Simulation Models for Economics: A simulation model for defense of sensitive targets in an urban context

Umberto Forte

Contents

Part I
Introduction

The current events indicates how the risk for a terrorist attack it is not just a risk for safety of millions of people, but also an expense's item which authorities responsible for public order have got to deal with. The problem for the defense of sensitive targets certainly represent a complex problem not easy to analyze, and in order to address it in an appropriate manner, we should take into account several factors.

In such cases it is useful to consider the issue in a simplistic way by breaking down its main features, only focusing on the characteristics involved. The main objective of our study is to analyze the defensive strategies to functionally manage costs without compromising the safety of citizens.

0.1 Agent-based approach

In this paper we address the problem of the defense of sensitive targets through an agent-based approach, which allows us to focus on the single interactions between police officers and criminals, and to test performances of a particular security system.

In field of Social Science, ABMs prove to be very useful because they represent the only way to experimentally inspect some non reproducible phenomena such as the social phenomena, simulating the agnets' social behaviors and their interactions over time.

To this end we chose to build our model using the NetLogo 5.2 framework, a powerful tool for Agent Based simulations, in the specific we adopted an **intrusion model** in which criminals are able to blend in to the crowd of common citizens trying to beat the police officers' surveillance.

1 The model

1.1 An intrusion model and analyzed through Bayesian probability elements

The model consists on a game in which criminals aim to reach targets to destroy them, while the police officers have to try to arrest them in order to prevent the risk for an attack to anyone of the buildings located on the map.

We assign the *criminals* at random among the *citizens* breed by attributing to them an identifying characteristic, that is the possession of a weapon, by a percentage depending on the global rate of criminality.

The police officers divide in two classes: *garrison* officers, which are responsible for buildings' surveillance, and *patrol* officers, which move around in the map frisking the much closer citizens to them. Both types are able to examine the citizen characteristics in order to identify a potential criminal, assessing the level of dangerousness of the single individuals the inspect according to a specific subjective criterion.

To make such an assessment the officers compute an "*Alert Level*" by considering some specific inquiring characteristics of citizens; such as if they did travels abroad and for what destinations, if they paid their airfare with a credit card or cash, if they are dressed in a suspicious manner or normal.

In the end, after having estimated it, the police compare this "*Alert Level*" with two thresholds in order to classify the examined citizens as dangerous or innocuous. If the "*Alert Level*" detected for a given citizen should exceed the upper threshold, classifying the citizen as a security threat the police officers will remove them.

In case the Alert Level would lay in between the lower and the upper threshold the police officers, being unable to recognize with certainty a possible threat or not, will classify them as "*suspected*" so as to examine again later through a second level of inspections, in which the policemen will check if the suspected individuals possess or not a weapon, removing them in case of a positive result. The remaining citizens, those who show to have an Alert Level less than the lower threshold, will remain in the simulation because police officers consider them innocuous.

But is it true that the citizens with an Alert Level below the lower threshold are all innocents? And citizens exceeding the upper one? Are they really all criminals? The answers to this question is provided from the Bayesian statistics, which applies in case of an incomplete information about the event on the basis of which we have to take decisions. In these cases an agent will simply estimate the likelihood with which a statement may be true rather than false.

So what we can do is just make an estimate based on probability rules. In the bayesian approach the probability represent the degree of information about an event, for example it is assigned equal probabilities to all outcomes (with an uniform distribution) when we completely ignore the real nature of the event.

The Bayes theorem says that every time we receive a new information about the event we must update the assigned probability in order to improve the performances of our estimation.

In the article written bySalop et al. [2], he examines the episode of a car crash which happened in the small city of Lorain (Ohio) between a car and a taxi.

In that city there were only two taxi companies, the Blue Cab and the Green Cab. The car crash happened during the night and an eyewitness could observe it.

Now if a judge don't know anything about this car crash he will blame with 50% of probability the green taxi company and with 50% the blue one.

But he will consider at 75% the preponderance of evidence for a Blue taxi, after having heard the testimony of the eyewitness, who confess to have seen a blue taxi during the car crashing.

Then if the judge also considers that in the city the relative number of green taxi is the 85% of the total number, he will assign an other different level of likelihood for green or blue taxi.

The likelihood represents the probability that we assign to the event which we are studying assuming that we already dispose of some informations about it, for example if we know that there is an higher probability of being a criminal for 30 years old citizens we will suspect more about them.

So if we know some additional informations we can use the Bayes theorem for updating the "*prior probability*", which is the probability we have got before we receive new informations, to better estimate the "*true probability*":

$$P(A|B,I) = \frac{P(B|A,I)P(A)}{P(B)}$$

It says that we have to substitute the "*prior probability*" (P(A)) with the likelihood (P(A|B,I)) which is the updated probability to observe an event A given the previous level of informations "I" by assuming that event B already occurred.

In our case the suspects of policemen may be unfounded, so we have also to consider the percentages of False Positives, which are citizens who have an Alert Level that exceeds the Upper Threshold but that haven't got a weapon (citizens arrested wrongfully). At the same time we have also to consider the opposite case in which citizens can have an Alert Level lower than the bottom threshold but still be criminal (so criminals which are able to evade police controls and destroy targets).

In this work we made the number of False Positives depending on some social factors related to a tightening of controls, often the cause violations of civil rights of citizens. Therefore, we randomly assign the False Positives' rate by a distribution depending on social factors such as the "*level of democracy*" of the considered Country or the "*media attention*", trying to reproduce the grater presence of police abuses in countries where there is sufficient freedom of information.

We want to reproduce is that in this model the False Positives' rate, which represents the number of civilians that have been wrongly arrested, decreases when a Country *level of democracy* increases.

While by choosing to introduce or not the media attention through means of a switch command, we can reduce or not the False Positives' rate, because as a result of the media attention the authorities are usually more cautious in fielding such preventive measures.

For example just thinking to the dictatorial regimes, where leaders typically adopt repressive measures to preserve the public order, frequently arresting innocents in order to avert the risk of criminal attacks.

Then we make that the rate of False Negatives, which represents the number of true criminals misclassified during the repeated checks of policemen, depending on the tiredness of policemen.

To do this we created a policemen-own called "*tiredness*", which increases by one unit for each ten controls the considered officer carries out, and by one unit for every 500 walking steps for the patrol policemen. So we compute a "*tiringEffect*" that is a factor which acts on False Negatives rate, depending on the average tiredness of all the policemen.

1.2 Criminal preferences

In this model the criminals take into account the different targets in order to assess what is the best to attack according to some "*criminal preferences*".

We decided to set these preferences taking into account five possible goals which criminals try to achieve during an attack, such as: the goal to injure as many citizens as possible, probably aiming to give more clamor to their deeds;

to attack the symbol of a different religion, such in case of the fundamentalist terrorism;

to attack the politic establishment of a Country by damaging institutional buildings (such as the parliament, schools, or city halls), as in case of the political terrorism (Eire,or the italian Brigate Rosse);

to damage elements of a Country cultural heritage, such as museums, monuments or archeological sites, in order to erase historical traces and symbols of a Country ;

or even to focus on the less guarded targets, in order to increase the chances of succeed and to avoid being caught.

We compute the *criminal importance* of a target by considering some features we attribute to it, corresponding to percentages of belonging to a given category of targets ,such as: crowded places, cultural heritage sites, if it is or not an institutional place, if it is a religious site, and if it is a enough guarded site or not. Then we compute the weighted average of these features by the attacking preferences of criminals, that we set as external parameters at each simulation's run.

$$criminalImportance = w1\times[injureCitizens]+w2\times[attackReligiousTarget]+ \\ w3\times[attackCulturalHeritage]+w4\times[attackAnInstitution]+w5\times[noGarrisonsHere]$$

We compute in the same way the Alert Level level of citizens being inspected by policemen, by weighting features of citizens by the suspect parameters, representing the judgement criteria of policemen.

So the criminals can choose their favorite target among all the existent ones which are in a certain spatial radius corresponding to the one with the maximum "*criminalImportance*", and they will move to that aiming to destroy it.

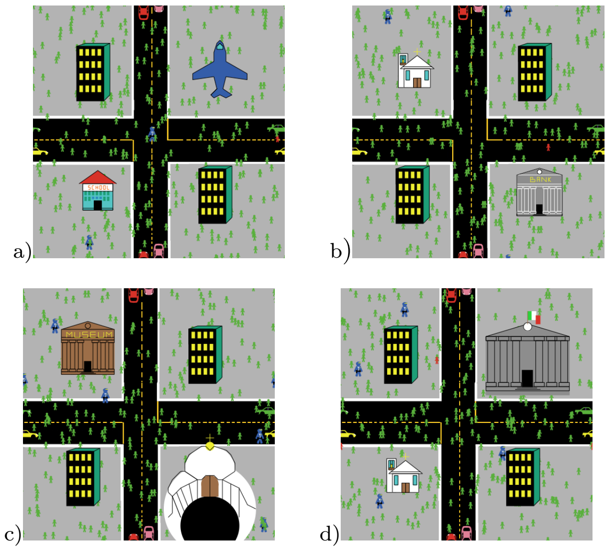
If the target is well guarded and the police officers are not too much tired the attacking criminals will not be able to destroy it, because we introduced the presence of undetectable criminals (the false negatives described by the Bayes theorem) that we have linked to the level of tiredness of policemen, and this "false innocents" will be able to to evade the surveillance destroying the targets anyway.

In this work we studied four different target settings with different target types and several allocations of patrol policemen per target.

By studying these settings we are able to analyze the results of different criminal behaviors and defense strategies, in order to find the optimal management of public security providing the best performances at the minimum cost.

1.3 The study of cases

Through means of the specific switch commands we are able to test four main scenery, by this command it is possible to read 8 external files containing informations on target types and number of policemen per target (4 files for target types, and 4 to set garrisons).



From the interface (fig. 1) it is possible to variate parameters characterizing the criminal preferences, and others such as the highest level of tiredness at which we decide to substitute officers, and also the two social factors “democraticLevel” and mediaAttention”.

We can observe how the effects of also very appropriate defense strategies can lose their effectiveness if we change the criminal preferences, underlining how much is important be informed about the type of criminals which we have to deal with for predicting risk for an attack.

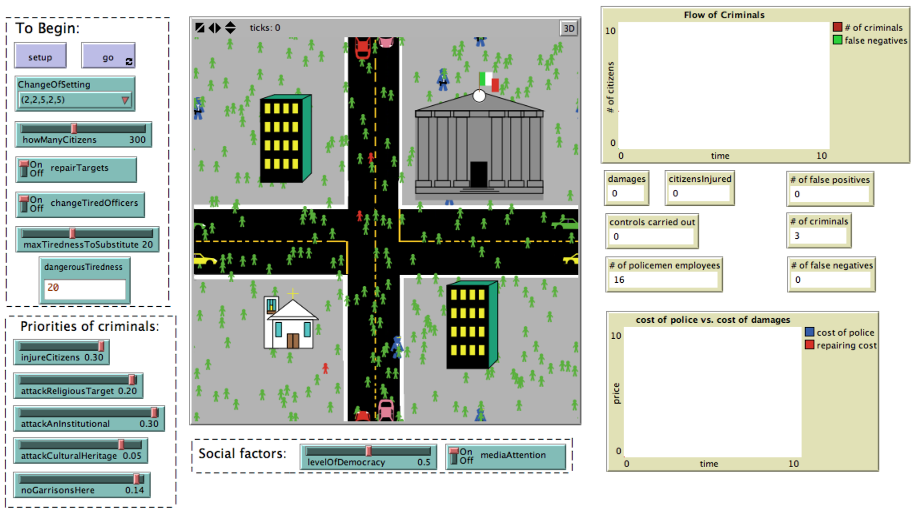
For example if the highest is “attackReligiousTarget, in the previous case (fig. a) to place more police at the church could be the right defense strategy.

Otherwise, if the highest is “attackAnInstitutional”, this deployment of police is not adequate, and criminals most probably will attack primarily the parliament .

As we previously discussed, changes in social factors would imply changes in number of false positives. We are able to change the specific setting through a switch command designed on the interface of the program, where we can choose among four cases. We gave names to different settings in which we describe the number of police officers we assigned to each target (from top to down and left to right), and the number of patrol officers inspecting all simulation area.

For example in figure 1 the combination 2,2,5,2,5 indicates the presence of 2 policemen on the first residential building (top left), 2 officers guarding the “parliament”, 5 for the church and 2 for the other residential building, and we assigned 5 agents as patrol policemen.

By using the *repairTargets* switch command we can decide if we want to replace the destroyed targets, and the *changeTiredOfficers* switch command if we want to substitute officers overcoming a certain level of *maxTirednessToSubstitute* in order to avoid disastrous tiredness effects, such as an higher presence of False Negatives.



1)

In the following pictures we show how, waiting a same time range of about 2000 ticks, we test different values for "*levelOfDemocracy*" varying the number of False Positives, that is the number of citizens which have been wrongly arrested.

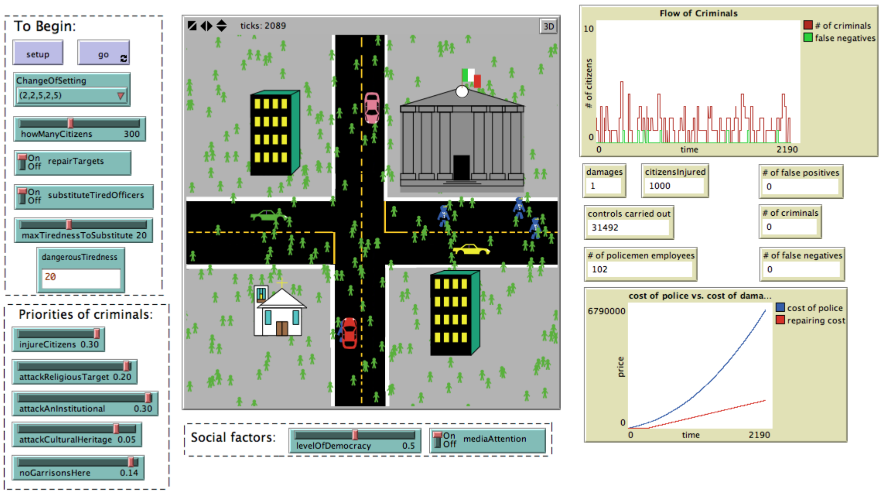
In this first picture (fig. 2) a medium level of democracy cause a very small number of False Positives and in this case zero.

From the costs plot (in the lower right side of the interface) we can observe higher repairing costs (implying higher damaging rate) with respect to the case in which we set a lower level democracy (fig. 3). This is the prove that undemocratic public order policies increase the effectiveness of defensive strategies resulting in higher cost in terms of False Positives, due to the reduction of individual freedoms.

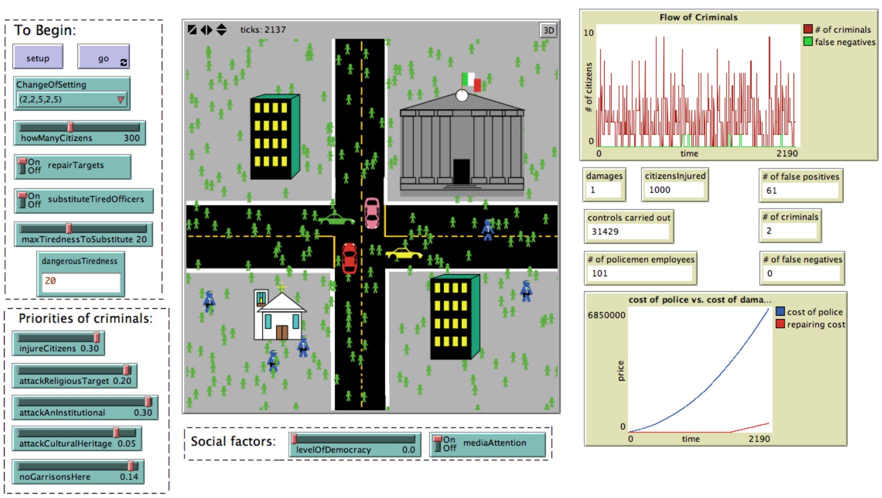
From the plot describing the "Flow of Criminals", in which we plot the number of criminals present in the simulation at each simulation step, we can observe for the first picture (fig. 2) a squared trend, determining a longer permanence of true criminals in the simulation due to an high level of democracy. While in the second case (fig. 3) we can note a piqued trend with higher maxima, because in this case true criminals are quickly identified and removed.

By comparing these two examples (fig. 2 and fig. 3) we do not observe a significantly change in cost for security caused by selecting democratic and undemocratic public order policies.

In the second case we can also observe on average an higher number of criminals, obtained by setting a lower value for the upper threshold in classifying the dangerous citizen, in this way we will randomly generate true criminals among a larger number of citizens matching these weaker conditions.



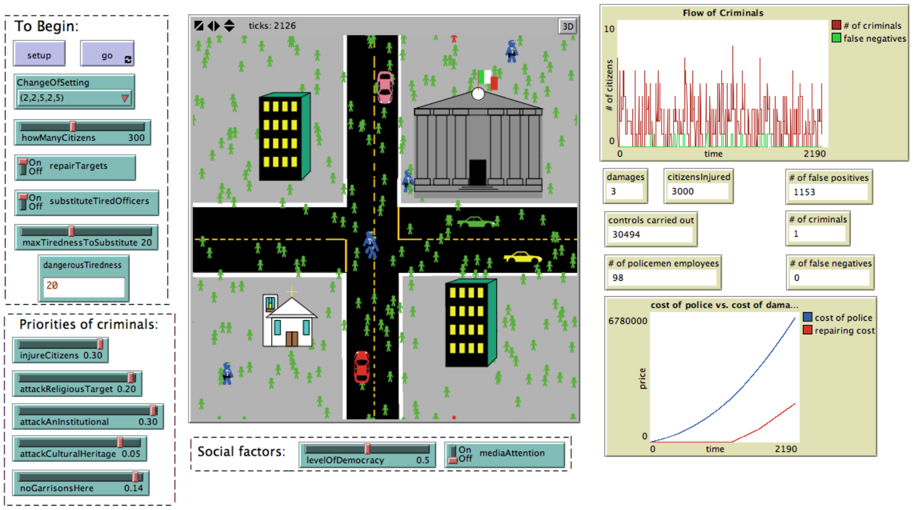
2)



3)

If we turn off the "mediaAttention" the upper threshold for alertLevel will be scaled down and so a higher number of criminals will be generated in this way, also causing a huge number of False Positives due to an increase in power abuses of policemen when no media denounces it.

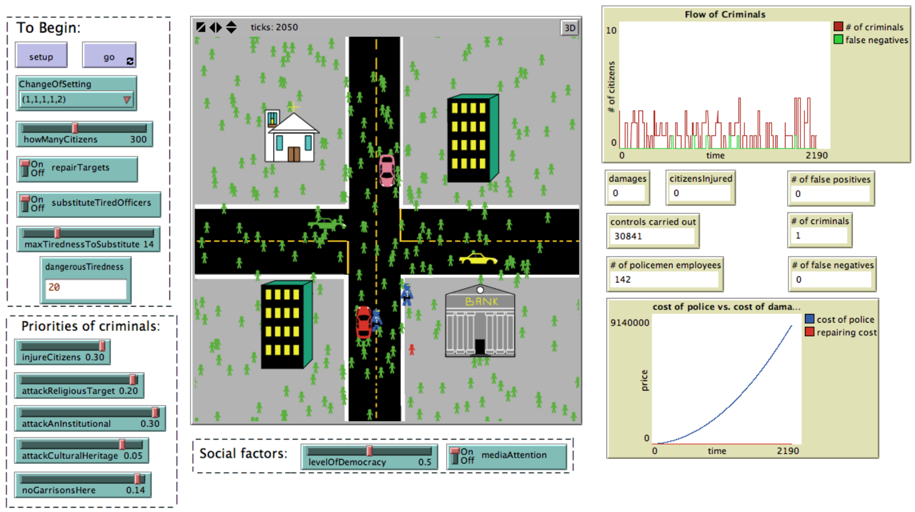
An other interesting effect from this scenery (fig. 4) can be seen through the plot of costs, in which the repair costs for building almost exceed the costs need for police management, reaching huge sums.



4)

Another interesting feature to note is that when we choose "*maxTirednessToSubstitute*" under the "*dangerousTiredness*", that is the tiredness level at which False Negatives begin to increase and so it represent the limit tiredness for policemen to continue to defend efficiently the targets. As showed in fig. 5 we obtain a lower average number of criminals but an higher number of policemen requested to put in place such a management, resulting in an higher cost. So in order to obtain an optimal defense policy in which we minimize costs providing an acceptable defense against possible attacks of False Negatives. In this model we can obtain a performance of this kind by selecting a *tirednessLevelToSubstitute* of $\frac{2}{3}$ of the dangerous tiredness for policemen(as in figure 5*).

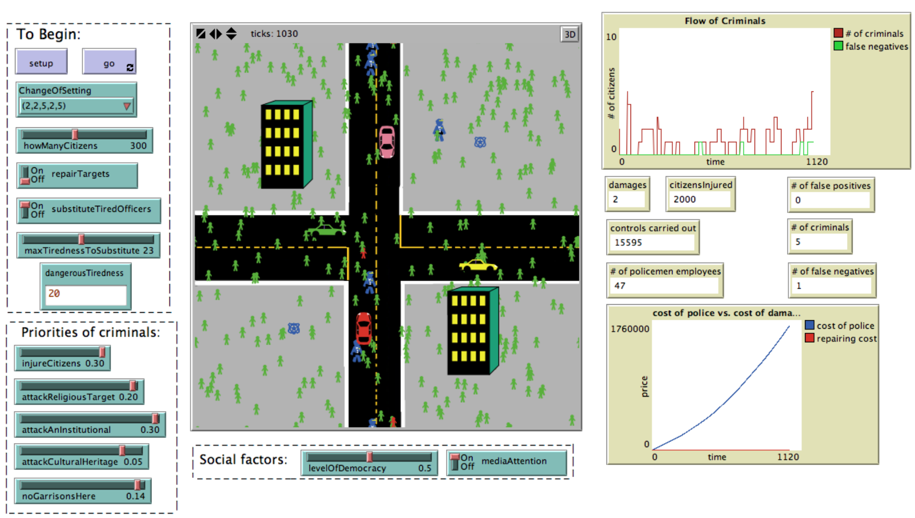
5)



5*)

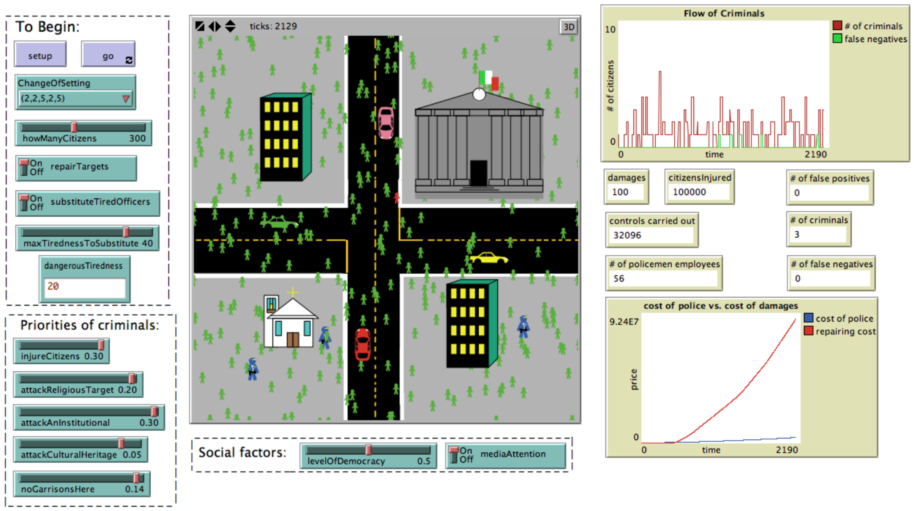
But the buildings still suffer a few damages inferred by those criminals who have not been arrested, the false negatives, as it is well in next picture (figure 6) .

In this figure it is portrayed the instant when the first damage occurred, and we can also note how this happen when there appear the first False Negative, which not happens so frequently because it can happen just when the tiredness of defenders reaches the 2/3 of the "*dangerousTiredness*" level.



6)

Studying the case in which we substitute the tired policemen after their tiredness exceeds the "*dangerousTiredness*", we can observe cost of replacing buildings exceeding cost for police management, and in particular we can observe what happen in case police average tiredness reaches too dangerous levels.

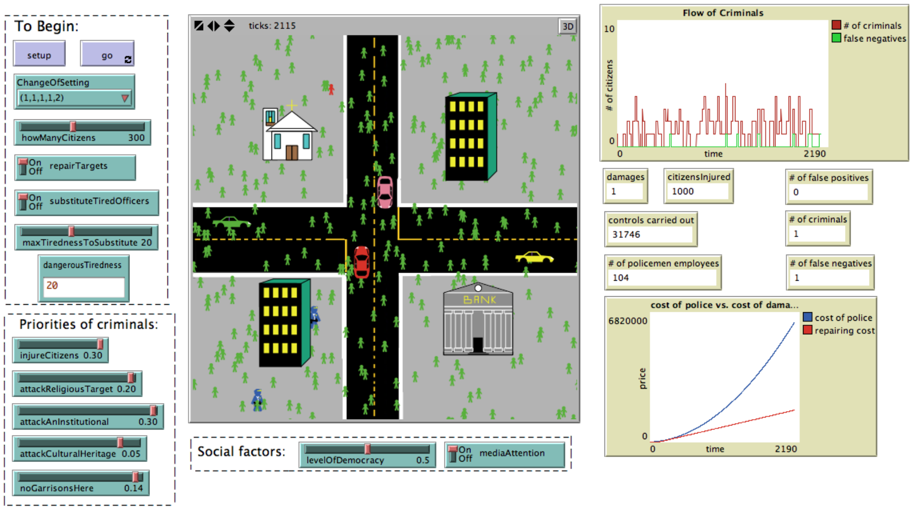


7)

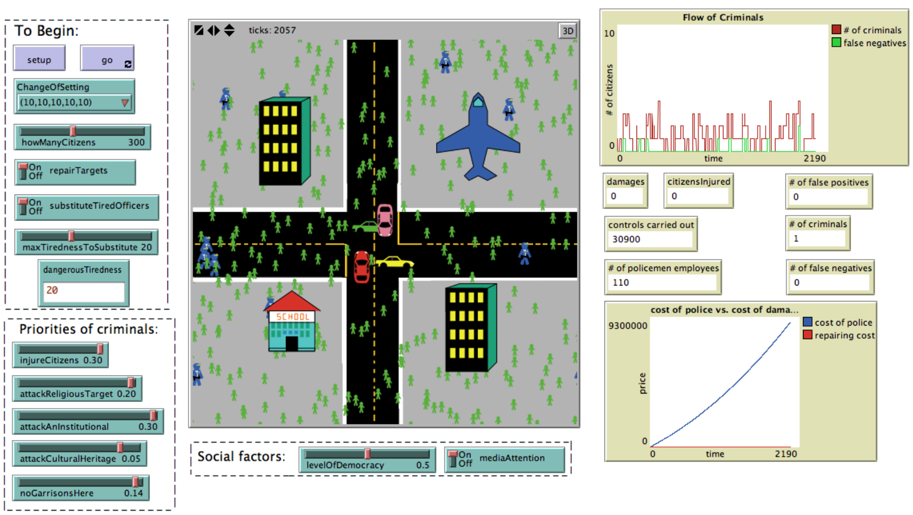
Then we investigated strategies which allow to maintain low management costs without attack risks increase, in the particular case in which the criminals prefer religious targets with low surveillance .

In figure 8 we can observe the low efficiency of a defensive strategy that consists of a low number of deployed agents (1,1,1,1,2), replacing them when they get medium “tiredness” levels. In this case the repair costs exceed the police management costs reaching about 12.2 millions, while damages increase to 15.

On the other hand, if we set the maximum number of policemen for each building (10,10,10,10 and 10), as it is shown in figure 9, we obtain lower repair costs and less damages inflicted on buildings and a smaller number of people injured.



8)



9)

Part II

The Code

In this section we report the most important parts of the program code where we describe the simulation dynamics, explaining in particular the reason of the technical choices we toke during the code implementation.

In this model I designed four main agent types (breeds), the patrol and garrisons policemen, which distinguish for moving around or standing still guarding targets, the targets, and the citizens which can be criminal or not at random, with different probabilities depending on their characteristics (their *turtles-own*).

First, in the "*setup*" procedure we assign the initial values of global variables, in particular we set the initial rate of False Negatives as a uniform distributed random variable in range between $x \in [0, 0.5\%]$. Then we set False Positives rate as a function of the "levelOfDemocracy" and "mediaAttention" in the same way of the upper bound of police judgment criteria, as we already discussed before.

The upper bound we use for determining the suspect individuals is an increasing function of *democraticLevel* and *mediaAttention*, while the FalsePositivesRate is a decreasing function of *democraticLevel* and it is rescaled by the presence of the *mediaAttention* in order to keep constant the number of criminals generated on a greater slice of the whole population because of a lower upperThreshold.

```
to setup
   ca
   assignArrays
   set FalseNegativesRate random-float 0.0001
   set BuildingsCost 0
   set citizensInjured 0
   set policemenCost 0
   set damages 0
   ifelse mediaAttention [set upperThreshold 0.8 + ((levelOfDemocracy) / 3 )
                          set FalsePositivesRate ((1 - levelOfDemocracy) / 12) ]
                         [set upperBound 0.6 + ((levelOfDemocracy) / 3 )
                          set FalsePositivesRate  ((1 - levelOfDemocracy) * 1.5)]
   set Controls 0
   set time 0
   set economicCost 0
   createTarget
   create-citizens howManyCitizens [ setxy random-xcor random-ycor
                                     set color green
                                     set shape "person"
                                     set steps 0   ]
   assignCitizensQualities
   reset-ticks
end
```

The *assignArrays* and *createTargets* procedures creates different target settings which reading 4 different text files we chose by a "switch command" from the interface, which allows to choose among four different settings of targets characteristics and number of policemen deployed. The results are shown in the following pictures.

Then we generate a specific number of citizens through a slider command assigning to everyone the green color.

By the *assignCitizensQuaities* procedure we random generate the characterizing qualities of each citizen by a uniform distribution, computing the alertLevel as the weighted sum of coefficients (c1, c2, c3, c4) corresponding to the agents own qualities (dress, ticket-buy, age, travels), times the percentages p1,p2,p3 and p4 of significance that the police gives to each of these citizens' qualities.

I chose the most plausible percentages of relevance as possible, that is: 20% of significance to the way of dressing of citizens, 30% for their airfare payment method , 10% for the age of citizens, and finally 40% of significance for their travels abroad.

Then I designed the most central part of the program, that is the part in which we decide how to generate the criminals; in this model citizens randomly become criminals with different percentages correlated to their *alertLevel* value.

if the have an *alertLevel* higher than the “upperThreshold”, a uniform distributed fraction of citizens in range $[0, (100 - FalsePositives)\%]$ become true criminals. And we assign them an additional identifying quality that is the weapon possession (weapon = “yes”) and we assign them red color.

In case they show an alertLevel between $[30\%, upperThreshold]$, citizens becomes criminals by a gaussian distribution centered on 0.6% and with standard deviation 0.05%.

Finally citizens with an *alertLevel* under 30% become criminals by a random uniform distributed fraction of citizens in range between $[0, FalseNegativesRate]$.

```
to assignCitizensQualities
   let newCitizens citizens with [steps = 0]
   ask newCitizens [let i random 2
                    if i = 0 [ set dress "normal"]
                    if i = 1 [ set dress "strange"]
                    ]
   ask newCitizens [let i random 2
                    if i = 0 [ set ticket-buy "credit card"]
                    if i = 1 [ set ticket-buy "invoice"]
                    ]
   ask newCitizens [set age random 100]
   ask newCitizens [let i random 2
                    let j random 4
                    if i = 0 [set travels "none"]
                    if i = 1 [
                    if j = 0 [set travels "South America"]
                    if j = 1 [set travels "Africa"]
                    if j = 2 [set travels "Middle East"]
                    if j = 3 [set travels "Asia"]
                    ]]
   ask newCitizens [let p1 0.2 let p2 0.3 let p3 0.1 let p4 0.4
                    let c1 0 let c2 0 let c3 0 let c4 0
                    if dress = "normal" [set c1 0.2]
                    if dress = "strange" [set c1  0.8]
                    if ticket-buy = "credit card" [set c2  0.2]
                    if ticket-buy = "invoice" [set c2  0.8]
                    if age <= 20 [set c3  0.2]
                    if age > 20 and age <= 30 [set c3  0.7]
                    if age > 30 and age <= 50 [set c3  0.9]
                    if age > 50 and age <= 70 [set c3  0.6]
                    if age > 70 [set c3 0.2]
                    if travels = "none" [set c4 0]
                    if travels = "South America" [set c4 0.4]
                    if travels = "Asia" [set c4 0.6]
                    if travels = "Middle East" [set c4 0.9]
                    if travels = "Africa" [set c4 0.8]
                    set alertLevel c1 * p1 + c2 * p2 + c3 * p3 + c4 * p4
                    ]
   ask newCitizens [if alertLevel > upperThreshold [
                       let i random-float 1
                       if i < (1 - FalsePositivesRate) [set weapon "yes" set color red ]
                    if alertLevel > 0.3 and alertLevel <= upperThreshold [
                            let i random-float 1
                                                       ]
```

```
                    if alertLevel <= 0.3 [
                           let i random-float 1
                           if i < FalseNegativesRate [set weapon "yes" set color red ]
                                        ]]
end
```

In the "*to go*" procedure we introduce the different procedure describing the model dynamics during the simulation run,such as the procedure by which the police controls occurs. I decided to give the police the ability to make two level control described by two different procedures, *firstControl* and *secondControl*.

In *firstControl* the policemen control citizens that have not yet been checked (with the *controlled* variable different from "yes") that are within a certain radius. They remove from the simulation all citizens having an alertLevel higher than the upperThreshold, assigning the "yes" value to the citizens-own called *suspected* for citizens having an alertLevel between the higherThreshold and the lowerThreshold,where we make this lowerThreshold depend on tiredness level of the police officers.

The global variable called "*Controls*" counts the number of citizens that have been already controlled, showing this value though an output on the interface. Then counting the number of citizens an officer controlled through the ControlsCarriedOut temporary variable, and we increase the tiredness of policemen by one every 10 controls.

In case of the patrolPolicemen we also increased the tiredness by counting the number of steps of the policemen ran in the "to move" procedure, by one unit every 500 steps.

By the secondControl procedure the policemen remove all citizens with weapon equal "yes" (the true criminals) who survived to the first control, identified as suspected.

Through this different kind of more deepened checks we avoid to repeat controls on citizens which have been already controlled, lightening the calculation time required in the process.

```
to firstControl
   ask patrolPolicemen [ let underControl citizens in-radius 5 with [controlled != "yes"]
                         let removed underControl with [alertLevel > upperThreshold]
                         let lowerThreshold 0.3 + (tiredness / 10)
                         ask removed [if weapon = 0 [set arrestedInnocents arrestedInnocents + 1]]
                         ask underControl [
                             if alertLevel > upperThreshold [die]
                             if alertLevel > lowerThreshold and alertLevel <= upperThreshold [
                                                                            set suspected "yes" ]
                             set controlled "yes"
                             set Controls Controls + 1]
                         set ControlsCarriedOut ControlsCarriedOut + count underControl
                         if ControlsCarriedOut >= 10 [
                            set tiredness tiredness + 1
                            set ControlsCarriedOut 0
                                                    ]
                         if steps >= 500 [set tiredness tiredness + 1 set steps 0]
                        ]
   ask garrisonPolicemen [let underControl citizens in-radius 5 with [controlled != "yes"]
                          let removed underControl with [alertLevel > upperThreshold]
                          let lowerThreshold 0.3 + (tiredness / 10)
                          ask removed [if weapon = 0 [set arrestedInnocents arrestedInnocents + 1]]
                          ask underControl [
```

```
                              if alertLevel > upperThreshold [die]
                              if alertLevel > lowerThreshold and alertLevel <= upperThreshold [
                                                                           set suspected "yes" ]
                          set controlled "yes"
                          set Controls Controls + 1]
                          set ControlsCarriedOut ControlsCarriedOut + count underControl
                          if ControlsCarriedOut >= 10 [set tiredness tiredness + 1
                                                          set ControlsCarriedOut 0]
                          ]
end

to secondControl
   ask patrolPolicemen [ let underControl citizens with [suspected = "yes"] in-radius 5
                         ask underControl [if weapon = "yes" [die]]]
   ask garrisonPolicemen [ let underControl citizens with [suspected = "yes"] in-radius 3
                           ask underControl [if weapon = "yes" [die] ]]
end
```

This is how we create targets and assign garrisons to them by reading the four files we mentioned before. The *import-drawing* command enable us to print the simulation graphic background by reading a specific file "city.png", realized painting the street view of a real city plan corner.

Then by reading the described files we assign main characteristics to targets, giving weights to them on the basis of their importance according to criminals preferences.

```
to createTarget
   create-targets 4
   if ChangeOfSetting = "(2,2,5,2,5)" [file-open "setting1.txt" import-drawing "city.png"]
   if ChangeOfSetting = "(4,2,2,4,2)" [file-open "setting2.txt" import-drawing "city.png"]
   if ChangeOfSetting = "(2,1,1,10,5)" [file-open "setting3.txt" import-drawing "city.png"]
   if ChangeOfSetting = "(2,5,4,2,10)" [file-open "setting4.txt" import-drawing "city.png"]
   while [not file-at-end?][
                    ask target file-read [setxy file-read file-read
                                          set color file-read
                                          set shape file-read
                                          set size file-read
                                          ifelse file-read = 1[set institutional "yes" set w4 1]
                                                              [set institutional "no" set w4 0]
                                          let i file-read
                                          if i = 0 [ set peopleInIt 0 set w1 0 ]
                                          if i = 1 [ set peopleInIt 100 set w1 0.2 ]
                                          if i = 2 [ set peopleInIt 1000 set w1 0.4 ]
                                          if i = 3 [ set peopleInIt 10000 set w1 0.6 ]
                                          if i = 4 [ set peopleInIt 50000 set w1 0.8 ]
                                          if i = 5 [ set peopleInIt 100000 set w1 1 ]
                                          set culturalHeritage file-read
                                           ifelse file-read = 1 [ set religious "yes" set w2 1]
                                                                  [set religious "no" set w2 0]
                             ]]
   file-close
   if ChangeOfSetting = "(2,2,5,2,5)" [file-open "settingPolice1.txt"]
   if ChangeOfSetting = "(4,2,2,4,2)" [file-open "settingPolice2.txt"]
   if ChangeOfSetting = "(2,1,1,10,5)" [file-open "settingPolice3.txt"]
   if ChangeOfSetting = "(2,5,4,2,10)" [file-open "settingPolice4.txt"]
   while [not file-at-end?][(foreach [8 8 -8 -8 ][8 -8 8 -8][
                             create-garrisonPolicemen file-read [setxy ?1 ?2
                                                                  set color blue
                                                                  set shape "police"
                                                                  set ControlsCarriedOut 0
                                                                  set tiredness 0]])
                             set policemenEmployees policemenEmployees + count garrisonPolicemen
                             let a file-read
                             create-patrolPolicemen a [setxy random-xcor random-ycor
                                                       set color blue
```

```
                                                    set shape "person police"
                                                    set size 2
                                                    set ControlsCarriedOut 0
                                                    set steps 0
                                                    set tiredness 0]
                             set policemenEmployees policemenEmployees + a]
   file-close
   ask targets [ let police count patrolPolicemen-here + count garrisonPolicemen-here
                 set w5  1 - ( police ) / 10]
end
```

Then I created a procedure thank to which criminals attack targets. As I previously said, the criminal preferences are based on a quality assigned to each targets called *criminalImportance*, calculated as the weighted average of criminal priorities, which are percentage we chose through some slider commands from the interface, times the coefficients assigned to targets by the createTargets procedure.

In this way we ask to criminals to move toward the target with the highest criminalImportance among the existent ones.

If a criminal reach a target where there isn't any garrison policemen, he will destroy it; otherwise, if the target is well defended by a garrison, we compare the average tiredness level of of all the garrison policemen who are guarding the "favorite" target with a threshold value (the dangerousTiredness) beyond which criminals are able to destroy this target. The same happen for FalseNegatives (if alertLevel $<= 0.3$), but with a lower threshold for tiredness.

When a target has been destroyed we increase by one the damages, represented by a global variable which counts the destroyed buildings, and the number of citizensInjured, which is a counter variable for people injured during the attack.

In the end, if some target has been destroyed, we call the "repair" procedure in order to check the four assigned positions of targets looking for what target has been destroyed, and by reading again the list containing the targets qualities which have been assigned during the "setup" procedure by the "assignArrays" procedure, we recreate the destroyed target assigning it the same qualities of the destroyed target.

```
to move-and-destroy
   let i 0
   ask targets [if criminalImportance = 0[
              set criminalImportance (w1 * injureCitizens + w2 * attackReligiousTarget +
                                      w3 * attackCulturalHeritage + w4 * attackAnInstitutional +
                                      w5 * noGarrisonsHere)]]
   let AverageCriminalPower 0
   let TrueCriminals citizens with [weapon = "yes"]
   ask TrueCriminals [
       if count targets != 0 [
          let bestTarget one-of targets with [
                                criminalImportance = max[criminalImportance] of targets]
                                face bestTarget]
       fd random 10
       lt random 360
       if count targets-here != 0 [
           ask targets-here [if alertLevel > 0.3[
               if count garrisonPolicemen-here = 0 [die
                                                     set damages damages + 1 set i i + 1
                                                     set citizensInjured citizensInjured + peopleInIt]
```

```
            if count garrisonPolicemen-here != 0[
               let averageTiredness sum[tiredness] of garrisonPolicemen-here / count garrisonPolicemen-here
               if averageTiredness >= dangerousTiredness [
                                          set i i + 1
                                          set damages  damages + 1
                                          set citizensInjured citizensInjured + peopleInIt die]]]]
              if alertLevel <= 0.3 [
                 let averageTiredness (sum[tiredness] of garrisonPolicemen-here / count garrisonPolicemen-here)
                 if averageTiredness >= (2 * (dangerousTiredness) / 3)[
                    ask targets-here[
                          set i i + 1
                          set damages  damages + 1
                          set citizensInjured citizensInjured + peopleInIt die] ]]]
    if i != 0 [if rebuildTargets[repair]]
end
```

In addition to the aforementioned procedures, in the "to go" procedure, we find a new one called "assignFalseNegatives" procedure in which we update the-False Negatives rate by taking into account average tiredness of officers scaled by 0.005%. Then by two switch commands we decide to repair targets or not, increasing the repair costs of targets, and and that we need for substituting the tired officers where we increase the costs by the amount of the salaries of the policemen.

In fact we increase the repairingCost by the average daily wage of 15 workers($\frac{20.000{,}00\$(pay)}{12months \cdot 30days} \simeq 56\$$), and we repeat the same thing when we substitute any tired policeman. Then there is the "reset" procedure in which at every 20 ticks counted by the *time* global variable, which is a ticks counter that we can be reset, is equal to forty all citizens will be removed and then they will be replaced in the same number recalling "assignCitizensQualities" procedure and setting time equal zero.

```
to go
   assignFalseNegatives
   if repairTargets [ set repairingCost repairingCost + (15 * 56 * damages)]
   if changeTiredOfficers [set costOfPolicemen costOfPolicemen + ( 56 * policemenEmployees)
                           substituteGarrisons
                           substituteOfficers]
   move
   move-cars
   reset
   firstControl
   secondControl
   move-and-destroy
   set time time + 1
   tick
end
```

Part III

Conclusions

As found from our observations, the role of the information about the type of intruders is crucial in determining the most appropriate defense strategy, in fact by these it is possible to infer their most probable attack choices and . As we can see from the following figure (fig. 6), the criminals firstly choose to attack that target corresponding to their preferences.

Differently from **Gutierrez et al. [1]** we examined the case in which the criminals attempt to reach targets we are defending, and in this work we dealt a more complex situation in which defenders have got to detect criminals among citizens thanks to the usage of Bayesian statistics. In this work we found that by assigning a huge number of police guarding the targets and substituting the tired policemen when they get a "medium" tiredness level we can generally prevent attacks and contain costs for the defense of targets.

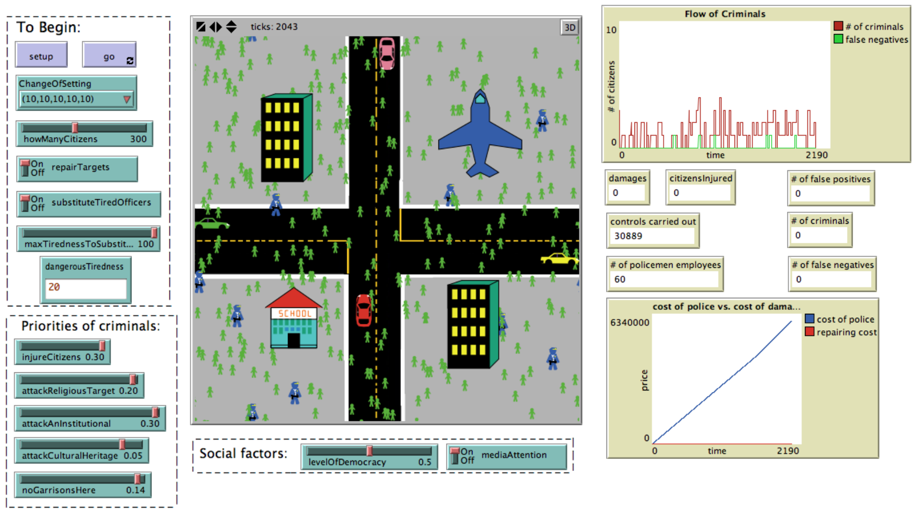
It has been noticed that the most efficient strategies to reduce the risk of a criminal attack consist in quite frequently substituting the tired officers or alternatively to reduce the level of democratic management of public order allowing civil rights violations as in the case of "preventive detention".

The latter method is questionable with regard to the protection of individual freedoms because it increases the number of wrongly arrested innocent citizens, but weighing both individual freedoms that the security of the citizens, we understand that there are special cases where security is certainly the priority and therefore has also arranged at a cost in terms of preventive detention.

From the point of view of economic costs a low "levelOfDemocracy" proves to be with no doubt the way in which we can reduce risk for a criminal attack without increasing or decreasing the costs for police management.

In this work we tested different kind of very detailed strategies in order to stop avoid criminal attacks, such as setting a low *maximum tiredness level* to reach before substitute the policemen agents, or choosing to assign the maximum number of garrison policemen (10,10,10,10) to every target both keeping the *maximum tiredness level* next to the dangerous one. Both strategies request quite the same cost, which is not so much expansive compared to the other strategies.

Finally the second one reveals to be the best defense strategy, because when we chose a *maximum tiredness level* of 100, so very much higher than the dangerous level of 20 as in the case showed in following figure (fig. 10), we can clearly observe significantly lower costs with respect to those obtained in the previous case (figure 9) in which we replaced officers more frequently.



In the article by Gutierrez et al. [1], instead, they deal with the problem of organizing the U.S. border patrolling in case of limited resources. He analyzes through mathematical tools as Monte Carlo simulation techniques and Game Theory the possible strategical choices of criminals, simulating the interactions between intruders and defenders.

He developed an Agent Based simulation using GAMMASys, a statistical simulating framework which allows to create a "world" by printing the image of a geographical area on hexagonal grids where the agents can move around. In this model a patrol have got to capture one single intruder, and this is made possible when they found both in the same hexagon. The criminals use a Monte Carlo estimation of the shortest and less dangerous path to reach a target, while defenders follow markovian strategies in the sense that their next step will depend only on their actual position an not on the other previous positions.

From the point of view of the mathematical formalism the work of Gutierrez presents a well designed model, but he didn't take into account of the case in which there are more intruders and these intruders can be mistaken with common citizens and vice versa, introducing the issue of civil right violations by policemen during checks in which they try to distinguish dangerous intruders from regular immigrants or simple citizens.

In our work, instead, by using a Bayesian approach, taking so into account about false positive and false negatives, it is possible to take into account of judgment mistakes. He also didn't take into account of the linkage between the level of tiredness and the efficiency of the defenders in detecting intruders, and this was explored in our work whereas the level of tiredness depends on how much the agents walk.

We have also quantified what we mean for limited resources introducing a cost study trying to minimize it. That led us to better deploy a fixed number of police garrisons in order to better protect each target without increasing the overall expense, and it is possible to better deployment police units on the basis of criminal preferences that in most of practical cases we don't know.

As we already said this subject it is a non trivial field of studies in which a lot of works in literature try to give different interpretations. The our work can be seen as one of the study useful for authorities responsible for public order, providing an estimate of the costs both economical and of relative importance of the targets to be defended and not only studying the possible criminal strategies as in Gutierrez et al. [1], but also all the possible policemen deployments on sensitive target.

References

[1] Gutierrez, E., J. Juett, and C. Kiekintveld: 2013, 'Generating Effective Patrol Strategies to Enhance US Border Security'. *Journal of Strategic Security* **6**(5), 152.

[2] Salop, S. C. et al.: 1988, 'Evaluating Uncertain Evidence with Sir Thomas Bayes: A Note for Teachers: Response'. *Journal of Economic Perspectives* **2**(2), 178–79.