SCIF_32, SCIF_64
	A discrete *shape complexity* index to characterize the evenness of radii along *n (n*= 4/8 16/32/64) directions in a sparse viewshed.
Visibility	Min_CEA, Med_CEA, Max_CEA
	Minimal (Min), Medium (Med), and Maximum (Max) to the cumulative escape adjacency (CEA)
Visibility	RtVisMin_250, RtVisMed_250, RtVisMax_250
	p points around R with the Minimal (Min), Medium (Med), and Maximum (Max) visibility to an R point, whose distance to E is ≤ 250 meters.
Visibility	RtVisMin_500, RtVisMed_500, RtVisMax_500
	p points around R with the Minimal (Min), Medium (Med), and Maximum (Max) visibility to an R point, whose distance to E is ≤ 500 meters.
Visibility	RtVisMin_1k, RtVisMed_1k, RtVisMax_1k
	p points around R with the Minimal (Min), Medium (Med), and Maximum (Max) visibility to an R point, whose distance to E is ≤ 1000 meters.

Appendix 2. Use Cases

The purpose of this note is to illustrate how to use MECH-App, MECH-WPS, and the MECH-CTS to perform different tactical studies. In the first two examples, we illustrate how to use the behavioral modeling subsystem to answer tactical inquiries, and the last three for the MECH-CTS related applications.

MECH-App

1. *Where may be the best long-term monitoring spots over a route?*

   The two key words "long-term" and "monitoring" imply that the top locations should have the best observability over R, as well as good protection from being seen by the victim. As a result, either the "observability + concealment" analysis, or "observability x concealment" could best answer the query.

2. *Where may be the three best target locations on R for planning a roadside IED/DF attack, and where might be the best monitoring locations for such attacks?*

   For rational insurgents
   A. A successful attack would require good observation of the targets, good aiming, and also good protection. To serve this purpose, the user can first use the basic measurement to produce a P heat map of some basic measurements in CA1, and then invoke CA2 to produce the results.
   B. Starting from the exposure or curvature analysis in CA1 to identify the highest valued positions on R. Then, use these points to identify best P points by using the R2P study.

3. *Where may be a good E location to attack a convoy?*

Different studies can be done in a hierarchical fashion. It starts with basic measurements of the R. Then, high level studies can be built up the heat map on R, based on the assumptions on the level of sophistications, team sizes, etc.

MECH-CTS

Case 1

It is known that the attack patterns of insurgents change with seasons. A team needs to plan for operations centered at a position X ($32^{o}\,00'\,00''$N, $67^{o}\,00'\,00''$E) for the spring season. It is known that the insurgent groups around X have high mobility, with their reach to 120 km.

Case Design

The team leader wants to use the classifier to predict high risk locations at proximity of X at the spring time.

Analysis Procedure

1. The team leader decides to include all possible attacks that may have been planned by the insurgent groups within their mobility, but confines the time period of the training data

to 1/1/2012-5/1/2012. He selects the event type as IED and then enters the time period, and the study location. (See user guide Figure 3.2)

2. He then selects z-score and min-max for normalization. He further chooses stepwise (STP) and PCA for feature reduction, and kNN and DA as the two ML methods. By clicking the Run button, MECH-CTS will produce 8 classifiers based on the combination of the normalization method, feature reduction, and ML training method as follows: z-score-STP-kNN, z-score-STP-DA, z-score-PCA-kNN, z-score-PCA-DA, min-max-STP-kNN, min-max-STP-DA, min-max -PCA-kNN, min-max-PCA-DA.
3. The team leader can view the performance of these classifiers by clicking the "Result" button. The lower the error rate, the better the classifier is.
4. Any or all of these classifiers can be uploaded to the MECH-WPS for classification of an unknown location nearby X during the operation phase.

Case 2

As one part of periodic system maintenance work, an analyst is tasked to review the ML classifiers for the area around X to identify which of the features have most and least impact on the classification training process.

Case Design

The analyst can use the "New Group" panel under the "Feature Group Management" to visually inspect the performance of each feature.

Analysis Procedure

1. Same as step 1 of Case 1.
2. Open "Feature Group Management" window by clicking on the "Expert" button.
3. Click on "New Group" button to switch to "New Group" panel.
4. The analyst wants to examine the whole feature list to identify all the features that exceed an empirical threshold of discriminative power. He clicks on the feature's plain text label to view the discriminant ability from the box plot figure, e.g., Figure 3.4. The less the boxes of two classes overlap, the better this feature is in discriminating event points vs. non-event points.
5. He clicks on the “select” checkbox for every feature that exceeds the threshold.
6. He enters a description of this study in the view NG3 area (see Figure 3.4), and then clicks the “Add group” button. By clicking the “Confirm” button, the display returns to the classifier training interface.

Case 3

For the same team to operate at the location X, the team leader knows specifically that the insurgents would choose locations that are easy to escape as one of the major criteria in planning their actions. He had created a feature group called “Escape Features” in earlier planning.

Case Design

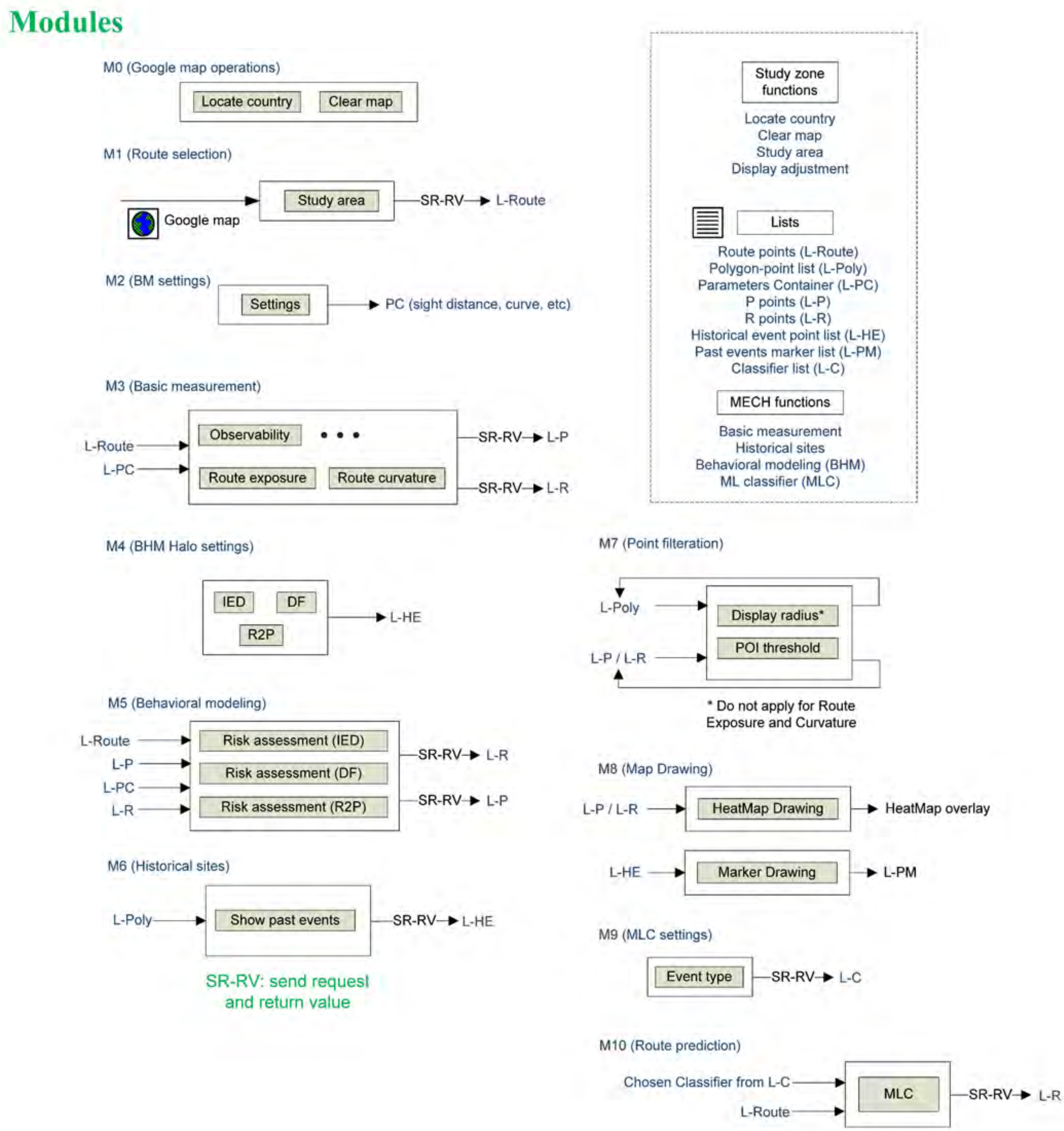
To ensure that the escape route is reflected in the classifier training, the team leader can manually select features by clicking the "Expert" button in addition to the automatic training process mentioned in Case 1.

Analysis Procedure

The general procedure for Case 2 is the same as that of Case 1, except that the following actions need to be added to step 2.

1. The user clicks the "Expert" button to open the feature group management window. (e.g., Figure 3.3)
2. The user clicks the "Select Groups" button to switch to the Select Feature Group panel and clicks the checkbox for the "Escape Features" feature group.
3. The user clicks the "Confirm" button to return to the Classifier Training interface and goes on with step 3 of Case 1.

Appendix 3 MECH-App software modules



Appendix 4. MECH classifier training flow chart