2015 IEEE Symposium Series on Computational Intelligence

Enhancing Environmental Surveillance Against Organised Crime with Radial Basis Neural Networks

Christian Napoli*, Emiliano Tramontana* and Marcin Woźniak†
*Department of Mathematics and Informatics, University of Catania, Viale A. Doria 6, 95125 Catania, Italy
Email: napoli@dmi.unict.it, tramontana@dmi.unict.it
†Institute of Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland
Email: Marcin.Wozniak@polsl.pl

***Abstract*—A huge amount of data concerning the position of individual is often gathered in surveillance scenarios, to prevent crimes or to collect evidence of unlawful behaviour. Given the abundance of data available, detectives need advanced analysis means in order to set apart the interesting locations. This paper proposes a solution that makes use of radial basis neural networks to find the points of interests, i.e. locations that have been used for meetings, by surveilled people whose paths have been traced. In our solution, newly gathered data will be analysed in order to find the points of interest, and will also be given to our neural network for further training. Our results show that the proposed approach is accurate enough and can improve the unaided search for meeting points between observed individuals.**

I. Introduction

For the last several decades, many countries have resorted to geographical localization data for law enforcement [20]. Moreover, with the new technological advances on the area of Global Positioning System (GPS), such as the novel Galileo and Glonas systems [2], and thanks to the precisions and responsiveness of such systems, a big step ahead is expected in the operational management of modern criminal justice agencies and their strategic planning capabilities [18].

In the last years, the interest for such technologies and their application for the prosecution of offenders an the fight against the unlawful activities has tremendously grown and many national and international authorities are exploring the ways in which computer aided analysis can be applied in order to assist their operations. The main application areas for such a technology would be the geographical positioning of law officers, assisting the selection of proper locations for institutional facilities, identifying areas and streets affected by an high degree of crimes, focus the attention to geographical sites that are prone to organised criminal activities, follow the movements of known criminals or people suspected of being involved in criminal activities, pinpoint the location of members belonging to gangs or criminal structures in order to analyse or predict their behaviour, locate important meeting points where the criminal organisations manage their activities. A promising approach to fulfill such purposes could take benefit of crowdsourcing initiatives such as [1], where people are asked to indicate the locations of known criminal activities or unlawful behaviors. Unfortunately, while the crowd can be a powerful source of reliable data, such data must then be analysed, also in the plausible case of a very big amount of data.

It is then paramount to develop automated analysis systems that can cope with such a big amount of data, as well as able to give predictions or summarise knowledge about the possible commonalities among different data series (e.g. people coming to the same meeting point). An appropriate automatic tool in in this kind of scenario would: (i) be able to automatically extract data record, (ii) integrate the missing information by interpolation, (iii) reach a sufficient generalization degree, (iv) constitute a robust approach due to a low sensibility to data noise, (v) associate the raw data to the related knowledge and (vi) discover relations among different data. This work moves toward the development of such a tool, satisfying the above requirements. Our developed approach is based on the analysis of non homogeneous data fluxes gathered during environmental interceptions in order to pinpoint offenders or potential offenders involved in unlawful activities. Such data are generally collected on a large scale and are related to different people and geographical areas, as well as being characterised by a different time-frame and location sizes.

The field of neural network presents many solutions for the extrapolation of relations and models in associative systems [16], for human-behaviour clustering [26] and also to model human-based relational interactions [15]. Therefore the developed approach has been mainly based on Radial Basis Function Neural Networks (RBFNNs) [14], which are able to cluster and classify the data also revealing their geometrical properties. In our approach, the RBFNN has been developed in order to classify the data coming from pin-pointed vehicles belonging to people suspected of unlawful activities. The developed training procedure permits us to use the RBFNN to detect the most probable common destination for a group of vehicles, therefore uncovering the hidden location of a possible meeting point for those individuals. As we can observe from experimental data, it is highly probable that the suspected will not drive directly to the established meeting point, instead he will try to confuse the agent trying to follow their car. Moreover, the suspects probably will not park the car in the immediate proximity of the meeting point, and they will not come at the same moment. The RBFNN has been proven robust to this kind of human-introduced noise, being able to spot the locations in the form of centroids in their Radial Basis Function (RBF) units. The developed system is continuously fed with new coming data from both mobile sensors (GPS spying devices) and human made observations (law enforcers investigating in real time the movements of a subject). In a typical scenario, the RBFNN of the system are continuously trained and the updated centroids continuously extracted in

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.209

order to give some advice on the possible location of a meeting point. In this manner an agent or a police team could take preemptive actions (e.g. place new spy devices, audio or video recorder, wiretap the telephone lines, or even prepare an ambush, etc.). The functionality and structure of the system will be explained in the following.

II. Radial Basis Function neural networks

For identifying the locations where people meet, we have devised a solution based on Radial Basis Neural Networks (RBFNN). Such a kind of network exhibits a topology similar to Feed Forward Neural Networks (FFNN), with the main difference being the activation function that for RBFNN, as the name suggests, is a radial basis function (RBF), which is used by neurons on the first hidden layer that, consequently, is called RBF layer. RBFs are extremely useful when the statistical distribution of data and the related geometrical properties in the data space are relevant for the problem at hand.The selection of this kind of transfer functions is indeed decisive for the speed of convergence in approximation and classification problems (for more details see [8]). The kinds of activation functions used for RBFNNs have to meet some important properties to preserve the generalisation abilities. In addition, these functions have to preserve the clusterisation abilities of the RBFNNs. This kind of neural architecture if correctly trained can generate a model for the data features [21], [37].

A. Background on RBFNN structure and topology

In the standard RBFNN model, the input neurones are used as distribution units that supply the same input values to all the neurones in the first hidden layer. Such hidden neurones are called *RBF units*. Each *RBF unit*, also dubbed *pattern unit*, is activated by means of a RBF function f so that, given the centroid vector $\mathbf{c}$ and the input vector $\mathbf{u} \triangleq \mathbf{y}^{(0)}$, the j-th output from the related *RBF neuron* $\mathbf{y}_j^{(1)}$ is

$$\mathbf{y}_j^{(1)} = f\left(\frac{\sqrt{\sum_k |\mathbf{y}_k^{(0)} - \mathbf{c}_j|^2}}{\beta}\right) \tag{1}$$

where β is a parameter that determines the cluster shape. The $\mathbf{y}_j^{(1)}$ is given as input to another hidden layer (called summation layer) where a weighted sum is performed so that the i-th output $\mathbf{y}_i^{(2)}$ results

$$\mathbf{y}_i^{(2)} = \sum_j \mathbf{W}_{ij}\mathbf{y}_j^{(1)} \tag{2}$$

where $\mathbf{W}$ represents the weight matrix consisting of a weight value for each connection from the j-esime *pattern unit* to the i-esime *summation unit*. The summation units work as the neurones of a linear perceptron network. Incidentally, such summation units give us the global output of the network $\mathbf{y}_i \triangleq \mathbf{y}_i^{(2)}$.

B. 2D RBFNN, centroids and region clustering

In our approach, we locate geographical *attractors* starting from the movement data related to a set of surveilled people. The devised RBFNN is given an appropriate representation of the localisation data inherent the person under surveillance and related to a predetermined geographical area. The attractors are automatically selected by the RBFNN in the form of centroids for a space region.

Each pattern unit within the RBFNN determines the characteristics of a centroid and a 2D radius identifying a 2D region, and such parameters are are given to the Radial Basis Function (RBF). The number of pattern units determine the number of regions and centroids that the RBFNN will be able to recognise. Once the number of pattern units and output neurones have been designed for the network, then a clustering is obtained that selects the proper centroid and the radius for each RBF.

For the problem under investigation, we need to obtain a clustering geometry for the geographical coordinates representing the known positions held by a surveilled person during time. Therefore, we will ask the implemented RBFNN to give us the coordinates of the centroids and the dimensions of the related region as $\{c_1, c_2, \sigma_1, \sigma_2\}_j$. In order to obtain the desired behaviour we have chosen a 2D-Gaussian-like RBF as activation function for neurones in the second layer, then the equation (1) becomes our implemented network in the form

$$\mathbf{y}_j^{(1)} = e^{-\left(\frac{\sqrt{\sum_k |\mathbf{y}_k^{(0)} - \mathbf{c_1}_j|^2}}{\sigma_{\mathbf{1}j}} + \frac{\sqrt{\sum_k |\mathbf{y}_k^{(0)} - \mathbf{c_2}_j|^2}}{\sigma_{\mathbf{2}j}}\right)} \tag{3}$$

Therefore, in the implemented solution it is possible to express $\mathbf{y_i}$ as

$$\mathbf{y}_i = \sum_j \mathbf{W}_{ij}\ e^{-\left(\frac{\sqrt{\sum_k |\mathbf{y}_k^{(0)} - \mathbf{c_1}_j|^2}}{\sigma_{\mathbf{1}j}} + \frac{\sqrt{\sum_k |\mathbf{y}_k^{(0)} - \mathbf{c_2}_j|^2}}{\sigma_{\mathbf{2}j}}\right)} \tag{4}$$

The weight matrix $\mathbf{W}_{ij}$, centroids vectors $\mathbf{c_1}$ and $\mathbf{c_2}$ and parameters $\sigma_\mathbf{1}$ and $\sigma_\mathbf{2}$ are found and adjusted during the training of the RBFNN. However, since the number of *output units* is very small, the RBFNN training can be simplified and the speed greatly increased [7].

III. The RBFNN-based proposed approach

A. The proposed training procedure

The main problem that we have to solve with the said RBFNN-based approach, is that of properly training the RBFNN in order to be then able to have centroid identification, considering that, in general, a RBFNN requires a supervised training to work [17], [29], [22], [23], [24], [32], [12].

We have solved such a problem by initially training the RBFNN using an unsupervised training with fake targets, i.e. a constant centroid vector (see Figure 1). In our experiment such a value is the linearised coordinates of the center for the rectangle enclosing all the observed geographical points that have to be clustered. In [5] it is shown that, in such a configuration, by means of the gradient descent training algorithm the following set of updating equations for parameters may be obtained. Firstly, let us consider the global mean squared error (MSE) of the network as

$$E = \frac{1}{2}\sum_i e_i^2 = \frac{1}{2}\sum_j (y_i - t_i)^2 \tag{5}$$

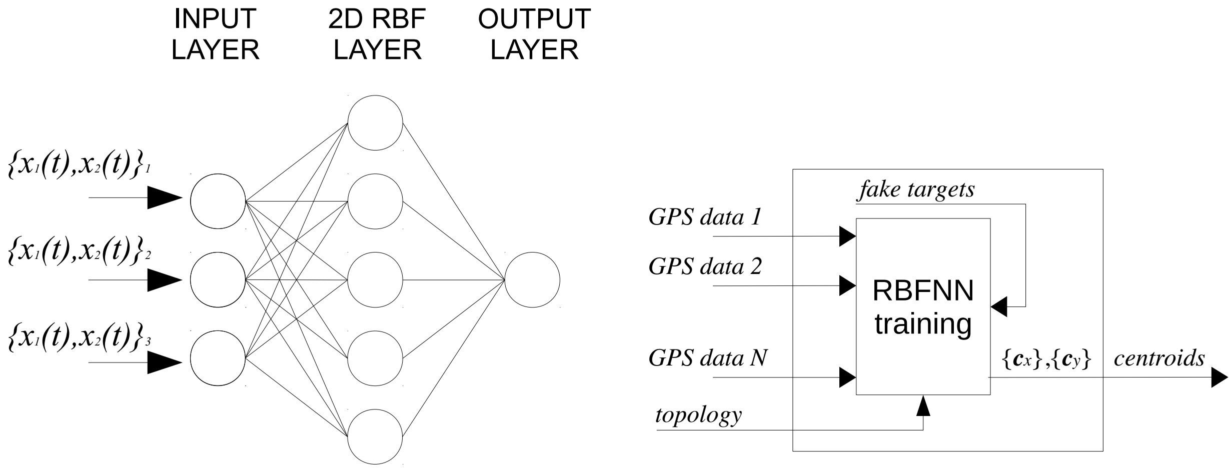

Fig. 1. On the left: the adopted Radial Basis Function Neural Network (RBFNN) architecture. The RBFNN is fed with GPS coordinates time series of different individuals, then trained in order to extract the centroids related to the given time series. On the right: the usage of the selected architecture for the extraction of the needed centroids coordinates.

where t_i represents the target value. Since the error is computed at each training iteration, it is then possible to update the weight matrix $\mathbf{W}$ as

$$\Delta \mathbf{W}_{i,j}(k+1) = \eta_w \, \Delta \mathbf{W}_{i,j}(k) \, e_i(k) \tag{6}$$

where k is the iteration number and η_w the learning coefficient for the weight update. It follows that at the next step the new weights will be updated in the form of

$$\mathbf{W}_{i,j}(k+1) = \mathbf{W}_{i,j}(k) + \Delta \mathbf{W}_{i,j}(k+1) \tag{7}$$

Similarly, it is then possible to update the centroid locations and the related spread as

$$\begin{aligned}
\Delta \mathbf{c_1}_j(k+1) &= \eta_c \frac{y_j^{(1)} - \mathbf{c_1}_j(k)}{\sigma_{\mathbf{1}j}(k)^2} \sum_i \Delta \mathbf{W}_{i,j}(k+1) \, e_i \\
\Delta \mathbf{c_2}_j(k+1) &= \eta_c \frac{y_j^{(1)} - \mathbf{c_2}_j(k)}{\sigma_{\mathbf{2}j}(k)^2} \sum_i \Delta \mathbf{W}_{i,j}(k+1) \, e_i \\
\Delta \sigma_{\mathbf{1}j}(k+1) &= \eta_s \frac{\left(y_j^{(1)} - \mathbf{c_1}_j(k)\right)^2}{\sigma_{\mathbf{1}j}(k)^3} \sum_i \Delta \mathbf{W}_{i,j} \, e_i \\
\Delta \sigma_{\mathbf{2}j}(k+1) &= \eta_s \frac{\left(y_j^{(1)} - \mathbf{c_2}_j(k)\right)^2}{\sigma_{\mathbf{2}j}(k)^3} \sum_i \Delta \mathbf{W}_{i,j} \, e_i
\end{aligned} \tag{8}$$

where η_c, η_s respectively represent the learning coefficient for the centroids and spread update. On the other hand, due to the network topology we have selected for this work, the output layer is composed by only one output neuron, therefore by defining

$$\begin{aligned}
\Delta \mathbf{W}_j &\triangleq \Delta \mathbf{W}_{1j} \\
e &\triangleq e_1
\end{aligned} \tag{9}$$

still respecting the formalism the equations in (8) can be simplified as

$$\begin{aligned}
\mathbf{c_1}_j(k+1) &= \mathbf{c_1}_j(k) + \eta_c \frac{y_j^{(1)} - \mathbf{c_1}_j(k)}{\sigma_{\mathbf{1}j}(k)^2} \Delta \mathbf{W}_j(k+1) \, e \\
\mathbf{c_2}_j(k+1) &= \mathbf{c_2}_j(k) + \eta_c \frac{y_j^{(1)} - \mathbf{c_2}_j(k)}{\sigma_{\mathbf{2}j}(k)^2} \Delta \mathbf{W}_j(k+1) \, e \\
\sigma_{\mathbf{1}j}(k+1) &= \sigma_{\mathbf{1}j}(k) + \eta_s \frac{\left(y_j^{(1)} - \mathbf{c_1}_j(k)\right)^2}{\sigma_{\mathbf{1}j}(k)^3} \Delta \mathbf{W}_j \, e \\
\sigma_{\mathbf{2}j}(k+1) &= \sigma_{\mathbf{2}j}(k) + \eta_s \frac{\left(y_j^{(1)} - \mathbf{c_2}_j(k)\right)^2}{\sigma_{\mathbf{2}j}(k)^3} \Delta \mathbf{W}_j \, e
\end{aligned} \tag{10}$$

Moreover, it is possible to demonstrate by means of a statistical theory method that the equations (10) make the centroids converge and stabilise (see [34]). Training in the said way has to use some precautions. Firstly, a small enough number of centroids have to be chosen, hence a network with a few *RBF units* is to be used; then when centroids are found a pruning procedure must be performed in order to locate a few (possibly only one) significative centroids. Duplicated centroids must be removed as well as the related neurones. In order to obtain a unique centroid distribution, also the value of σ must be changed during time (e.g. by increasing it of a constant factor). When a solution has been reached, the number of obtained centroids can be different than the number of required centroids. In this case a second pruning takes place: the centroid pairs, i.e. centroids close to each other, are substituted with a new centroid being the mean point between the two centroids. Such a second pruning is followed by successive training steps that make it possible for the remaining centroids to converge to a better value.

In this manner along with the weights training procedure (a standard gradient descent algorithm) we can determine the values of all the centroids and weights. While the RBFNN could be designed and trained as a standard feed-forward neural network (even having a greater number of neurons), the network is instead built up iteratively by removing RBF

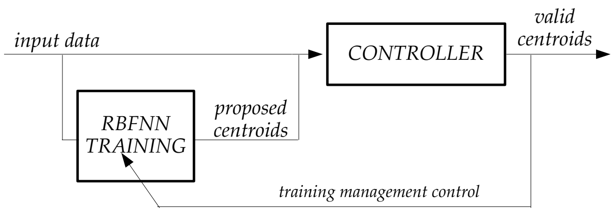

Fig. 2. The developed feedback system for the correct and timely management of the training sessions for the selected RBFNN architecture.

neurons.

In fact, the RBF layer (the first hidden layer) of our RBFNN is responsible for the fundamental task expected (see [38] for more details), therefore the pruning procedure should primarily take into account the RBF neurons instead of the output neuron that, for the purpose of this architecture, is only one unit.

B. Controlled training approach

Due to the nature of the chosen neural network architecture, and thanks to its small size in terms of number of RBF neurons, the training of such a network is very fast. This makes it possible, in principle, to have a continuous training, i.e. feed new gathered data to perform training (in a few seconds), and receive as output the needed data centroids.

Moreover, a realistic application to the scenario of environmental surveillance should cope with continuous updates of the GPS localisation time series of the people under surveillance. Such a requirement provides the motivation to have some degree of automation for the training of the RBF network with the new and updated data, perfecting, therefore, the obtained centroids.

The training procedure has to be monitored in order to avoid malfunctions (e.g. polarisations, overtraining, etc.). In order to manage and enhance the training session of the described RBFNN architecture, in this work we provided the RBFNN with an additional module for obtaining reinforcement learning. Figure 2 presents our applied solution. The RBFNN classifier is trained with a classical back-propagation training algorithm (BPTA), and embedded into a larger modular system.

After each training session, the controller evaluates whether it is possible to perfect the RBFNN by means of a pruning procedure. The synaptic weights of the connections among each RBF unit and the output unit are evaluated. If a weight results relatively too small with respect to other weights the connection is severed and the related RBF unit removed (see Figure 3). This permits us to restrict the number of centroids identified by the network, hence reducing the number of possible locations to place under surveillance after the RBFNN suggestions.

For the work here presented we used real data coming from the GPS localisation of several people with the intent to meet in a selected location. Unfortunately, at this stage of development this approach could not take advantage of the needed clearance in order to use data collected during official

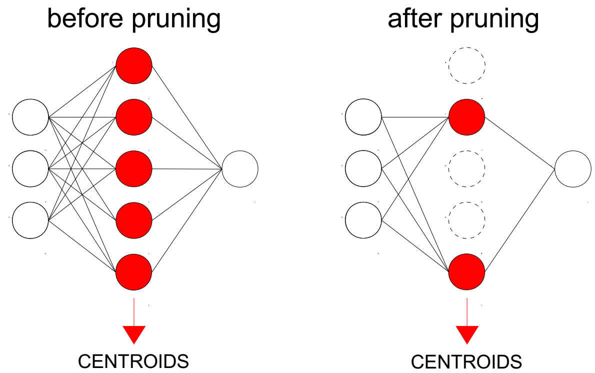

Fig. 3. Our training and pruning procedure

inquiries, therefore it was not possible to use data coming from real police investigations. In order to overcome this limitation, we asked to a large number of students to partecipate in a scientific gameplay.

C. Experimental data gathering

In order to collect realistic data, we have organised a gameplay by dividing students in 3 groups of players: citizens, criminals and policemen. While the policemen identities were disclosed, no information has been given to the players regarding the other people, therefore, without any further instruction, it was impossible to distinguish citizens and criminals. Each citizen and criminal was given a set of tasks. The given tasks should lead the players of the three groups to meet each other in the city. Meetings were not casual events, they were indeed organised on purpose, in order to simulate events like business meeting, shared activities, friendly encounters and, of course, unlawful affairs. Players were asked to note the route on the streets and the exact timing of every direction change. Notes were then translated in GPS coordinates by common route tracking and planning software.

To enrich the gameplay and make it a realistic scenario, some of the criminals were signaled to the policemen, other criminals had unknown identity, however all the criminals try to avoid meeting some policemen. For this, the criminals have been given the task to discover if they were suspected or not, and therefore if they could have been followed by electronically surveillance systems. Finally, it was a duty for the policemen to discover new criminals and arrest them. Such an action was possible only after a meeting with a known criminal, and, in this case, criminals would be arrested and out of the game. A realistic scenario was guaranteed by a scoring system, which benefits the policemen arresting a criminals and criminals avoiding arrest, but penalizes policemen arresting citizens and criminals being arrested.

IV. EXPERIMENTAL SETUP

A. Collected data and preprocessing

The data collected during the gameplay has been converted in GPS coordinates $\mathbf{x}$ associated to the related timestamp t, in

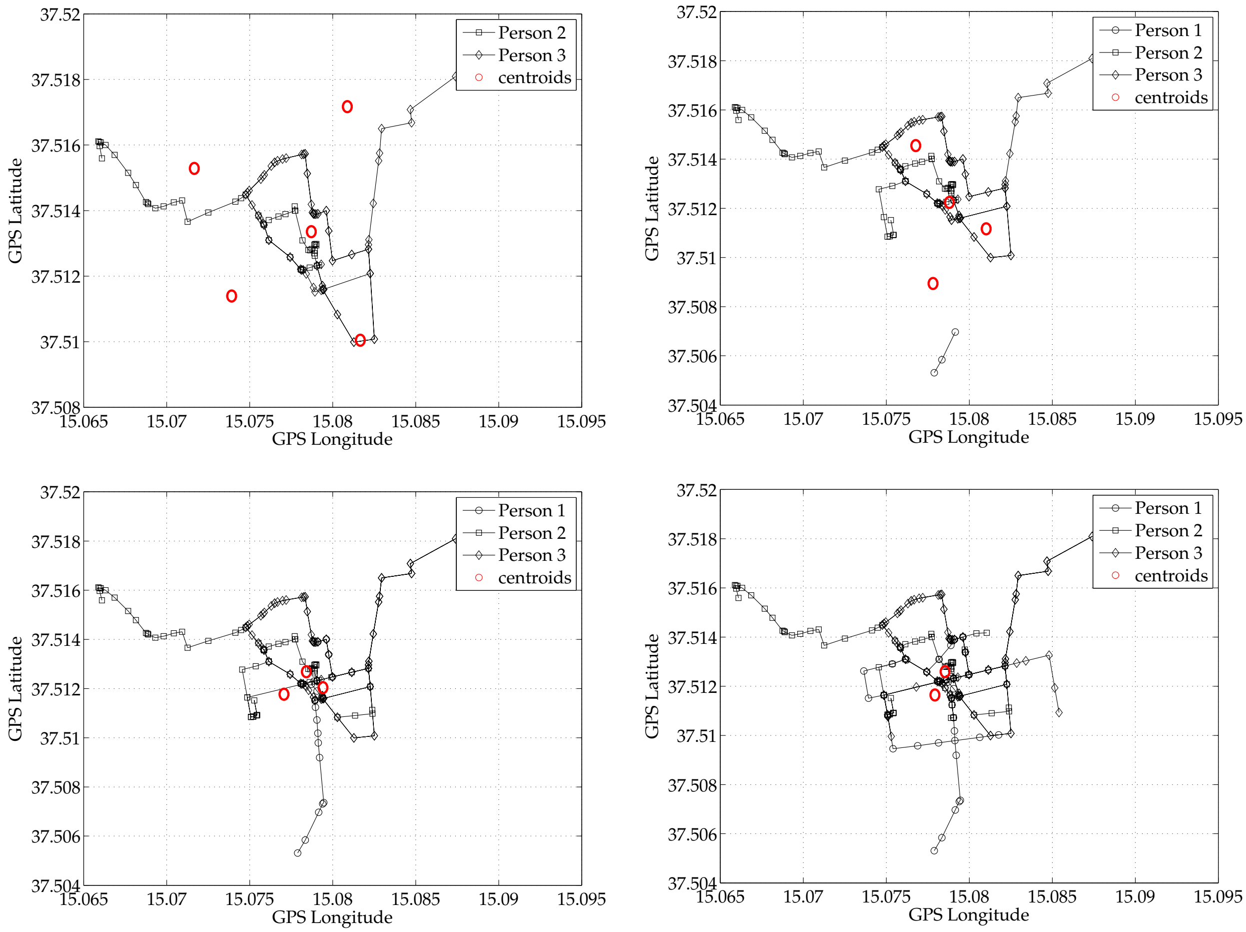

Fig. 5. Centroids positions with respect to the paths followed by surveilled people

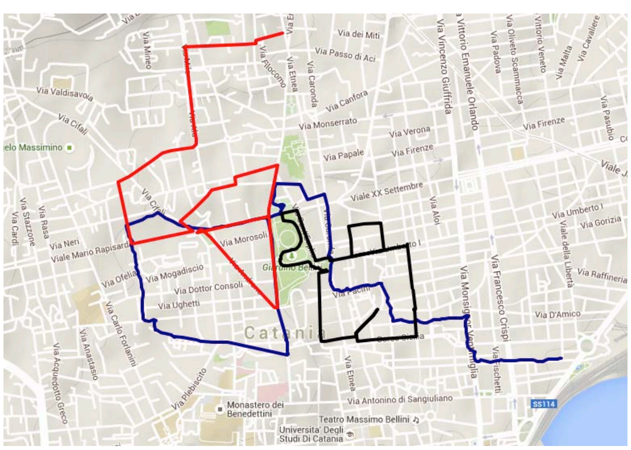

Fig. 4. An overlap of several traced paths and the corresponding geographical map

the form of a position vector in the GPS system:

$$\mathbf{x}(t) = [x_1(t), x_2(t)] \tag{11}$$

where x_1 and x_2 are respectively the longitude and latitude in GPS coordinates on the WGS84 spheroid [13]. In our hypothesis the students were moving at almost constant speed, therefore taking into account two successive entries $\mathbf{x}(t_1)$ and $\mathbf{x}(t_2)$ regarding the route followed by one player, it is possible to compute the GPS coordinates at a time $t_* \in [t_1, t_2]$ as

$$\mathbf{x}(t_*) = \mathbf{x}(t_1) + \frac{t_*}{t_2 - t_1} \{\mathbf{x}(t_2) - \mathbf{x}(t_1)\} \tag{12}$$

By means of equation (12) it is possible to cope with missing data (e.g. in a real investigation the spying device could lose the ability to record GPS coordinates for a certain time interval). Finally, the data are represented in time series of uniformly sampled GPS coordinates. In this way it is possible to omit the timestamp from the considered parameters, due to fixed sampling frequency of the data. In this case we will denote the time step in the time series with the greek letter τ in order to discern it from the GPS time stamp. Then, it is possible to univocally associate to a person (identified with an id μ) a time series X^μ composed by a certain number N of data regarding his GPS position during time

$$X^\mu = \{\mathbf{x}^\mu(\tau)\} \tag{13}$$

so that

$$\mathbf{x}^{\mu}(\tau) = \mathbf{x}(t_0 + \tau\Delta t)|_{\mu} \tag{14}$$

If several people are suspected to meet for unlawful reasons, the time series related to those people are fed as input to a RBFNN in order to discover the most probable location for a meeting point. Due to the extremely fast training and use of such an architecture this analysis can be performed online and in real time while surveilling the subject in motion and feeding at each time step the RBFNN with new data.

B. A typical case study

Let suppose that three people ($\mu \in [1, 3] \cap \mathbb{N}$) are suspected to conduct criminal affairs and that those people are followed by a spying device. Therefore, gathered data, i.e. the time series of the people positions, will be fed to the RBFNN. In this scenario the input vector at a time step τ has the following form:

$$\mathbf{u}(\tau) = \begin{bmatrix} \mathbf{x}^1(\tau) \\ \mathbf{x}^3(\tau) \\ \mathbf{x}^2(\tau) \end{bmatrix} = \begin{bmatrix} \mathbf{x}(t_0 + \tau\Delta t)|_1 \\ \mathbf{x}(t_0 + \tau\Delta t)|_2 \\ \mathbf{x}(t_0 + \tau\Delta t)|_3 \end{bmatrix} \tag{15}$$

On the other hand, since $\mathbf{x}$ is a position vector, the input set will be constituted by an array of data pairs in the form of

$$\mathbf{u}(\tau) = \begin{bmatrix} \{x_1(t_0 + \tau\Delta t), x_2(t_0 + \tau\Delta t)\}_1 \\ \{x_1(t_0 + \tau\Delta t), x_2(t_0 + \tau\Delta t)\}_2 \\ \{x_1(t_0 + \tau\Delta t), x_2(t_0 + \tau\Delta t)\}_3 \end{bmatrix} \tag{16}$$

where in equations (15) and (16) Δt represents the uniform sampling interval used for the time series.

V. Experimental results

To train this network, as stated before, we used a continuos training approach. This allowed us to continuously feed the RBFNN with new data at any new time step, in this manner the training procedure was perfecting the centroids estimate at each epoch. Moreover after a training cycle, when the error gradient reaches a near-static condition, the RBFNN was pruned from one neuron. In this experimental survey we tested the network with 45 case studies, each time starting with 5 RBF neurons, and then pruning the network until only 2 RBF neurons where remaining.

Figure 4 shows several paths that we have traced. Figure 5 shows an example of the continuous learning and pruning progression, and the related centroid motion (each diagram shows a different number of centroids). Initially, only two persons were moving, while a third person was added to the scenario after a certain amount of time. At the beginning, the 5 RBF units suggest 5 different possible centroids, which could indicate possible locations were the people could have met. While the people in this case study were followed (we have simulated the movements by means of a spying GPS device on their cars), the RBFNN was fed with new data, in the meantime the number of RBF neurons and related centroids was decreased due to the pruning effect. As we can see in Figures 5, the centroids typically converge to a small area. Such a small area, given by the rectangle enclosing the last remaining centroids, is a suggested area for the law enforcers to prepare all needed countermeasures (ambushes, human observation, further environmental spying devices, etc.).

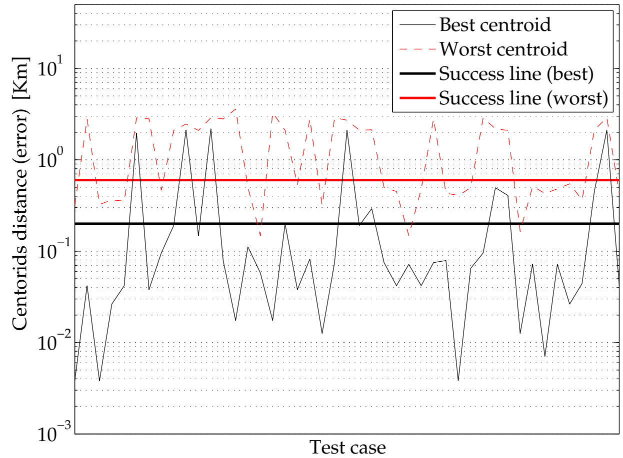

Fig. 6. Error as centroids distance from point of interests in several tests

TABLE I. Distances between centroids and point of interest

	Best centroid	Worst centroid
Min. error	$\sim$ 0 Km	0.16 Km
Mean error	0.04 Km	0.40 Km
Max. error	1.11 Km	1.56 Km
Success %	**82.2%**	**42.2%**

After the online testing of the RBFNN, we have compared the returned centroids with the GPS locations were the meeting took place. Figure 6 shows the error of the proposed system in terms of distance of the two centroids from the real position of the meeting. The mean absolute error was of 0.04 Km (with a maximum error of 1.11 Km) for the best centroid, while the worst centroid mean absolute error was of 0.40 Km (maximum error of 1.56 Km), as shown in Table I. We considered a successful goal for the best centroid when its distance of the error was less than 200 meters, and a successful goal for the worst centroid when its distance error was less than 600 meters. In this setup we reached a successful positioning for the best centroid in the 82.2% of the cases, and a successful positioning for the worst centroid in the 42.2% of the cases. Note that in several occasions the second centroid was positioned under the 200 meters distance. Therefore, we can conclude that the approach has given useful results for the purposes of this work.

VI. Conclusion

In the presented work we have analysed data consisting of the GPS locations of several people going around and having some interest in order to meet. The proposed approach is based on Radial Basis Neural networks due to their efficiency as scale based clustering systems [5], [28]. This kind of neural architecture is outstandingly versatile and prone to training-related architectural improvements such as neural growth [17] and pruning. The proposed approach automatically selects the most likely location of the meeting. The experiments have confirmed the efficacy of our solution.

For the future work we are developing a modular [9], [10], [11], [4], [33] custom analyser based on swarm intelligence approaches [25], [36] that can process a huge amount of data [19], [27], [30] by offloading to cloud resources [35], [3] the computational demand. For further works will be explored solutions based also on alternative approaches such as flexible neuro-fuzzy models [6], self organizing maps, UCT-based [39] and statistical decision threes based [31] approaches.

ACKNOWLEDGMENT

This work has been supported by project PRIME funded by Sicilia Region within PO FESR 2007-2013 framework and Consorzio COMETA.

REFERENCES

[1] Mafia maps. http://www.wikimafia.it/mafiamaps/.

[2] R. Alkan, H. Karaman, and M. Sahin. Gps, galileo and glonass satellite navigation systems gps modernization. In *Recent Advances in Space Technologies, 2005. RAST 2005. Proceedings of 2nd International Conference on*, pages 390–394, June 2005.

[3] G. Borowik, M. Woźniak, A. Fornaia, R. Giunta, C. Napoli, G. Pappalardo, and E. Tramontana. A software architecture assisting workflow executions on cloud resources. *International Journal of Electronics and Telecommunications*, 61(1):17–23, 2015. DOI: 10.1515/eletel-2015-0002.

[4] A. Calvagna and E. Tramontana. Delivering dependable reusable components by expressing and enforcing design decisions. In *Proceedings of IEEE Computer Software and Applications Conference (COMPSAC) Workshop QUORS*, pages 493–498, Kyoto, Japan, July 2013.

[5] S. V. Chakravarthy and J. Ghosh. Scale-based clustering using the radial basis function network. *IEEE Transactions on Neural Networks*, 7(5):1250–1261, 1996.

[6] K. Cpalka, O. Rebrova, R. Nowicki, and L. Rutkowski. On design of flexible neuro-fuzzy systems for nonlinear modelling. *Int. J. General Systems*, 42(6):706–720, 2013.

[7] H. Deshuang and M. Songde. A new radial basis probabilistic neural network model. In *Proceedings of IEEE Conference on Signal Processing*, volume 2, 1996.

[8] W. Duch. Towards comprehensive foundations of computational intelligence. In *Challenges for Computational Intelligence*, pages 261–316. Springer, 2007.

[9] R. Giunta, G. Pappalardo, and E. Tramontana. Aspects and annotations for controlling the roles application classes play for design patterns. In *Proceedings of IEEE Asia Pacific Software Engineering Conference (APSEC)*, pages 306–314, Ho Chi Minh, Vietnam, December 2011.

[10] R. Giunta, G. Pappalardo, and E. Tramontana. AODP: refactoring code to provide advanced aspect-oriented modularization of design patterns. In *Proceedings of ACM Symposium on Applied Computing (SAC)*, pages 1243–1250, Riva del Garda, Italy, March 2012.

[11] R. Giunta, G. Pappalardo, and E. Tramontana. Superimposing roles for design patterns into application classes by means of aspects. In *Proceedings of ACM Symposium on Applied Computing (SAC)*, pages 1866–1868, Riva del Garda, Italy, March 2012. ACM.

[12] J. Graham, J. Starzyk, and D. Jachyra. Opportunistic behavior in motivated learning agents. *IEEE Trans. Neural Netw. Learning Syst.*, 26(8):1735–1746, 2015. DOI: 10.1109/TNNLS.2014.2354400.

[13] J. W. Hager, L. L. Fry, S. S. Jacks, and D. R. Hill. Datums, ellipsoids, grids, and grid reference systems. Technical report, DTIC Document, 1992.

[14] S. Haykin. *Neural Networks: A comprehensive foundation*. Prentice Hall, 1998.

[15] A. Horzyk. How does generalization and creativity come into being in neural associative systems and how does it form human-like knowledge? *Neurocomputing*, 144:238–257, 2014. DOI: 10.1016/j.neucom.2014.04.046.

[16] A. Horzyk. Innovative types and abilities of neural networks based on associative mechanisms and a new associative model of neurons. *Lecture Note in Artificial Intelligence - ICAISC'2015*, pages 26–38, 2015. COI: 10.1007/978-3-319-19324-3_3.

[17] N. B. Karayiannis and G. W. Mi. Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques. *IEEE Transactions on Neural Networks*, 8(6):1492–1506, 1997.

[18] J. Karuppannan. Mapping and corrections, management of offenders with geographic information systems. *Corrections Compendium*, 30(1):7–9, 2005.

[19] Z. Marszałek, G. Woźniak, M. Borowik, R. Wazirali, C. Napoli, G. Pappalardo, and E. Tramontana. Benchmark tests on improved merge for big data processing. In *Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*, pages 96–101, 14-16 July, Quito, Ecuador, 2015. IEEE. DOI: 10.1109/APCASE.2015.24.

[20] R. Mercer, M. Brooks, and P. T. Bryant. Global positioning satellite system: tracking offenders in real time. *Corrections Today*, 62(4):76–83, 2000.

[21] K. Miller, T. Griffiths, and M. Jordan. Nonparametric latent feature models for link prediction. *Advances in neural information processing systems*, 22:1276–1284, 2009.

[22] C. Napoli, G. Pappalardo, and E. Tramontana. A hybrid neuro–wavelet predictor for qos control and stability. In *Proceedings of Artificial Intelligence (AIxIA)*, volume 8249 of *LNCS*, pages 527–538, Turin, Italy, December 2013. Springer.

[23] C. Napoli, G. Pappalardo, and E. Tramontana. An agent-driven semantical identifier using radial basis neural networks and reinforcement learning. In *XV Workshop "From Objects to Agents" (WOA)*, volume 1260, Catania, Italy, September 2014. CEUR-WS.

[24] C. Napoli, G. Pappalardo, and E. Tramontana. Improving files availability for bittorrent using a diffusion model. In *Proceedings of IEEE International WETICE Conference*, pages 191–196, Parma, Italy, June 2014.

[25] C. Napoli, G. Pappalardo, E. Tramontana, Z. Marszałek, D. Połap, and M. Woźniak. Simplified firefly algorithm for 2d image key-points search. In *Symposium on Computational Intelligence for Human-like Intelligence (CHILI)*, Symposium Series on Computational Intelligence (SSCI), pages 118–125. IEEE, 2014. DOI: 10.1109/CIHLI.2014.7013395.

[26] C. Napoli, G. Pappalardo, E. Tramontana, R. Nowicki, J. Starczewski, and M. Woźniak. Toward work groups classification based on probabilistic neural network approach. In *Artificial Intelligence and Soft Computing*, volume 9119 of *Lecture Notes in Computer Science*, pages 79–89. Springer International Publishing, 2015. DOI: 10.1007/978-3-319-19324-3_8.

[27] C. Napoli, G. Pappalardo, E. Tramontana, and G. Zappalà. A cloud-distributed gpu architecture for pattern identification in segmented detectors big-data surveys. *Computer Journal*, 2014.

[28] C. Napoli, E. Tramontana, G. Lo Sciuto, M. Woźniak, R. Damaševičius, and G. Borowik. Authorship semantical identification using holomorphic chebyshev projectors. In *Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*, pages 232–237, 14-16 July, Quito, Ecuador, 2015. IEEE. DOI: 10.1109/APCASE.2015.48.

[29] W. W. Ng, D. S. Yeung, M. Firth, E. C. Tsang, and X.-Z. Wang. Feature selection using localized generalization error for supervised classification problems using rbfnn. *Pattern Recognition*, 41(12):3706–3719, 2008.

[30] B. Nowak, R. Nowicki, M. Woźniak, and C. Napoli. Multi-class nearest neighbour classifier for incomplete data handling. In *Artificial Intelligence and Soft Computing*, volume 9119 of *Lecture Notes in Computer Science*, pages 469–480. Springer International Publishing, 2015. DOI: 10.1007/978-3-319-19324-3_42.

[31] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda. Decision

trees for mining data streams based on the gaussian approximation. *IEEE Trans. Knowl. Data Eng.*, 26(1):108–119, 2014. DOI: 10.1109/TKDE.2013.34.

[32] J. Starzyk, J. Graham, P. Raif, and A. Tan. Motivated learning for the development of autonomous systems. *Cognitive Systems Research*, 14(1):10–25, 2012. DOI: 10.1016/j.cogsys.2010.12.009.

[33] E. Tramontana. Automatically characterising components with concerns and reducing tangling. In *Proceedings of IEEE Computer Software and Applications Conference (COMPSAC) Workshop QUORS*, pages 499–504, Kyoto, Japan, July 2013.

[34] Y.-f. Wong. Clustering data by melting. *Neural computation*, 5(1):89–104, 1993.

[35] M. Woźniak, D. Połap, G. Borowik, and C. Napoli. A first attempt to cloud-based user verification in distributed system. In *Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*, pages 226–231, 14-16 July, Quito, Ecuador, 2015. IEEE. DOI: 10.1109/APCASE.2015.47.

[36] M. Woźniak, D. Połap, M. Gabryel, R. Nowicki, C. Napoli, and E. Tramontana. Can we process 2d images using artificial bee colony? In *Artificial Intelligence and Soft Computing*, volume 9119 of *Lecture Notes in Computer Science*, pages 660–671. Springer International Publishing, 2015. DOI: 10.1007/978-3-319-19324-3_59.

[37] M. Woźniak, D. Połap, R. K. Nowicki, C. Napoli, G. Pappalardo, and E. Tramontana. Novel approach toward medical signals classifier. In *IEEE IJCNN 2015 - 2015 IEEE International Joint Conference on Neural Networks, Proceedings*, pages 1924–1930, 12-17 July, Killarney, Ireland, 2015. IEEE. DOI: 10.1109/IJCNN.2015.7280556.

[38] W. Zhao, D.-S. Huang, and L. Guo. Optimizing radial basis probabilistic neural networks using recursive orthogonal least squares algorithms combined with micro-genetic algorithms. In *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 2277–2282, 2003.

[39] J. Mandziuk, C. Najman. UCT-Based Approach to Capacitated Vehicle Routing Problem. In *Artificial Intelligence and Soft Computing*, volume 9120 of *Lecture Notes in Computer Science*, pages 679–690. Springer International Publishing, 2015.