A Kernel Methods Approach to Reducing Handover Occurrences within LTE

Neil Sinclair, David Harle, Ian A. Glover, and Robert C. Atkinson
Department of EEE
University of Strathclyde
Glasgow, UK
Email: neil.sinclair@strath.ac.uk, {d.harle, ian.glover, r.atkinson}@eee.strath.ac.uk

***Abstract*— This paper presents a kernel-based approach to indoor-outdoor handover management for 4G femtocells. It is a necessary but difficult task to perform seamless handover from indoor femtocells to outdoor macrocells whilst maintaining call continuity. This paper describes a machine learning algorithm to operate in conjunction with 4G handover triggering mechanisms to reduce the rate of unnecessary handovers between femtocells and macrocells. The results of this algorithm show that handovers can be reduced by 65% by detecting where unnecessary handovers are likely to occur and minimising them. By reducing the number of unnecessary handovers, the system resources efficiency may be improved as a result of the potential reduction in signalling exchange taking place which in turn reduces bandwidth usage, the power used by both the femtocell and the mobile terminal and, the level of signal processing necessary.**

I. Introduction

As a consequence of the rapid uptake of smart phones, demand for Internet access on mobile handsets continues to increase towards what has been termed the “data explosion” [1]. Furthermore, studies have shown that 70% of all voice and data traffic is attributable to users located indoors [2]. However, due to the high penetration loss of exterior walls, they often experience relatively poor service quality, limiting them to low bit-rate connections. Since high-bit rate services are in greater demand by users located indoors, femtocells introduce a convenient means of providing high data-rates to those subscribers by relieving some of the strain on the macrocell layer. Cellular data provision is an area of increasing concern for operators due to the rapid uptake of smart phones, and femtocells represent a significant technology in addressing these capacity concerns. The addition of femtocells increases the complexity of the network architecture and increases the requirement for self management capabilities.

Today’s mobile networks need to be frequently re-parameterised in order to accommodate upgrades to coverage and changing traffic loads. Planning, deployment, configuration and optimisation of these network parameters requires significant expenditure from network operators as a result of the time and expertise required to maintain the network. The error prone manual tuning process may also result in non-optimal performance of the network due to the inherent delays associated with changing parameters in the entire network. This has resulted in an industrial pull from operators to lower both the CAPital EXpenditure (CAPEX) and the OPerational EXpenditure (OPEX) of their networks by introducing a degree of self management. Within LTE, the self management functionality is referred to as a Self Organizing Network (SON) [3] [4], a multi-faceted term that encompasses self-configuration, self-healing, self-protection and self optimisation. SON offers LTE a plug-n-play functionality that allows both macrocells (eNodeBs) and femtocells (HeNodeBs) to first be deployed and then autonomically self-configure to the requirements of the network. A SON allows tuning of a network to be completed in a timely manner with minimum human interaction. Moreover, a SON can be deployed to optimize handover performance between neighbouring base stations [5] [6] [7], including femtocells.

A key advantage of LTE femtocells over competitor technologies, such as Wi-Fi, is the ability to support high quality voice traffic. The ability to support seamless handover (and hence retain high voice quality) between indoor and outdoor coverage areas represents a unique selling point of 4G technologies. In order to exploit the potential advantages offered, a handover mechanism has to be adopted that provides a delicate balance between being too timid or too aggressive. A mechanism that is too timid may result in calls being dropped due to the signal strength from the femtocell dropping below the minimum level required sustain the connection before handover has been completed. A mechanism that is too aggressive may result in unnecessary handovers. Unnecessary handovers place additional demands on the network: through consumption of radio channels (Random Access Channel) and fixed links; through additional processing load in admission control, bearer setting and path switching; and have the potential to degrade the Quality of Service (QoS) of ongoing connections [8].

Consider an active and mobile user within an indoor environment. When the mobile terminal approaches, and passes through, an exterior door it will detect an increase in the Reference Signal Received Power (RSRP) [9] from an externally located macrocell. As a consequence, a measurement report will be transmitted from the mobile terminal to the femtocell base station, informing the femtocell that another base station has been detected and is a candidate for handover. The femtocell will use the measurement report to initiate handover to the macrocell if required. Now, consider the situation where an active mobile terminal approaches a large window (with low penetration loss). The increase in RSRP from the macrocell will cause a measurement report to be transmitted from the mobile terminal to the femtocell and subsequently invoke a handover, as in the previous example. However, as the mobile terminal continues to move past the window, the relatively high received signal level from the macrocell is likely to decline rapidly and thus trigger another measurement report form the mobile terminal to the macrocell, indicating that a better signal can be obtained from the femtocell. This will invoke a second handover, in quick succession, from the macrocell back to the femtocell. Clearly, the second example represents a scenario where unnecessary handover has occurred. The aim of the algorithm presented in this paper is to identify indoor regions where handover to external base stations should be permitted and regions where handover should be suppressed. Three principal regions are defined:

1) Areas of low signal strength from the macrocell. In these regions a measurement report will not be generated and therefore the proposed algorithm need not consider them. For this reason these areas can be regarded as 'null zones'.
2) Areas of high signal strength from the macrocell, and where few unnecessary handovers occur. These regions are referred to as 'permissive zones' since handover to the external base station will be beneficial. It is believed that these zones will coincide with architectural features such as exterior doors.
3) Areas of high signal strength from the macrocell, and where many unnecessary handovers occur. These regions are referred to as 'prohibition zones' since handover to the external base station should be suppressed because it is likely that a second handover (in the opposite direction) will soon follow. These regions will be consistent with architectural features such as windows and large glass exterior walls.

Within LTE, there are a number of parameters that can be tuned to optimise handover [10], among them Time To Trigger (TTT) and Handover Hysteresis are of most interest when optimising the handover process. However, tuning these parameters can be challenging as changes can incur adverse effects. Increasing these parameters reduces the likelihood of unnecessary handovers but also increases the probability of dropped calls; decreasing the parameters has the opposite effect. The scenario considered within this paper could result in the Hysteresis and TTT increasing for every unnecessary handover to such an extent that an active call will be dropped when it genuinely requires handover.

The problem under investigation in this paper is how to facilitate handover to the macrocell layer in a timely fashion whilst minimising unnecessary handovers. Reducing the number of unnecessary handovers increases the energy efficiency of the femtocell resulting from lower signalling within the network and more efficient use of the network resources. To facilitate an improved handover algorithm, positional information is incorporated in order to optimise the handover decision locally and minimise any adverse effects of parameter alterations (for an entire cell). For clarity, it should be noted that the positional information used in this algorithm is the location of regions within the radio environment in which handover occurs and not the true physical location of the user. However, there may be strong correlation between both of these forms of location.

The remainder of this paper is structured as follows. Section II provides a brief overview of Autonomics; Section III provides a brief explanation of the kernel Self Organizing Map (SOM) and the k-means algorithm used to inhibit handover; Section IV evaluates the performance of the proposed handover inhibition algorithm; and finally Section V provides Conclusions and Future Work.

II. Autonomic Networking and Self Organizing Networks

The term Autonomic Computing was coined by IBM to deliberately invite comparison with the autonomic nature of biological systems [11]. The concept has been extended beyond computing to the domain of computer/communications networking [12]. The motivation for autonomic approaches is derived from the need to manage complexity in large systems. The autonomic paradigm is one in which time-consuming and error-prone tasks are undertaken by self managing components, leaving human administrators free to concentrate on high-level policies. The policies specify desired system outputs or behaviour but not how these are to be achieved; that is the role of the autonomic element. In many cases, the high-level policy will specify utility functions, for example, minimising dropped calls during handover.

The basic building block of an autonomic system is a control loop consisting of the following phases: Monitor, Analyse, Plan, and Execute.

- The Monitor phase is concerned with the acquisition, collation and filtering of data concerning the managed element or its environment. Within the context of this paper, the Monitor phase collects user location and measurement reports.
- The Analyse phase examines the data and makes a decision on potential actions to be taken to optimise the performance of the system or take corrective action. In this work the kernel SOM, as explained in Section III, performs this task.
- The Plan phase uses the conclusions of the Analyse phase to determine which specific actions should be taken to reconfigure or optimise the managed element. In the example of handover optimisation, the Plan phase would specify which parameters to adjust and the degree to which they should be adjusted as well as determining if handover should take place.
- The Execute phase translates the planned actions into a sequence of technology-specific commands.

Autonomic computing identifies four key facets where complexity management is necessary: self-configuration, self-optimisation, self-healing, and self-protection. This Autonomic paradigm has been embraced by the 3GPP where its principles provide the foundation for SON. When translated into autonomic networking, these facets are subtly different but have broadly similar objectives. The work described in this paper is concerned with the self-optimisation of handover.

Specifically, the autonomic system will monitor when unnecessary handovers between the femtocell and macrocell occur and seek to reduce their number over time. However, the system should not inhibit seamless handover from the femtocell to the macrocell when genuinely required. The 3GPP have defined two metrics that capture ineffective handover timing: handover-too-late (an indicator for call dropping) and handover-too-early (an indicator of unnecessary handover) [13]. A system that is too timid will lead to an increase by handover-too-late occurrences due to lack of responsiveness and a system that is too aggressive will lead to an increase in handover-too-early occurrences.

The problem is complicated by the fact that every building has a unique radio environment which is a function of: the femtocell base station placement, the architecture of the building (including building materials), and the number and location of external macrocell base stations. Therefore, each building will have a unique topography of permissive and prohibition zones. Given the economic drivers for autonomic approaches, the femtocell base station cannot be assumed to be pre-programmed with this information in advance. Rather, the femtocell should be able to configure and optimise its performance with experience. In terms of the types of machine learning approaches that can be applied to this problem, supervised learning strategies are therefore not applicable.

The LTE handover parameters Handover Hysteresis and Time-To-Trigger are central to the SON concept of self-optimisation. They can be tuned to realise the optimum crossover point for a particular geographical region. Inherent in the design philosophy is the notion that the mobile user follows a consistent trajectory moving from one external cell site to another. In reality, this is a simplification since the user will have a somewhat more random motion. Furthermore, the fading characteristics of the wireless link gives rise to a non-monotonic decay of signal strength. Nonetheless, it is believed that these parameters should be sufficient to optimise handover between two neighbouring macrocells. However, in the case of indoor-outdoor handover such parameters will not be sufficient for optimisation since they cannot capture whether handover is actually required, i.e. whenever the mobile terminal is located within a permissive or prohibition zone it may not be possible to infer a user's intent to leave the building or not. The parameters must be augmented by positional information that can be used to infer such intent. After all, it would be counter-productive to successively increase these parameters whenever an unnecessary handover occurred (a user being close to a window) because this would lead to an increase in handover failure probability when the user attempts to exit the building.

For a building of arbitrary shape and construction, an algorithm is required that can optimise handover performance. To realise such an objective, the direction finding capability of MIMO systems is exploited to provide a profile of locations (or more correctly regions in the radio environment) where handover is genuinely required (permissive zones) and those where unnecessary handovers are likely to occur (prohibition zones). The kernel Self Organizing Map is particularly useful in this context by continually mapping regions where either successful or unnecessary handover has occurred and using this information to identify the periphery of the permissive and prohibition zones as explained in Section III.

III. Kernel Self Organizing Feature Map using k-means

Within this work, a kernel Self Organizing (Feature) Map (SOM) with k-means is used to provide the femtocell with a profile of locations in which handover may take place. The location of the user when a handover trigger is transmitted (The Monitor phase of the autonomic system) is used as the input to the SOM (The Analysis phase of the autonomic system) to allow the femtocell to learn which locations within the radio environment correspond to permissive and prohibition zones (The Plan phase of the autonomic system). Handover can then be prohibited or allowed within the LTE network (The Execute phase of the autonomic system). This learning is completed in a group based manner to allow faster convergence of the neurons within the network. The convergence of the neurons into accurate locations minimises the error inherent in the vector quantisation based algorithm, SOM.

The Kohonen SOM [14] is a form of artificial neural network where neurons are arranged in either a one or two dimensional lattice. Generally, a two dimensional lattice is preferred; the case in this work. The lattice can be regarded as a special case of a feed forward neural network with a single computation layer that is arranged into logical rows and columns. Within the SOM, all neurons are connected to all inputs and, unlike other neural networks, the neurons have no activation function. Each input to the system is compared to all neurons and the closest neuron is then used as the basis for the neurons to learn. These neurons can also hold information about previous handovers within the radio environment. The kernel SOM algorithm is a version of the SOM algorithm that non-linearly transforms the data into a feature space. Once transformed the distances between the weights and the inputs can be calculated non-linearly.

Kernel methods are a class of algorithms that are used for pattern/cluster analysis. These methods perform non-linear mappings of input data into a higher dimensional feature space and then perform linear operations on the data. Each coordinate from the input space maps to an element within the feature space; transforming the data into a set of points in a Euclidean space. Once the data has been converted into the feature space, the inner product used within clustering methods can be replaced with the kernel trick [15].

The kernel trick allows for the computation of a dot product in a high dimensional feature space using simple functions defined on pairs of input patterns. This allows for a non-linear mapping to the feature space which gives more detail at the points of interest (this lowers the vector quantisation error). The mapping of $\mathbf{x}$ to $\phi((x))$ can be implicitly carried out with no knowledge of ϕ. This means that only knowledge of the inputs, the weights and the kernel function $(K(\cdot,\cdot))$ is required. By using the kernel trick rather than Euclidean distance, the resulting reduction in the vector quantisation error increases the convergence rate of the network. The kernel Self Organizing Map can be regarded as a kernel methods modification of Kohonen's equivalent.

The kernel SOM [16] [17] [18] is particularly useful for detecting clusters within data. Here, we use it to determine the areas of the permissive and prohibition zones based on an estimate of distance (received signal strength) and angle of arrival at the femtocell base station. The kernel SOM algorithm has an input space that is highly multidimensional, a weight space of the same dimension as the input and an output space of smaller dimension than the input. The kernel SOM used follows closely to that described by MacDonald [16].

The kernel SOM utilised in this study incorporates k-Means as a method of cluster analysis: k-Means partitions a set of data into a predefined number of clusters, k. Each cluster is allocated a centroid that is the mean value of the data in the cluster, resulting in the data being partitioned into Voronoi cells. Such a process can be used within the kernel Self Organizing (feature) Maps algorithm to improve, in comparison with the Kohonen SOM, the convergence time and the system's accuracy by only updating the relevant weights within the neural network.

The kernel SOM algorithm consists of four phases describing the learning process: initialisation, competition, cooperation, and synaptic adaptation. There is, however, an additional step added to this algorithm in order to complete k-means.

A. Initialisation

Initialisation of the SOM network presets the individual weight values of each neuron in the lattice to values drawn from a uniform distribution. The initial weight values for this work will be distributed within the propagation region of the femtocell. Each input will be associated with a weight within the high dimensional feature space. This represents the geographical location obtained using the RSRP and angle of arrival from the mobile terminal at the time the measurement report was generated.

Both the weights and the inputs must be converted from the input space to a feature space. This conversion takes place using a kernel function that allows more detail at the points of interest (this reduces the vector quantisation error). The distance between the input and the weights can be determined by many methods; in this case, using the kernel trick. The weights will converge to the areas that the inputs occur, i.e., the locations of the handover triggers.

B. Competition

The next step of the process is for inputs to be applied to the system. Under operational conditions this would occur every time a mobile terminal generates a measurement report (triggered by detection of a base station other than the serving base station). Since each input is connected to each neuron, the input and weight vectors have the same dimensions. The representation for an a-dimensional input is defined in Equation (1) and the weight vector associated with each neuron in the lattice is defined in Equation (2).

$$\mathbf{x} = [x_1, x_2, ..., x_a]^T,\ \mathbf{x} \in \mathbb{R}^a \tag{1}$$

$$\mathbf{w}_j = [w_{j1}, w_{j2}, ..., w_{ja}]^T,\ j = 1, 2, ..., l,\ \mathbf{w}_j \in \mathbb{R}^a \tag{2}$$

Here, l is the total number of weights in the network. Given that there is no activation function, the output of each neuron will be a combination of both the input and weight vectors. From a geometrical perspective, the winning neuron in SOM is calculated based on Euclidean distance using the dot product, therefore, the lower the value, the closer the weight vector is to the input vector. Within the kernel SOM, Euclidean distance is replaced with the kernel trick. There is a competitive aspect of this algorithm in that the neuron whose weight vector provides the best match to the input vector will be selected as being the winning neuron. If the index of the winning neuron is denoted by $i(\mathbf{x})$ within the lattice $\mathcal{L}$ (denoting the grid of neurons in the weight space) then the winner is given by Equation (3).

$$i(\mathbf{x}) = \arg\min_j \|\mathbf{x} - \mathbf{w}_j\|,\ j \in \mathcal{L} \tag{3}$$

The distance can also be written in terms of the kernel function, as is shown in Equation (4). The mapping to the feature space is completed using a kernel such that $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ where $\phi(\mathbf{x})$ is the function that maps the data onto the feature space.

$$\begin{aligned} \|\mathbf{x} - \mathbf{w}_j\|^2 &= \|\phi(\mathbf{x}) - \phi(\mathbf{w}_j)\|^2 \\ &= K(\mathbf{x}, \mathbf{x}) + K(\mathbf{w}_j, \mathbf{w}_j) - 2K(\mathbf{x}, \mathbf{w}_j) \end{aligned} \tag{4}$$

A Gaussian Kernel function is used as shown in Equation (5).

$$K(\mathbf{x}, \mathbf{w}_j) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{w}_j\|}{2\sigma^2}\right) \tag{5}$$

Once the winner, the closest match to the input, has been selected as it can be utilised by the cooperation stage of the algorithm.

C. k-means

Once the weights within the network have been initialised they can be partitioned into k clusters and the k-means algorithm can be completed. The number of clusters k, in this case, is the combined number of both permissive and prohibition zones as dictated by the architecture of the building.

If this number of clusters is incorrect, the algorithm may lose accuracy but the weights within the network will still converge towards a suboptimal solution that is an improvement over the standard SOM algorithm. This is illustrated in Figure 4. The k-means algorithm is executed using the following steps:

1) The partitioning is commenced by allocating k centroids randomly within the area of the area of the network.
2) Each weight can then be allocated to its nearest centroid location using Equation (6) where m denotes the centroid, c the index and $q(\mathbf{w})$ the index of the winning centroid.

$$q(\mathbf{w}) = \arg\max_{c} \|\mathbf{w}_j - \mathbf{m}_c\|,\ j \in \mathcal{L},\ c \in [0, k] \quad (6)$$

This results in the generation of Voronoi cells.

3) After each weight has been allocated to its corresponding centroid, the centroid can be updated using Equation (7).

$$\mathbf{m}_c = \frac{1}{N}\sum_{j=1}^{N} \mathbf{w}_j \quad (7)$$

Each centroid location ($\mathbf{m}_c$) is the mean value of all the weights allocated to it. N is the number of weights allocated to mean c.

4) Steps 2 and 3 are repeated until convergence of the centroid and it's allocated weights have been achieved.

D. Cooperation

Once the winner for a given input vector has been selected and the weight has been assigned to its closest centroid, the weights of the neurons within the winner's sphere of influence are updated if they are linked to the same centroid as the winner. This constitutes a cooperative learning process since, unlike other competitive learning strategies, it is not just the winning neuron that has its weight values modified. This group learning strategy permits the network to converge more rapidly and accurately compared to the case where only the winner would modify its weights. Adding k-means into this algorithm allows only the weights that are in the same cluster as the winner to be updated which improves the accuracy of the weight locations.

The sphere of influence is governed by a neighbourhood function which determines how many of the winner's neighbours can undergo learning, and also the degree to which they will learn. Within the sphere of influence, neighbours closer to the winning neuron will have their weights updated by a greater amount than those located further away. In order to achieve this a distance metric between neurons in the lattice is required, where the distance between two neurons e and f is given by Equation (8).

$$d_{f,e} = \|\phi(\mathbf{r}_f) - \phi(\mathbf{r}_e)\| \quad (8)$$

$\mathbf{r}_e$ and $\mathbf{r}_f$ represent the locations of neurons e and f in the lattice respectively.

The neighbourhood function should decay monotonically with distance from the winner. Furthermore, the function should be maximum at the winner ($d_{f,e} = 0$) and decay to zero as $d_{f,e} \to \infty$. A popular choice for the neighbourhood function which satisfies these requirements is the Gaussian function as shown in Equation (9), and this is the function adopted in this work.

$$h_{f,e} = \exp\left(-\frac{d_{f,e}^2}{2\sigma^2}\right),\ f \in \mathcal{L} \quad (9)$$

The parameter σ defines the width of the Gaussian function and, in essence, σ determines the size of the sphere of influence around the winning neuron. When using a kernel SOM, the size of the sphere of influence (i.e. σ) is reduced over time; in practice this translates to number of iterations, n. The width of the neighbourhood function can be made to decay with time by making σ decay with time. In this work an exponential decay is assigned to σ as described in Equation (10):

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right) \quad (10)$$

Here, n is the iteration number, σ_0 is the initial value and τ_1 is a temporal decay time constant chosen by the designer. By incorporating temporal decay, Equation (9) can now be re-written in the form in Equation (11).

$$h_{f,e}(n) = \exp\left(-\frac{d_{f,e}^2}{2\sigma^2(n)}\right),\ j \in \mathcal{L} \quad (11)$$

E. Synaptic Adaptation

The adaption process is concerned with the execution of the weight update procedure for all neurons within the sphere of influence of the winner. Unfortunately, standard Hebbian learning is not suitable for this type of network because repeated reinforcement (positive increase) of weights within a particular sphere of influence will cause the neurons contained therein to be driven to a state of saturation. Thus a so-called forgetting term is used to augment the update algorithm such that it progressively decreases the strength of the weights; the forgetting term is given by Equation (12):

$$\mathbf{u}(y_j) = g(y_j)\mathbf{w}_j \quad (12)$$

Generally $g(y_j)$ is a positive scalar function of neuron j's output. An appropriate choice for this function is given by Equation (13):

$$g(y_j) = \eta y_j \quad (13)$$

Parameter η is the learning rate. In practice the learning rate also decays with time (or iterations); therefore, it is an exponentially decreasing function as shown in Equation (14):

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right) \quad (14)$$

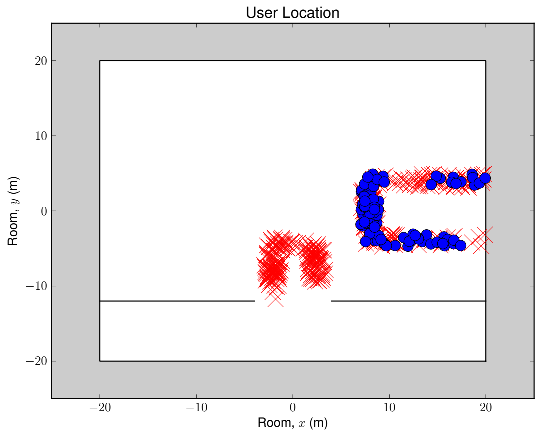

Fig. 1. Handover Locations: requests 1 to 100

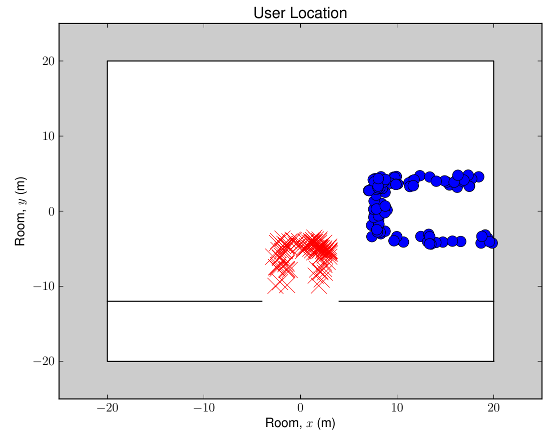

Fig. 2. Handover Locations: requests 301 to 400

η_0 is the initial value and τ_2 is a second time constant. The augmented Hebbian weight update equation can be written as shown in Equation (15):

$$\Delta \mathbf{w}_j = \eta y_j \mathbf{x} - \eta y_j \mathbf{w}_j \tag{15}$$

By setting $y_j = h_{j,i(\mathbf{x})}$ the weight update equation can be written as shown in Equation (16).

$$\Delta \mathbf{w}_j = \eta h_{j,i(\mathbf{x})} \left(\mathbf{x} - \mathbf{w}_j \right) \tag{16}$$

Thus, the weights for neuron j within the sphere of influence of the winner are updated iteratively according to the rule given by Equation (17).

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n) h_{j,i(\mathbf{x})}(n) \left(\mathbf{x}(n) - \mathbf{w}_j(n) \right) \tag{17}$$

When the neurons have been continuously updated over a period of time the locations of the neurons will converge to optimal location due to the learning rate becoming very low and the neighbourhood no longer updating any nodes other than the winner. Once this happens, the locations of both permissive and prohibition zones have been identified.

IV. Simulation Model and Results

The performance of the algorithm described in Section III has been evaluated using a model developed within the network simulator, NS3. To demonstrate the effectiveness of the kernel SOM and k-means algorithms, a scenario has been modelled that incorporates a single prohibition zone (window) and one permissive zone (external door); the simulation was used to determine the effectiveness of the proposed algorithm in identifying both types of zone. The dimensions of the area being modelled is 40m by 40m. The simulation area consists of a large room (40m by 32m) and a space to allow the user to leave the room (8m by 40m); the user must leave through the door. A random walk mobility model is used to simulate the movement of a single user (with an active mobile device), this model allows a random change of direction after a defined period of time, in this case every second, with the user moving at speeds of 2 to 4m/s. The initial position of the mobile user is the centre of the simulation area. When the mobile user moves around the radio environment the propagation characteristics, perceived by the mobile device, change and the signal strengths are updated accordingly using a distance based propagation model. Within the propagation model, both the signal strengths of a single femtocell and a single macrocell are used to determine whether there is a requirement for handover. Once the system has detected that another base station has offered a stronger signal by a fixed Hysteresis value for a prescribed period of time (TTT), handover is requested; a decision whether or not to allow or prohibit this handover then takes place using the mechanism described within this paper. As the modelled system operates using an event-based paradigm focusing on handover instances and takes a generic approach, the choice of propagation and mobility models has only a secondary effect upon the results. The choice of such models does not affect the generality of the results.

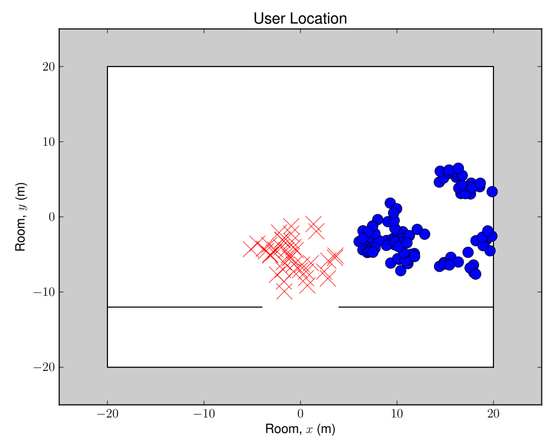

Fig. 3. Handover Locations with Error: requests 301 to 400

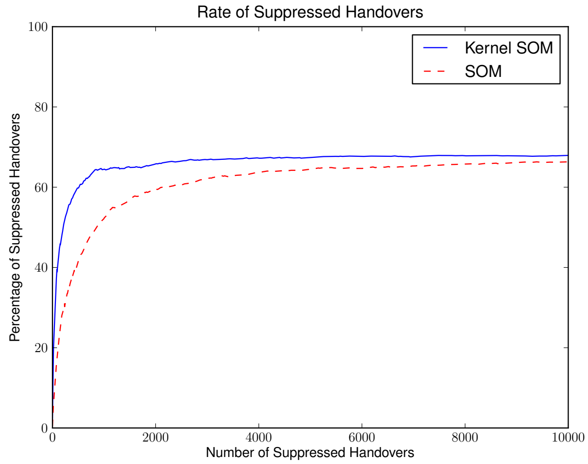

Fig. 4. Handover Suppression Rate

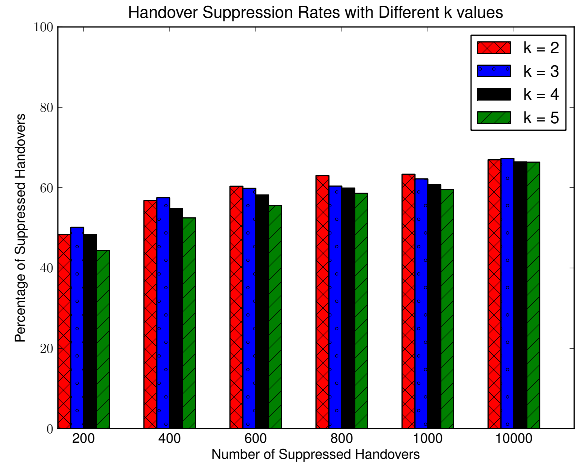

Fig. 5. Handover Suppression Rate with different k values

In an unmodified LTE system, if a mobile terminal detects a base station with a higher RSRP than the serving base station a measurement report is triggered that may initialise handover execution. Within any LTE system utilising the proposed algorithm, handover may be permitted or suppressed depending on previous experience of handovers in that area, as perceived by the femtocell, i.e. whether the mobile terminal is located in a permissive or prohibition zone. The initial setup of the femtocell allows convergence of the neurons within the SOM to the locations of the radio environment where handover may take place by using the location of the handover triggers as the input to the neural network. Each neuron within the SOM has the ability to retain knowledge of past experience with regards to handover. A decision as to whether or not the mobile user is within a permissive or prohibition zone is then made based on the neuron's experience of its area of the radio environment; the decision to permit or prohibit the handover request is then made.

Examples of handover locations are depicted in Figures 1 to 3. In these figures: the (black) lines denote the walls that are the lower bound of the room; a permitted handover is depicted by a (red) 'X', and a suppressed handover by a (blue) 'O'. It should be noted that the handover triggers occur around the regions where the macrocell RSRP is greater than the femtocell RSRP; this can be either macrocell to femtocell handover or femtocell to macrocell handover. These figures show that the neurons have quickly clustered to the areas that represent a permissive zone and a prohibition zone. Permissive zones are detected by successful handovers and prohibition zones are detected by handover ping-pong occurences being detected. Figures 1 and 3 depict that the system detects where handovers occur; these handover triggers can occur anywhere within the region of permissive and prohibition zones, not just at the periphery. The location of the neurons constitutes the analysis phase of the autonomous system but the decision whether this is a permissive or prohibition zone is determined in the Plan phase and implemented in the Execute phase.

The ability of the algorithm to learn from experience within a perfect system is depicted in Figures 1 and 2, which provide two snapshots of performance during two time intervals. Figure 1 shows the first 100 handover requests; it is evident that all handovers are permitted (red 'X') in the permissive zone, but within the prohibition zone there is a mixture of permitted handovers (red 'X') and suppressed handovers (blue 'O'); this is consistent with the nonrestrictive nature of the algorithm at initialisation. However, as the system progresses and the algorithm has the opportunity to gain more experience it becomes more restrictive; Figure 2 demonstrates that handovers are executed successfully only within the permissive zone and suppressed only within the prohibition zone.

However, indoor locations are inherently complex radio environments due to clutter and other obstacles. This gives rise to position estimation error. Figure 3 shows that including a location estimation error of up to 3m within the simulation, the area of the permissive and inhibition zones can still be determined. The learning rate and the accuracy of the system are unaffected by the addition of error due to both the x and y elements of the error being of uniform distribution. The movement of the neurons (due to error), during learning will, in effect, cancel each other out.

Since the machine learning algorithm proposed in this paper makes decisions based on experience, there is a learning curve involved. The system is initially nonrestrictive: it does not attempt to suppress any handovers. As the mobile terminal continues to move around the indoor area the algorithm learns the locations that generate unnecessary handovers. Once many unnecessary handovers occur within any area, handover is prohibited.

The total number of handovers can then be reduced over time by suppressing handover in prohibition zones. It should be noted that the algorithm is not given any prior information regarding the location of the prohibition or permissive zones. This knowledge is gained through an unsupervised learning approach. Once a handover trigger has been paired to its closest neuron, learning can occur to optimise handover performance. k-means (described in Section III) is used to aid the process of the kernel SOM algorithm (described in Section III) to increase the rate of convergence which reduces the learning time of the system. It should be noted that the system starts to reduce the number of handovers quickly and that any reduction in the number of handovers represents an improvement in system performance quality.

Figure 4 shows the rate at which handovers are suppressed using both a (Kohonen) SOM algorithm and the kernel SOM algorithm proposed within this paper. The graphs within this figure illustrate the nature of the learning curves inherent in both algorithms. The graphs were generated using 30 parallel simulation runs to provide a statistical time average.

As can be seen by comparing both the SOM algorithm and the kernel SOM algorithm in Figure 4, the kernel SOM provides an improved performance because its convergence time is faster than the original SOM. The improved performance of the algorithm is due to the change in distance metric (now kernel trick and not Euclidean distance) and the addition of k-means into the weight updating process.

When using k-means, the value of k must be provided in order for the system to split the neurons into the appropriate number of Voronoi cells; in this case k should be 2. However, when using a non-ideal value for k, the proposed system will still represent an overall improvement over the unmodified LTE system but the convergence rate may be sub-optimum. Figure 5 shows different values of k at different numbers of handover suppressions and offers a comparison of these k values. It shows that with k as 2, 3, 4 or 5, the overall performance is not significantly different and the percentage of suppressed handovers converges to close to a common value. Thus, the use of k-means is valid within plug-n-play functionality of SON within the indoor environment.

The system is improved by reducing the overall number of handovers that occur which, in effect, increases the effective overall capacity of the system by reducing the level of network resources required. By modifying the LTE system using the proposed mechanism, the number of unnecessary handovers is significantly reduced and, when using the kernel SOM rather than the (Kohonen) SOM algorithm, there is an increase in the learning rate making the system reduce unnecessary handovers more proficiently. Figure 4 shows that the proposed algorithm offers a reduction in handovers of nearly 65%.

V. Conclusions and Future Work

In this paper, an effective algorithm to reduce unnecessary handovers in an indoor-outdoor scenario has been proposed. This self-optimising algorithm uses kernel methods and neural networks in order to improve handover efficiency while retaining the required plug-n-play functionality of SON in LTE systems. By monitoring the location of the user when a handover trigger is made, the Kernel SOM algorithm can be used to analyse the situation and decide whether the mobile user is within a zone that handover should be permitted or prohibited. By implementing the proposed algorithm, the system can reduce the number of handovers that occur by about 65% by minimising the number of unnecessary handovers. Within this work, the assumption is made that a mobile user generally will walk past a window and through a door but the femtocell is given no prior knowledge as to where these locations are within the environment. In a situation where the system has incomplete knowledge about the number of permissive and prohibition zones, the algorithm is still an improvement over a typical LTE system. It may be possible to propose values for k by a cursory survey of the indoor area by noting the number of doors and windows. Future work will be to improve the k-means algorithm. Currently, the system works more efficiently when the correct value of k is known. A system will be devised that autonomically determines the best value for k and therefore, the number of permissive and prohibition zones within the area of the femtocell.

References

[1] R. Thanawala, M. El-Sayed, T. Morawski, A. Mukhopadhyay, J. Zhao, and C. Urrutia-Valdes, "The mobile data explosion and new approaches to network planning and monetization," in *Telecommunications Network Strategy and Planning Symposium (NETWORKS), 2010 14th International*, sept. 2010, pp. 1 –4.

[2] Steven Brightfield, "Femtocells - How to Get the Most Out of Your Nework," 4G World Presentation, October 2011.

[3] 3GPP TS 32.500 V8.0.0, "Technical Specification Group Services and System Aspects; Telecommunications Management; Self Organizing Networks; Concepts and requirements (Release 8)," December 2008.

[4] White Paper, "The Benefits of SON in LTE," Tech. Rep., 2009.

[5] T. Jansen, I. Balan, J. Turk, I. Moerman, and T. K andrner, "Handover parameter optimization in lte self-organizing networks," in *Vehicular Technology Conference Fall, 2010 IEEE 72nd*, sept. 2010, pp. 1 –5.

[6] K. Kitagawa, T. Komine, T. Yamamoto, and S. Konishi, "A handover optimization algorithm with mobility robustness for lte systems," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*, sept. 2011, pp. 1647 –1651.

[7] L. Ewe and H. Bakker, "Base station distributed handover optimization in lte self-organizing networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*, sept. 2011, pp. 243 –247.

[8] J. Alonso-Rubio, "Self-optimization for handover oscillation control in lte," in *Proc. IEEE Network Operations and Management Symposium (NOMS)*, Osaka, Japan, April 2010, pp. 950 – 953.

[9] 3GPP TS 36.214 V8.5.0, "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer - Measurements (Release 8)," December 2008.

[10] 3GPP TS 36.331 V9.3, "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (Release 9)," June 2010.

[11] P. Horn, "Autonomic computing: IBM's perspective on the state of information technology," Technical report, IBM corporation, 2001.

[12] J. Strassner, *Cognitive Networks: Towards Self-Aware Networks*, chapter 2: The Role of Autonomic Networking in Cognitive Networks, pp. 23 – 52, Wiley, first edition, 2007.

[13] ETSI TS 132.521 V9.0.0, "Digital cellular telecommunications system (Phase 2+); UMTS; Telecommunications Management; SON; Policy NRM; IRP; Requirements," April 2010.

[14] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[15] Bernhard Schlkopf, "The kernel trick for distances," in *TR MSR 2000-51, Microsoft Research*, 1993, pp. 5–3.

[16] D. MacDonald and C. Fyfe, "The kernel self-organising map," in *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, 2000.

[17] Manuel Martn-Merino and A Munoz, *Extending the SOM Algorithm to Non-Euclidean Distances via the Kernel Trick*, pp. 150–157, 2004.

[18] S. Haykin, *Neural Networks and Learning Machines*, chapter 9: Self-Organizing Maps, pp. 453 – 502, Pearson, third edition, 2009.