

平成21年度 特別研究報告書

センサーネットワークを用いた  
ホームネットワークシステムの構築  
および便性の向上

龍谷大学 理工学部 情報メディア学科

学籍番号 T060601 中山 達哉

指導教員 三好 力 教授

## 内容梗概

ホームネットワークシステムは、多様なネットワーク家電を相互接続させて構成される家庭内のネットワークである。この家庭内にネットワークを構成するということは、例えば、映画鑑賞する上での利用環境をあらかじめ設定することにより、従来では環境を整えるために各家電に設定を施していた操作を単一な操作で一気に環境を整えることや、介護が必要な同居人がいる場合に家庭内に設置した WEB カメラを通して外出先からも安否の確認が可能になるなど、利用者にとって家電の利便性を向上させる効果がある。

本論文では、上記の様なホームネットワークが提供する多様なサービスの中でも特に家電が相互接続されることによるセキュリティ能力に注目する。今日、家庭でのセキュリティとは特殊なセンサーを家庭内に設置し、そのセンサーを起動させることにより不法侵入者や、火災等の煙を感知して状況に応じたセキュリティ会社に連絡するシステムである。ここで、一般家庭であることを考慮すると特殊なセキュリティ対策用のセンサーを設置している家庭は増加傾向にあるものの、まだまだ全世帯には設置されていない。

そこで、ネットワーク家電に近年多く見られるセンサー搭載家電(エアコン、照明等)を有効に利用して、センサーの設置の様な利用者に負担を掛けることなく、上記の様なセキュリティを実現させるシステムを提案する。尚且つ自己組織化マップという技術を用いたセキュリティの自己学習環境を提案する。本実験では、提案システムでシミュレーション実験を行ない、システム実現の可能性を自己組織化マップを用いて判定した。

## 目次

目次.....	3
1. はじめに.....	1
2. 基本事項.....	2
3. 既存技術の問題点と提案手法.....	7
4. 実験.....	8
5. 実験結果.....	16
6. 考察.....	20
7. 結論.....	20
謝辞.....	21
参考文献.....	22
付録.....	24

## 1. はじめに

家電は、1990年頃より家電でネットワークを構築する「ホームネットワークシステム(HNS)」という技術の開発が大手企業ベンダや政府機関によって進められている。これは、家電を相互接続することにより、家電の利用環境の効率化を図り、ネットワークを構築することにより宅内、宅外からの遠隔監視、遠隔操作など、インターネットを通じた付加価値サービスの提供を期待できるものである。また、現在いくつかの企業は既にHNSを商品化しており、一般家庭にも近い将来普及すると期待されている。

HNSの利用例としては、映画視聴に於いて行う操作で、テレビの設定、スピーカーシステムの設定、照明の設定(明るさ調整、カーテンの開閉)がある。これらの操作を「シアターモード」という一つの操作にまとめておくことにより、利用者はそのシアターモードという操作さえ行えば、上記の家電の環境設定を単一的な操作で出来るというものである。このように本来複数の工程を必要とする作業でも、利用者への負担は最小限に抑えた操作にできる。またこの内容に伴い、家電のリモコンの様なインターフェースの単一化を図ることや、音声認識・モーションセンサー等の人感センサーを用いた従来のリモコン操作から大きく負担を減らし、快適にHNSを利用出来るための開発もされている。また、ホームネットワークは現状でも振動センサーや人感センサー、熱感知センサー等を家庭内に設置すれば不審者の侵入や火災等を感知することは出来る。しかし、一般家庭に於いて、わざわざ感知センサー等を設置するのはまだ一般的とは言えない状態である。

本研究内容は、既存の家電技術を大きく変更することなく家庭内のセキュリティ能力を向上させるというものである。つまり、家電に搭載される技術の進歩により多くの家電に搭載されてきている様々なセンサー群を代用することで、上記の様なセキュリティのための特別なセンサーを設置せずとも同等の効果を期待することが出来るということである。例えば、高性能なセンサー搭載の家電であるエアコンが持つ人を感知する高性能なセンサーをエアコンの利用だけに留めず、上記のセキュリティ対策にも利用するということである。また、HNSの要でもあるサーバの起動に対するトリガーとしても上記の提案を利用することも期待できるため省エネも期待できる。

具体的な流れとしては、家電のセンサーが感知したデータをサーバで学習させ、正規の利用者の行動パターンを記憶し、この記憶されたパターンを正常・異常の判定に用いるという手法である。

そこで、この学習、判定に教師なし競合近傍学習が可能な自己組織化マップ(Self-Organizing Maps, SOM)を用いた。SOMは、コホネンによって開発された大脳皮質の神経機能をモデル化したニューラルネットワークである。また、ニューラルネットワークとは人間の脳の仕組みを模倣した情報処理モデルである。人間は、物体の色や形、香り等の特性を五感(視覚・聴覚・臭覚・味覚・触覚)で知覚し、その特性に合致する特定の物体だと認識することが出来る。このような対象物の特性から物体を判別することを「パターン認識」という。つまりパターン認識とは、ある種概念より特定の事象を導き出すということであり、SOMはこの概念獲得に対して外部から特定の分類基準(教師信号)を受け取る事無く、判断の仕組みを入力信号の学習から自己組織化的に形成を行い、その獲得した概念を元に事象を分類する特性を持つ。これが、教師なし競合近傍学習と呼ばれる技術である。

## 2. 基本事項

以下に本論文に必要な前提知識を記す。

### 2.1 ホームネットワークシステム

ホームネットワークシステム (Home Network System, HNS) は、家庭内の家電ネットワークである。HNS を構成する通信規格は ZigBee, ECHONET, Bluetooth, IEEE1394 等の短距離で送信データ量は大きく無いが、消費電力の抑えられた通信規格が適用されており、このような通信規格に対応した家電が「情報家電」と呼ばれ、HNS はこの情報家電で構成されている。

構成は、複数の情報家電とホームサーバから成り (図 1)、これらは家庭内 LAN に接続されている。また各情報家電には外部のソフトウェアから家電に対して命令・操作・制御を可能にするための API (Application Program Interface) が備えられている。HNS のサービスは、情報家電に搭載された API を決められた手順で呼び出すソフトウェアによって実現している。ホームサーバの役割は、利用者へ提供される HNS サービスのアプリケーション管理、外部ネットワークへ接続する際のゲートウェイとして働いている。[1]

HNS サービスの一例として、シアターモードサービスを挙げる。これは、テレビ・スピーカーシステム・カーテン・照明の連動動作から成り、利用者が映画館に近い環境で映像を視聴出来るように働く HNS サービスである。利用者がシアターモードサービスを指定すると、サーバから各家電に命令が行き、照明を暗くし、カーテンを閉め、テレビを起動させ、連動するスピーカーシステムも適切な音量に調節される。といった一連の動作が HNS サービスの指定以降の操作を自動的に行われるといったものである。

また、家電が持つ API はいくつかの層に分類することが可能である。アドレス設定、通信プロトコル、信号の形式等のネットワークを介して通信するための処理を行う層だとネットワーク層といった具合である。対してアプリケーション層の API は下層のミドルウェアやネットワークプロトコルの補助をして、情報家電の機能呼び出しを容易に行えるように働いている。例えば、カーテンのアプリケーション層の API として `open()`, `close()` が考えられ、HNS としてはカーテンの開閉についての命令を簡略して記述出来る。[3,4]

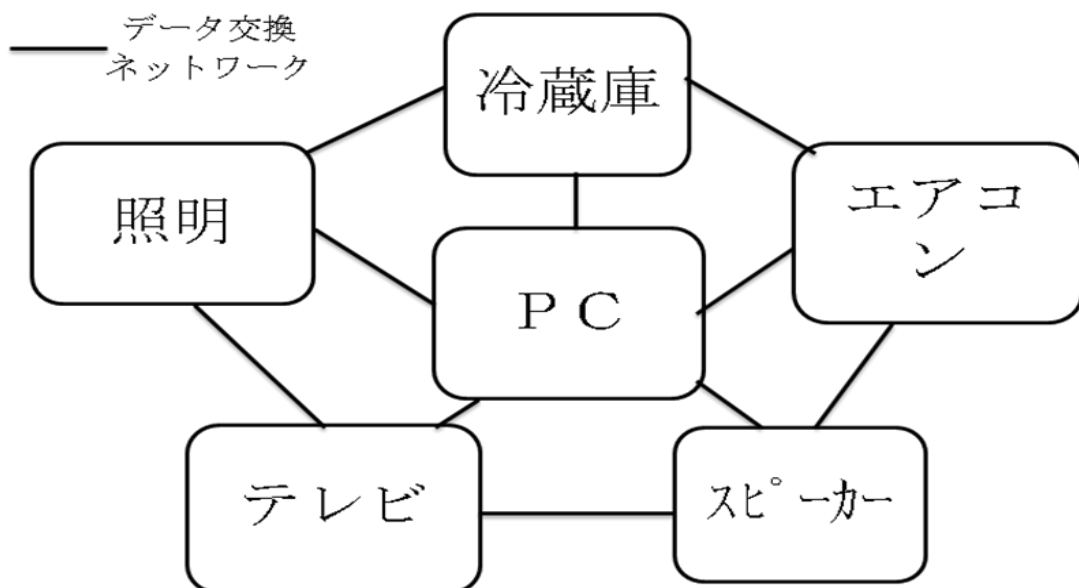


図 1. ホームネットワークの接続例

## 2.2 自己組織化マップ

自己組織化マップ (Self-Organizing Maps, SOM) は、コホネンによって提案されたニューラルネットワークの一種である「自己組織化マップ(Self-Organizing Maps, SOM)」と呼ばれる教師なし競合学習モデルである。

SOMには学習機能以外にデータを可視化する特性を持ち、特徴マップと呼ばれる表現形式でマッピングされる。特徴マップとは、データ空間のトポロジー(繋がり)を保ったまま二次元平面のデータに変換する技術である。イメージとしては立体的に浮遊する多量の球(データ)を紐(トポロジー)で相互に結んでいる状態から、紐(トポロジー)が結んでいた球同士の関係性を損なわない様に平面上に球を設置し直すというものである。この事より、多次元では認識が難しかったデータの判別も、関連性の濃いデータが近くに配置されている二次元平面的な図として表現されるので人間の視点からも直観的に判断出来る状態に出来る。この特徴マップは単純なデータの可視化と違い、各データ間の関連性を保っているため特定の特性のみ可視化する等、多次元データの要素毎の比較も容易に行う事も出来る。

これらの特性を元に SOM に独特な特性として、アソシエーション(連想)、リコール(想起)と呼ばれるものがある。これは不完全な情報から柔軟に判断を行うのに適した能力である。

先ず、アソシエーションとは、特徴マップが作成された時に使用されなかった新しい特性(属性)を既存のマップに関連づけるものである。この時、既存マップに用いられた全てのデータと新しい属性に関係がある場合、より正確なパターン分けを実現する。また一部のデータのみ関係のある属性だとしても少なからずパターンの推定に役立つことがある。

次に、リコールとは、作成された特徴マップに新しいデータを入力することでマップ内に分類された属性に入力データを対応させるというものである。この事により、例えば、顧客の特徴・購入物で作成されたマップに新しい顧客のデータを入力することでどのような商品を案内すべきかを自動で判断するシステム等が実現できる。

SOM の特徴である多次元情報を二次元空間で表示、可視化することができ、また、SOM で用いられるネットワークは各ユニットが二次元上に配置されて形成されている。ここで、出力ユニット  $i$  は入力ベクトル  $x$  と同次元の内部ベクトル  $m_i(t)$  の関係を持つ。入力ベクトルは高次元のものが一般的であり、高次元空間から二次元空間への射影を行うためにこのようなネットワークが用いられる。この関係を式に起こすと次のようになる。

時刻  $t$  に、ネットワーク上のすべてのユニットに同じ入力ベクトル  $x$  が入力された時、それに対するユニット  $i$  の出力  $o_i(t)$ .

$$o_i(t) = x \cdot m_i(t)$$

一般にこの出力  $o_i(t)$  そのまま用いることはなく最も大きな出力を出したユニットが勝者となり、そのユニットだけが出力出来るという考えが用いられる。

次に、自己組織化についてだが、これは教師なし、つまり出力結果が不定の状態での入力ベクトルの確率密度を近似するように学習する手法のことである。上記のネットワークに以下のアルゴリズムを取り入れることで、高次元での関係を崩さず自己組織的にネットワークを構成される。

つまり、入力データを高次元の特徴ベクトル  $x$  にパターン化を行い、出力層にある各ユニット  $i$  は入力パターン  $x$  と同次元のベクトル  $m_i$  を持つので二次元平面上に配置される。この SOM の概念を図 2 に示す。学習はこのユニットを入力パターンに近づけるという動作を選択的に行いながら進行する。この時 SOM は入力パターンと近いパターンを持つ出力ユニット、及びその近傍のユニットの集合のみ入力パターンに近づけるアルゴリズムを持っている。また、正確性の観点から学習回数には一定の学習回数を必要としている。

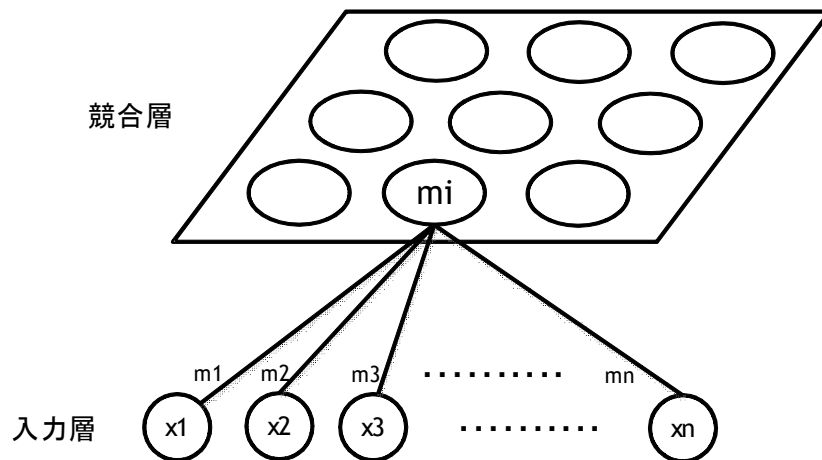


図 2. 自己組織化マップの概念図

## 自己組織化のアルゴリズム

SOMのアルゴリズムを以下に示す.

1. 各入力データをパターン化する.

$$X = \{x_1, x_2, \dots, x_k \mid x_k \in R^n\}$$

2. 出力層にある各ユニットの持つパターンを初期化する.

$$M = \{m_1, m_2, \dots, m_i \mid m_i \in R^n\}$$

3. 入力パターンに一番近いパターンを持つ出力ユニット  $c$  を探す.

次式の様な  $m_c(t)$  を持つユニット  $c$  を求める.

$$\|x - m_c(t)\| = \min_{\text{for all } i} \{ \|x - m_i(t)\| \} \quad * \dots \| \text{は距離を表している.}$$

4. 各出力ユニット  $c$  とその近傍のユニットの集合  $N_c(t)$  を入力パターン  $x_k$  に近づける.

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t) [x_k(t) - m_i(t)] & (i \in N_c(t)) \\ m_i(t) & (i \notin N_c(t)) \end{cases}$$

$$\alpha(t) = \alpha_0(t) \exp(-\|r_c - r_i\|^2 / \sigma(t)^2)$$

$\alpha(t)$  は学習率であり, 時間とともに 0 に単調減少していき,  $\|r_c - r_i\|^2$  はユニット  $c$  と  $i$  との距離である. また,  $N_c(t)$  の大きさも時間とともに単調に減少する.

5.  $k = k + 1$ ,  $k \leq m$  とし, 3.~4.を繰り返す.

6.  $t = t + 1$ ,  $t \leq T$  ( $T$  は設定された学習回数) とし,  $N_c(t)$  と  $\alpha$  を次第に小さくしながら 3.~5.を繰り返す.

1., 2.によって SOM に入力するためにパターン化を行い, 出力層の初期化を行う. 次に 3.で入力ベクトルに最も距離が近いユニット  $c$  が探索される. また, 4.によって探索されたユニット  $c$  のネットワーク上での近傍ユニットのみ更新される. これにより入力ベクトル空間で近いものは, そのネットワーク上でも互いに近傍のユニットへと射影されるような写像が完成する. 近傍学習の流れを図 3 に示す.



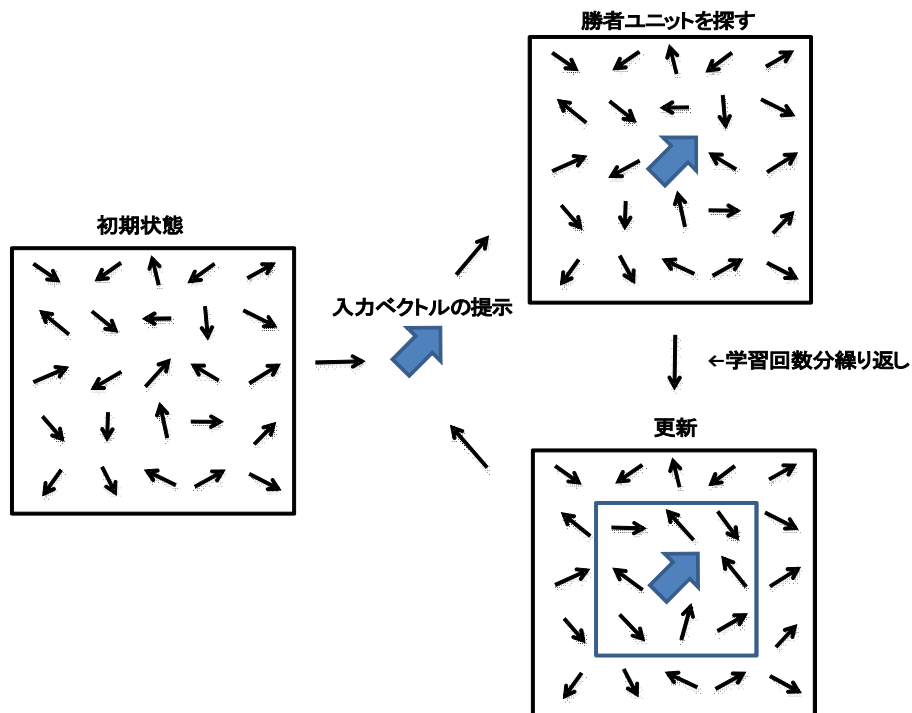


図3. 自己組織化マップの学習ステップ

本論文では以上の内容を包括したパッケージである「som\_pak-3.1」を用いて実験に臨んだ。パッケージはヘルシンキ大学のホームページにて公開されており、非営利目的ということであれば誰でも使用可能なフリーライセンスのソフトウェアである。

パッケージ内容はSOMを用いる上での必要な処理工程が記載されたソースコードが工程毎にあり、そのソースコードをコンパイル、目的とする状態に合わせて設定値を決め実行をすれば学習と二次元マップへの可視化を行ってくれる。

### 3. 既存技術の問題点と提案手法

HNS 自体は非常に有用であり従来では実現し得なかった様々なサービスの提供に結びついている。現状 HNS を通じてセキュリティ能力の向上を図っているものは、セキュリティ専用のセンサーを用いて、異常を感知した際に通報するという形式を取っている。

HNS を構成する家電にはそれぞれが独自の高性能なセンサーを搭載している。しかし、現在 HNS では家電間のネットワーク形成までは実現している[5,6,7]が、付加価値の機能に関する情報については共有していないことが多く、このため家電が搭載するセンサーをセキュリティに用いることが出来ない。これは、家電が持つセンサーとは、本来搭載している家電の機能を補助する機能であり、それ自体にセキュリティ能力は無く、単純にユーザーやユーザーの周囲の環境状況を感知するものであるため、単純にセンサーの情報をサーバで管理してもそこからの判断基準が各家庭環境に於いて一定では無いためシステムとして前以って搭載することが難しいためである。判断基準とは、セキュリティ専用のセンサーは例えば、窓の破損や警戒中の状態で動く対象を感知した時に異常と判断するのに対し、家電のセンサーはそれ単体ではそういった判断をする基準が無いということである。

これらの解決案としての手法は、家電が持つセンサー群を利用し、さらに利用者の行動パターンを学習、判定を行う機能を HNS に追加することによってセキュリティ機能を実現することである。これは利用者毎の個体差があろうともシステム自体が学習していくので汎用性があり、利用者の生活形態に沿ったセキュリティシステムを提供する。具体的な手法としては、教師なし競合近傍学習を行う SOM を用いた利用者の行動パターン学習である。人はそれぞれ個体毎の一定のパターンが見られるため、あらかじめ共通のパターンを入力しておくことは出来ないが、各自が家庭内の要所で頻繁に行う行動を SOM に学習させることにより、異常発生を感知出来ると考えられる。例えば、帰宅時に於けるパターン例としては、玄関より入室、洗面所にて洗浄、冷蔵庫より飲み物を取る、玄関側の入り口よりリビングに入室等である。それぞれ一つ一つのデータでは正常か異常かを判定出来るほどのデータでは無いが、複数のデータを絡ませることにより利用者の行動パターンを絞ることが出来る。このように利用者の行動パターンを一定期間学習させ、この学習パターンにそぐわない行動を認識した時に異常という判定を下すことが可能になると考えられる。

近年のエアコンに見られる高機能家電では優秀な人感センサー以外にも室温、輻射熱等も認識出来る熱感知センサーもあるため、不審者以外にも火災等の感知も可能であると考えられる。

纏めると、HNS により家電の家庭内 LAN を形成、ホームサーバに SOM を導入することにより、各家電から送られてくるデータを学習することが可能となる。この学習により得られた結果を正常か異常かの判定に用いることで不審者のみならず、環境的な異常(主に火災)も判別出来るという手法である。

次に、データを収集した上での異常を判断するシステムについては、各家電に対する行動を取得し、そのデータを元にして SOM による行動パターンの特徴マップを作成することである。ユーザーの行動ログを逐次追加していく事により、必要な時間帯毎に正常か異常かを判断することが出来ると考えられる。また一日という具合に判定の期間を大きく取るのではなく、時間帯毎に区切ることで精密な判定が出来ると考えられる。

これらの内容を構築することにより HNS に家電を利用したセキュリティシステムも実現できる。

## 4. 実験

以下に実験に関する内容を示す.

### 4.1 システム構成

- ・本実験でテストケースに用いる環境を図4に示す.
- ・図4での矢印は利用者の帰宅時の経路の一例を示している.
- ・赤いオブジェクトが本実験でデータとして利用する家電を示している.
- ・各家電の利用するデータを以下に示す.
  - －エアコンの On/Off
  - －テレビの On/Off
  - －冷蔵庫の開閉
  - －照明の On/Off
  - －エアコンの人感センサーによる入室角度
  - －エアコンの温度感知データ

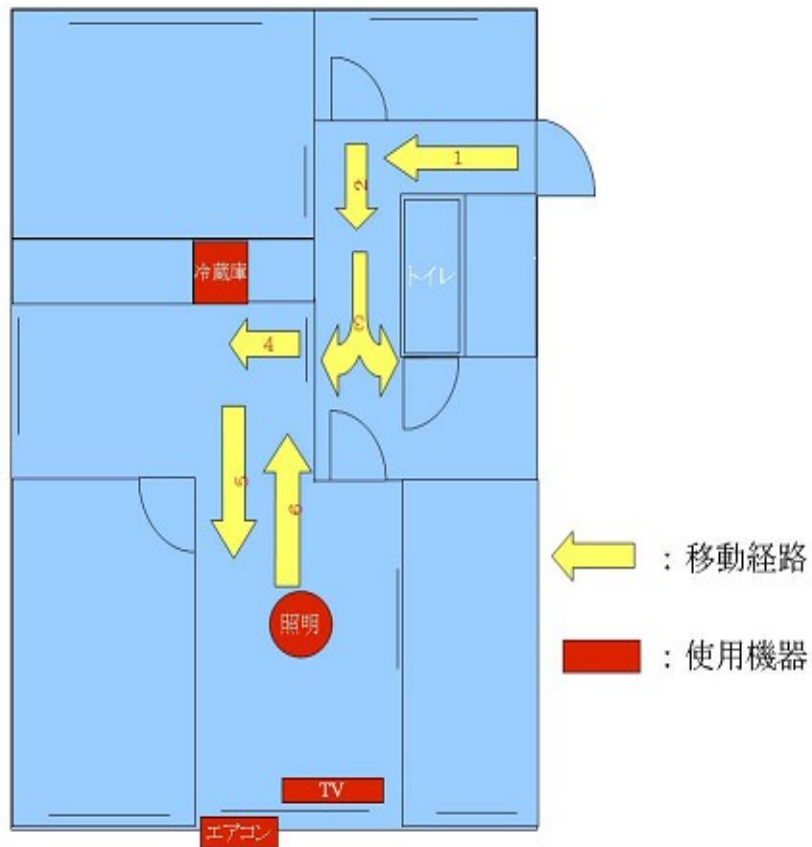


図4. システムに用いるデータの一例

## 4.2 入力データ

- ・入力するデータとして、自身の家庭内に於ける行動データのログを取得する。
- ・期間は1週間分を取得する。行動ログ(表1)。温度データ(表2)。
- ・要点としては、なんらかのアクション(家電操作等)と時間軸を取る。

活動日 時刻	12月3日	12月4日	12月5日	12月6日	12月7日	12月8日	12月9日
7:00	起床, トイレ 洗面所, 冷蔵庫開 テレビOn, エアコンOn						
8:00	テレビOff エアコンOff 外出	起床, トイレ 洗面所, 冷蔵庫開	起床, トイレ 洗面所, 冷蔵庫開 外出	起床, トイレ 洗面所, 冷蔵庫開 外出			
9:00		テレビOn エアコンOn 朝食				起床, トイレ 洗面所, 冷蔵庫開 エアコンOn	起床, トイレ 洗面所, 冷蔵庫開 エアコンOn
10:00		テレビOff エアコンOff			起床, トイレ 洗面所, 冷蔵庫開 エアコンOn	テレビOn 朝食	テレビOn 朝食
11:00		外出			朝食 エアコンOff	テレビOff エアコンOff 外出	テレビOff エアコンOff 外出
12:00					外出		
13:00							
14:00							
15:00							
16:00							
17:00							
18:00							
19:00					帰宅, トイレ 洗面所, 冷蔵庫開 エアコンOn, テレビOn		
20:00					夕食		
21:00				帰宅, トイレ 洗面所, エアコンOn			
22:00	帰宅, トイレ 洗面所, 冷蔵庫開 エアコンOn	帰宅, トイレ 洗面所, 冷蔵庫開 エアコンOn, テレビOn	帰宅, トイレ 洗面所, 冷蔵庫開 エアコンOn, 夕食	冷蔵庫開, 夕食 入浴	入浴		
23:00	テレビOn, 夕食 入浴	夕食, 入浴 エアコンOff テレビOff	入浴 エアコンOff 就寝	テレビOn テレビOff	テレビOff エアコンOff	帰宅, トイレ, 洗面所 冷蔵庫開, エアコンOn テレビOn, 夕食	帰宅, トイレ 洗面所, 冷蔵庫開 エアコンOn, 夕食
0:00	エアコンOff テレビOff	就寝		就寝		入浴 エアコンOff テレビOff	入浴 エアコンOff
1:00	就寝				就寝	就寝	就寝

表1. 1週間分の行動ログデータ

時間帯 活動日	朝 7:00~13:00	昼 13:00~19:00	夜 19:00~1:00
12月3日	8	13	11
12月4日	8	15	13
12月5日	6	16	14
12月6日	5	15	13
12月7日	3	14	12
12月8日	4	16	14
12月9日	4	15	13

単位 (°C)

表2. 1週間分の温度データ

エアコン、テレビ、冷蔵庫、照明の利用についてデータは、侵入者の目的(窃盗)から、これらを利用するとは考えにくいいため正常か異常かの判断基準になる。また、エアコンの人感センサーによって図4にあるような正規利用者が定例で通るルートとは違う窓の方角等からの侵入を感知出来るため判断基準として成り立つ。最後に温度の感知データからは、利用者の不在時に異常な高温を感知した場合に火災が発生しているという異常の判定ができると期待できる。

・データを元にして、(表3)(表4)(表5)にある様にデータの数値化を行う。

	On/開	Off/閉
エアコンの状態	1	0
テレビの状態	1	0
冷蔵庫の状態	1	0
照明の状態	1	0

表3. 家電利用に於ける設定値

角度※ 設定値	対象無し	$0 < X \leq 10$	$10 < X \leq 20$	$20 < X \leq 30$	$30 < X \leq 40$	$40 < X \leq 50$
	0	1	2	3	4	5
	$50 < X \leq 60$	$60 < X \leq 70$	$70 < X \leq 80$	$80 < X \leq 90$	$90 < X \leq 100$	$100 < X \leq 110$
	6	7	8	9	10	11
	$110 < X \leq 120$	$120 < X \leq 130$	$130 < X \leq 140$	$140 < X \leq 150$	$150 < X \leq 160$	$170 < X < 180$
	12	13	14	15	16	17

※ エアコン側から見て、正面左側を0°とする。

表4. エアコンの人感センサーに於ける設定値

温度(°C)	設定値
$0 < Y \leq 10$	0
$10 < Y \leq 20$	1
$20 < Y \leq 30$	2
$30 < Y \leq 40$	3
$40 < Y \leq 50$	4
$50 < Y$	5

表5. 温度センサーに於ける設定値

### 4.3 som\_pak

som\_pak-3.1 をヘルシンキ大学よりダウンロードすると、Tape ARchive format (TAR) によって1ファイルに纏められている。som\_pak-3.1 は基本的にLinux系OSを主として考えられているが、Win系OSでも使用は可能である。但し本論文についてはLinux系OSで実験を行ったためWin系OSの実行方法については割愛する。

まずは纏められているtarファイルを「tar xf som\_pak-3.1.tar」とコマンドを入力してファイルの解凍を行う。解凍後は、som\_pak-3.1 という表6の様なファイルで構成されたフォルダが出力される。

ヘッダーファイル	Cソースファイル	データファイル	その他のファイル
config.h	datafile.c	ex.dat	header.ps
datafile.h	fileio.c	ex_fdy.dat	makefile.dos
errors.h	header.c	ex_fts.dat	makefile.mak
fileio.h	labels.c	ex_ndy.dat	makefile.unix
labels.h	lvq_pak.c		
lvq_pak.h	map.c		
som_rout.h	mapinit.c		
umat.h	median.c		
version.h	planes.c		
	qerror.c		
	sammon.c		
	som_rout.c		
	umat.c		
	vcal.c		
	version.c		
	vfind.c		
	visual.c		
	vsom.c		

表6. som\_pak-3.1 フォルダの構成ファイル

次に実行可能形式のファイル作成のためにコンパイルを行う。

```
cp makefile.unix makefile
make
```

正常に実行可能形式のファイルが作成されたか以下のコマンドを用いて確認する。

```
File * | grep executable | cut -f 1 -d: | sort
```

出力結果として11個のファイルが確認出来れば正常に作成出来ている。しかし、ここでファイル数が11個に満たない場合は作成が失敗している可能性が大きいいため、Makefileの内容を修正するなどしてから再度コンパイルを行う必要がある。

次に用意する入力データについては、ベクトルの次元数を記入すること、アスキー文字列のみ使用可能、2バイト文字コード系の日本語は不可、データの区切りにタブは不可、UNIX版ではUNIX形式の改行のみ使用可(MS-DOS形式の改行は不可)というルールの下で作成する。またファイル形式はDATとする。

プログラムでは、表7の様なオプションを用いて学習のパラメータを指定する。

-din	入力用のデータを読み込むファイル名。
-dout	出力用のデータを保存するファイル名。
-cin	参照するコードブックベクトルを読み込むファイル名。
-cout	コードブックベクトルを保存するファイル名。
-rlen	学習回数。
-alpha	初期の学習率係数 $\alpha(0)$ 。学習が進むにつれ、0に近づく。
-radius	学習近傍領域の初期値(半径)。学習が進むにつれ、1に近づく。
-xdim	x方向のユニット数(次元)。
-ydim	y方向のユニット数(次元)。
-topol	マップに使われている学習近傍の形状。六角格子型(hexa)と直角格子(rect)のみ指定出来る。
-neigh	近傍関数の種類。ステップ関数(bubble)とガウス関数(gaussian)のみ指定出来る。
-plane	planesで平面表示する参照ベクトルの構成要素を指定する。
-fixed	固定した座標への強化に関する修飾語“fixed = n.n (nは整数)”を有効にする。規定値は無効(0)である。
-weights	強調因子に関する修飾語“weight = f.f (f.fは浮動小数点表記)”を有効にする。規定値は無効(0)である。
-rand	乱数の種を指定する。

表7. プログラムで用いるオプション

ここからマップの初期化, 学習, マップへの視覚化の流れを記載する.

マップの参照ベクトルは最初に仮の値で初期化され, 学習に使われる近傍形状・近傍関数はこの時に定義される. この時用いるプログラムは `randinit` である.

```
randinit -din ex.dat -cout ex.cod -xdim 12 -ydim 8
          -topol hexa -neigh bubble -rand 1
```

入力データ(学習させる元のデータ)は `ex.dat` であり, `rand 1` で定義される乱数でマップを初期化した後, 六角格子型の格子で  $12 \times 8$  にユニットを配置し, 近傍関数としてステップ関数を指定している.

学習に用いるプログラムは, `vsom` である. SOMは2段階の学習を必要としており, 1段階目でユニットの参照ベクトルの順序付けを行い, 2段階目で参照ベクトルが整理される.

```
vsom -din ex.dat -cin ex.cod -cout -rlen 1000 -alpha 0.05 -radius 10 -rlen 1
vsom -din ex.dat -cin ex.cod -cout -rlen 10000 -alpha 0.02 -radius 3 -rlen 1
```

1段階目の学習で大まかに形づくり, 2段階目の学習で正確な値に収束する. このため2段階目の学習は1段階目の学習より長くすることが一般的である. 上記の実行では学習回数を10倍多くし, 逆に近傍半径・学習率は減らして精密な学習としている.

学習されたマップはパッケージに含まれるプログラムで可視化することが可能である. この可視化にはマップの写像や時間に対する最整合したユニットの軌跡のプロットなどである. また以下のプログラムを用いることでデータ内にラベル付けされたデータがある場合, マップ中の各ユニットに対してラベル付けを行うことも可能である.

```
vcal -din ex.dat -cin ex.cod -cout ex.cod
```

最後に, `planes`, `umat` のプログラムを用いてカプセル化ポストスクリプト(EPS)画像へ変換することで学習されたデータを実際に平面画像で確認することが出来る.

```
umat -cin ex.cod -eps -o ex.eps
```

以上の実行の流れは, シェルスクリプトとして作成しておくことで作業の短縮が可能である. 作成例は付録に示す. [11,12,15,17,18]



#### 4.4 実験

1. 朝(7:00～13:00)・昼(13:00～19:00)・夜(19:00～翌 1:00)の取得データから som\_pak への入力データを作成する。入力データの例を(表 8)(表 9)(表 10)に示す。
2. 作成した入力データを som\_pak に入力，実行し，時間帯毎のマップを取得する。
3. 異常と見られるパターンのデータを想定し，入力データに対して，(表 11)に示す異常データを加え，異常パターンを含んだ入力データとして作成する。
  - ・異常パターンの内容
    - 火災発生．データ名：fire
      - － 時間帯共通
      - － 室温の異常温度 → 50 度以上の温度を検出することを想定。
      - － 夜の時間帯
        - ・ HNS の利用者が存在する上での火災を想定。
    - 泥棒の侵入．データ名：theft
      - － 朝の時間帯
        - ・ 室温は正常と同じ数値。
        - ・ 人感センサーによる感知。
        - ・ 家電の利用無。
        - ・ 侵入経路が特殊なパターン(theft1)。
        - ・ 侵入経路が正常と同様なパターン(theft2)。
      - － 昼の時間帯
        - ・ 室温は正常と同じ数値。
        - ・ 人感センサーによる感知。
        - ・ 家電の利用無。
      - － 夜の時間帯
        - ・ 室温は正常と同じ数値。
        - ・ 人感センサーによる感知。
        - ・ 家電の利用有。
        - ・ 侵入経路が特殊なパターン。
  - 4. 異常データを含んだ新しい入力データを som\_pak に入力，実行し，異常パターンを含んだデータの時間帯毎に対するマップの作成を行い，マップの正常のみの入力データと異常データを含んだ入力データの出力パターンを比較する。

実験に用いたデータの一例を以下に示す。

6 (ベクトルの次元数)						朝 (7:00~13:00)
エアコン	テレビ	冷蔵庫	人感センサー	室温	照明	データ名
1	1	1	10	0	1	12月3日
1	1	1	10	0	1	12月4日
0	0	1	10	0	1	12月5日
0	0	1	10	0	1	12月6日
1	1	1	10	0	1	12月7日
1	1	1	10	0	1	12月8日
1	1	1	10	0	1	12月9日

表 8. 朝の行動にあたる入力データ

6 (ベクトルの次元数)						昼 (13:00~19:00)
エアコン	テレビ	冷蔵庫	人感センサー	室温	照明	データ名
0	0	0	0	1	0	12月3日
0	0	0	0	1	0	12月4日
0	0	0	0	1	0	12月5日
0	0	0	0	1	0	12月6日
0	0	0	0	1	0	12月7日
0	0	0	0	1	0	12月8日
0	0	0	0	1	0	12月9日

表 9. 昼の行動にあたる入力データ

6 (ベクトルの次元数)						夜 (19:00~1:00)
エアコン	テレビ	冷蔵庫	人感センサー	室温	照明	データ名
1	1	1	10	1	1	12月3日
1	1	1	10	1	1	12月4日
1	0	1	10	1	1	12月5日
1	1	1	10	1	1	12月6日
1	1	1	10	1	1	12月7日
1	1	1	10	1	1	12月8日
1	0	1	10	1	1	12月9日

表 10. 夜の行動にあたる入力データ

6 (ベクトルの次元数)						朝 (7:00~13:00)
エアコン	テレビ	冷蔵庫	人感センサー	室温	照明	データ名
0	0	0	0	5	0	fire
0	0	0	16	0	0	theft1
0	0	0	10	0	0	theft2
6 (ベクトルの次元数)						昼 (13:00~19:00)
エアコン	テレビ	冷蔵庫	人感センサー	室温	照明	データ名
0	0	0	0	5	0	fire
0	0	0	8	1	0	theft
6 (ベクトルの次元数)						夜 (19:00~1:00)
エアコン	テレビ	冷蔵庫	人感センサー	室温	照明	データ名
1	0	1	10	5	1	fire
0	0	1	16	1	0	theft

表 11. 異常パターン入力データ (fire : 火災, theft : 泥棒)

## 5. 実験結果

以下に実験結果を示す.

学習前のマップを図5に示す. このマップはランダムに配置されたノードに対してデータが押し花の様に配置されているだけなので規則制は無い.

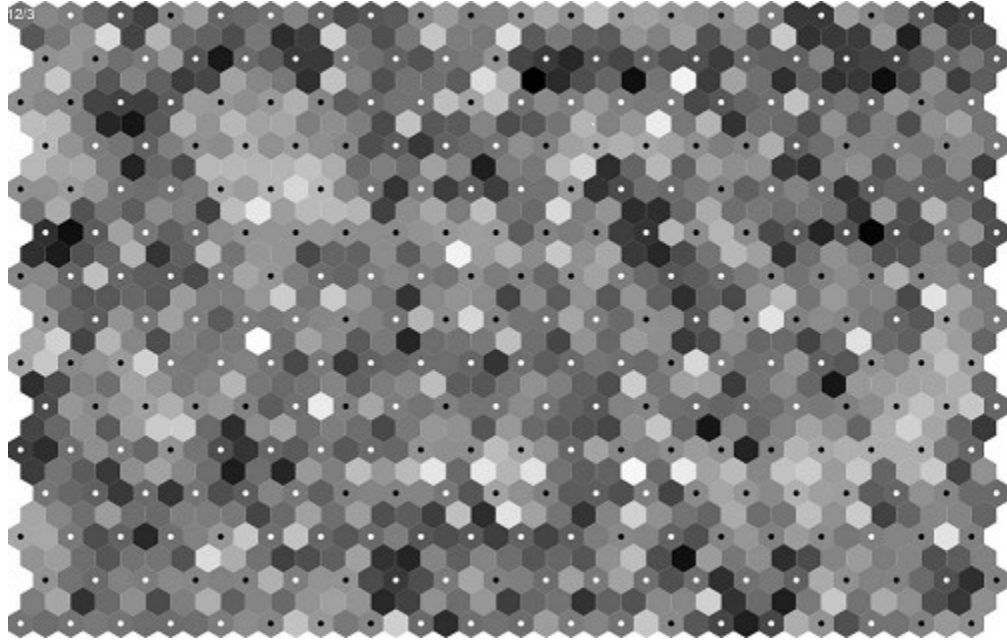


図5. 学習前マップ

次に学習され, グループ化された時間帯毎のマップを(図6)(図7)(図8)に示す.

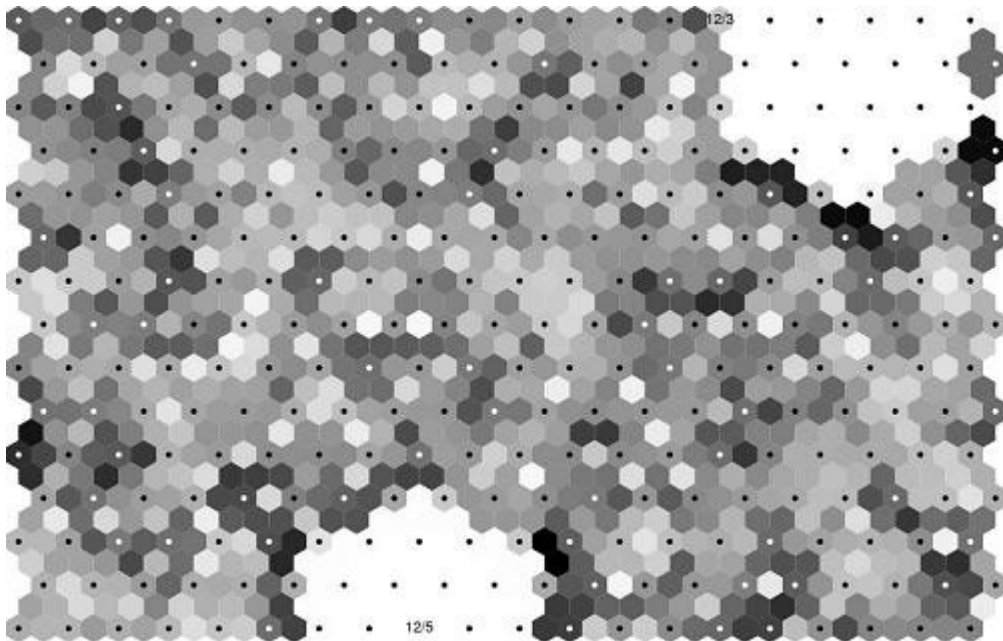


図6. 学習後マップ(朝の時間帯)

図6を見ると中央下部に12月5日, 12月6日の休日のデータが集まり, 右上部には, 平日の行動データが集中しているのが分かる.

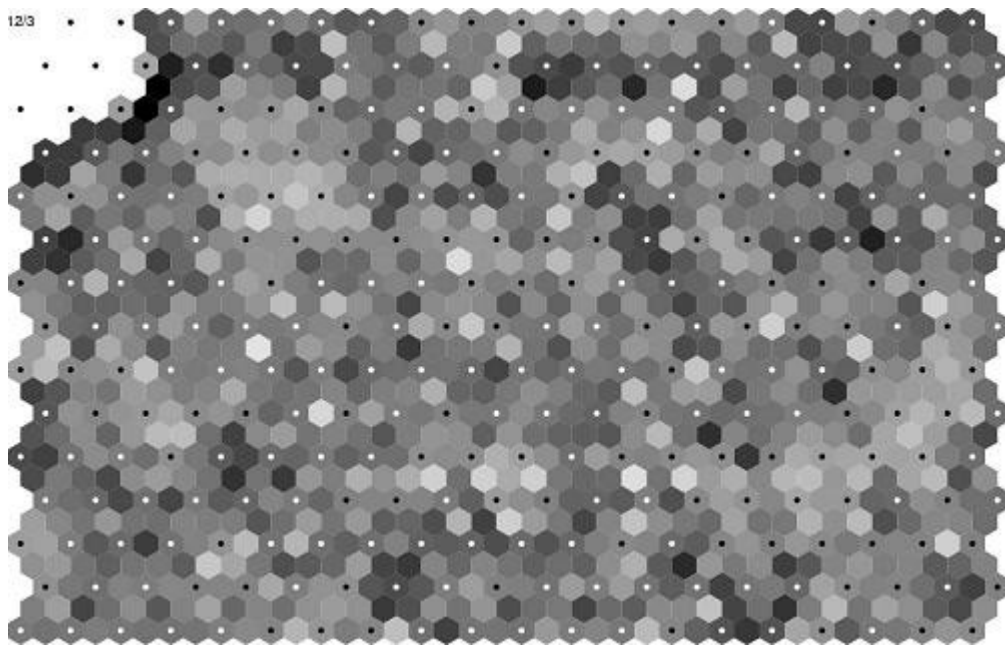


図7. 学習後マップ(昼の時間帯)

図7では、行動を測定した期間に於いて、常に外出をしていたため左上部に集中する結果となったのが分かる。ここから測定対象が不在であっても、温度等の環境条件よりマップを作成出来ることが分かる。

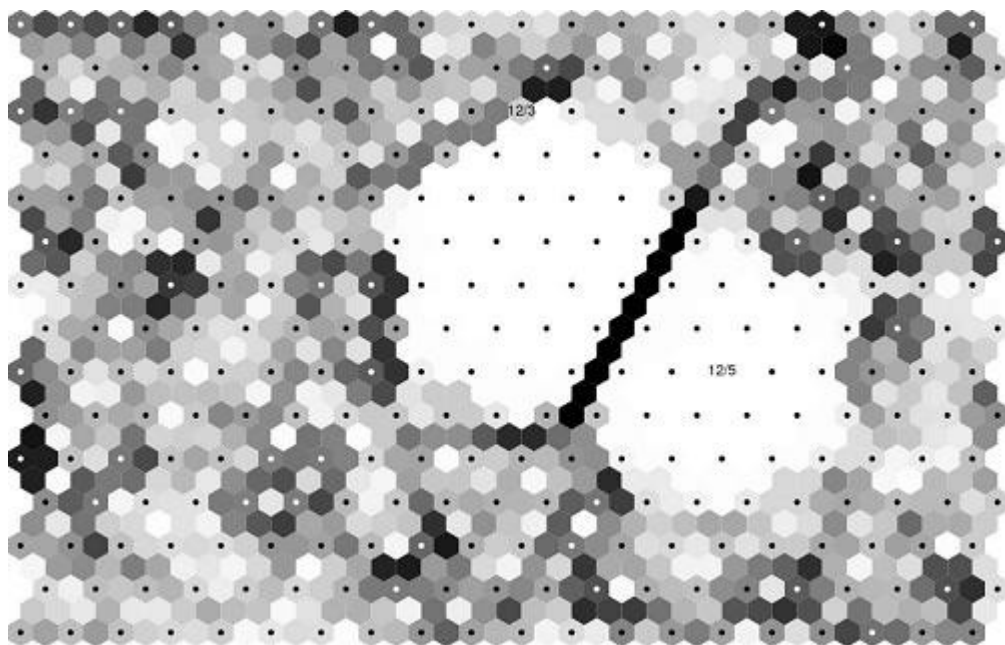


図8. 学習後マップ(夜の時間帯)

図8では、測定したログデータが大部分で同様の数値を出しているため、グループが中央付近に非常に近い位置で形成されているのが見て取れる。

最後に、異常データを追加した入力データから作成された時間帯毎のマップを(図 9)(図 10)(図 12)に示す。

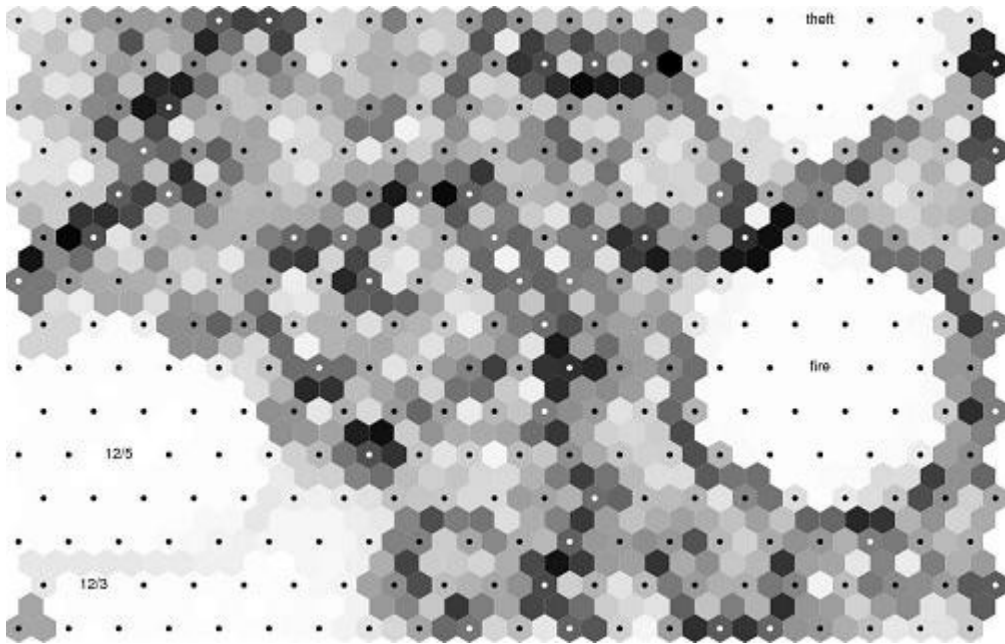


図 9. 異常データ追加マップ(朝の時間帯 theft1 のみ)

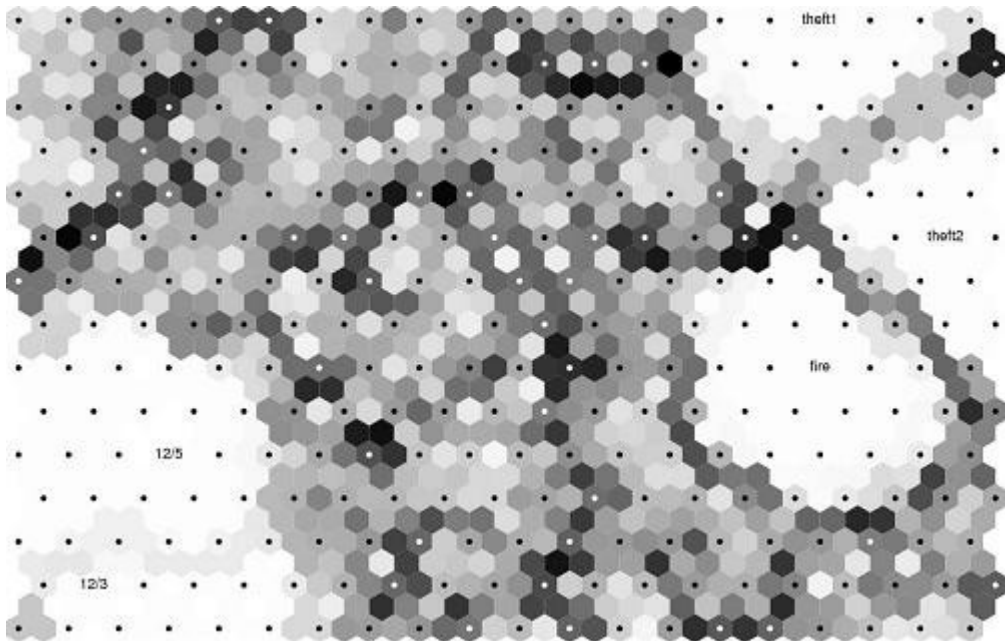


図 10. 異常データ追加マップ(朝の時間帯 theft1, theft2)

図 9 では、左下部に位置する、正常なデータで構成されるグループと、右中部、右上部に形成される異常なデータで構成されるグループという感じにはっきりと分かれているのが見て取れる。

また、進入経路を正常のパターンと同じ値にした theft2 を更に入力データに追加した図 10 に於いてもやはり異常なデータは右上部に近い位置に形成された。

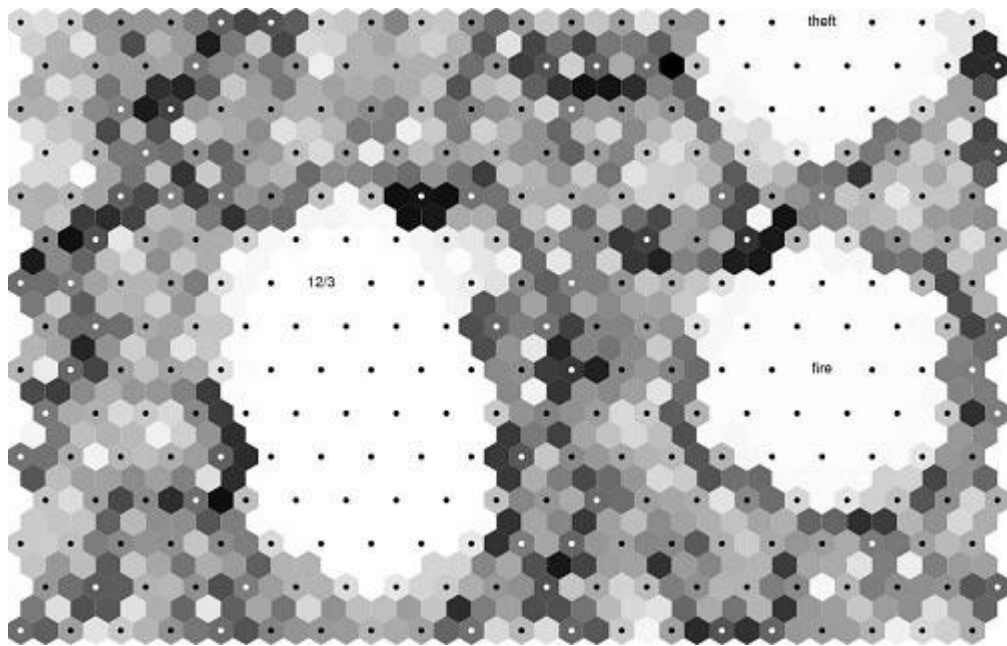


図 11. 異常データ追加マップ(昼の時間帯)

図 11 から同様に、中央や左の位置に正常データのグループが形成され、fire や theft の様な異常データについては右上部、右中部に形成されているのが見て取れる。

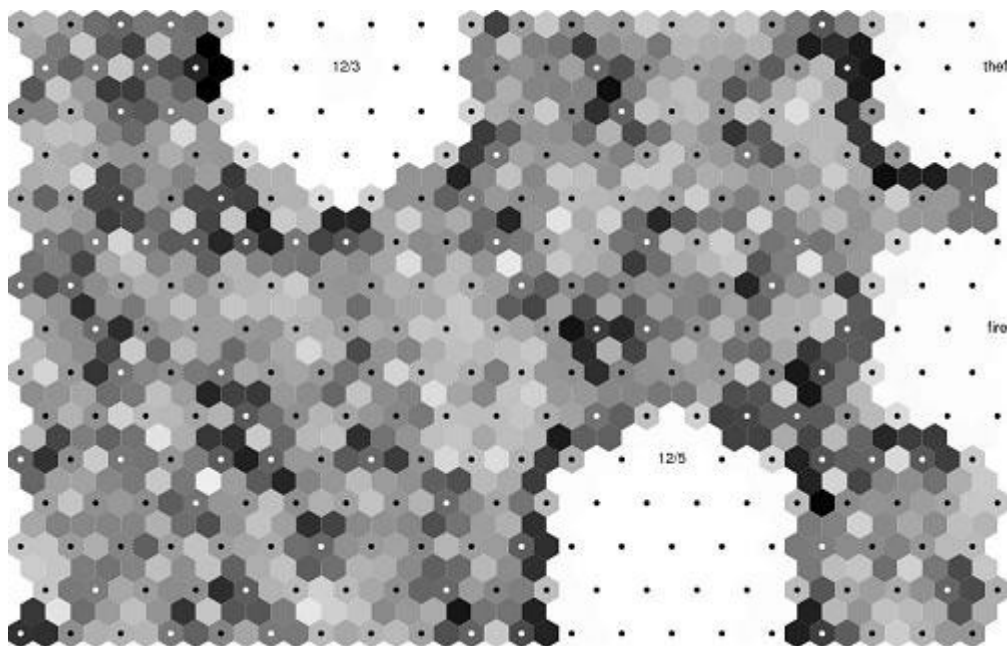


図 12. 異常データ追加マップ(夜の時間帯)

図 12 では、正常データのみで作成された図 8 のマップより大きく位置関係は変わったのが見て取れる。図 8 で隣り合う様に中央で隣接していたが、このマップで隣接はしていない。しかし他のマップ同様に異常データのグループは明確に分かれた位置で形成されている。

## 6. 考察

実験前の仮定として、屋内の利用者の行動に一定の決まった行動があるとした通りに som\_pak によって出力されたマップは基本的に固まった部分にグループパターンを作成した。これは人の生活習慣による物であり、また、本実験でのグループが分かれる要因としては平日、休日での行動の違いという事が大きく、この事を考慮すると、実験データを更に多く取得したとしても本実験結果に近い結果を示すと予測できる。具体的には図6のマップで見られるような、平日と休日という二パターンが大きなグループとして形成され、そのグループの中で微小な変化を伴うノードが出現すると考えられる。

また SOM の学習機能についても図7のマップから分かるように利用者が常に不在であるため比較となるデータが取得出来なくとも、利用者がいないという状態を正常状態としてマップを形成し、他の時間帯同様に異常を判別する汎用性の高さが見て取れる。

これの基本事項を踏まえた上で異常データを含んだ実験結果を観察すると、朝の時間帯に於けるデータに於いて、異常データの追加前、追加後でグループの併合を見てとれる。これは、異常データが追加される前では比較対象として大きな要素であった家電の利用内容が、異常データという更に大きな比較対象の出現によって性質的に近い正常の利用行動をグループとして併合させたと考えられる。つまり、SOM の近傍学習が正常に機能し、はっきりと別グループとして異常データをパターン分けさせ、尚且つ、性質的に近い行動である正常のデータを近傍するグループとして扱ったということが判別出来る。また、正常の行動以外にもその性質は見てとることが出来る。まず共通して火災や泥棒という異常行動は非常に近い位置にグループを形成する事が多く、正常、異常の判別にこういった要素も考慮出来ると考えられる。加えて、侵入経路を正常と同様に取ったとしても、普段の定例行動を行わなければ theft2 で見られるように異常のグループ付近に位置すると SOM は判断するため、正常、異常の判断を行う正確性も高いと言える。夜の時間帯に於いては、火災のデータ内に正常行動を行う利用者行動データを含めたが、こちらも同様に正常グループの近くに位置はしているが、全体としては、はっきりと分かれた別のグループとしてマップを作成しているため、在宅中の気付かない火災にも対応出来ると考えられる。

## 7. 結論

実験開始前の仮定は実験結果からも分かる様に、HNS を構成する環境に於いては、SOM を用いる事で専門的なセキュリティセンサーや火災センサー等を設置しなくとも、これらと同等のセキュリティを実現出来る。更に、このシステムは特殊な機器に依存しないシステムであるため拡張性、汎用性の高さを望め、構成の容易さ、特別な条件も要求しない事を考慮すると従来のセキュリティシステムより非常に優位であると言える。

拡張性とは、SOM の特性でもある取得データの内容に依らず客観的に判定を行う事が出来るという事であり、どのようなデータであってもその判断材料として加える事が出来る。つまり、本実験で挙げた要素以外にも、例えばエアコンによる空気成分を調べるセンサーのデータを加えれば、一酸化炭素中毒やガス漏れによる被害も未然に防ぐことが出来る様になる。この様に、従来ではその懸念毎に新たな機器の設置が必要であったが、本システムでは家電が持つ機能より得られるデータを取得するだけで屋内の安全性を容易に挙げる事が可能である。

以上を以て、家電の高性能化、HNS の構成の普及が目覚ましい今日に於いて、SOM によるセキュリティシステムの導入は非常に有用であると言える。

## 謝辞

本研究を進めるにあたり多くの方々の御指導，御支援を頂きました。心より深く御礼申し上げます。

龍谷大学 理工学部 情報メディア学科 三好力 教授には，本研究の指導教員を担当して頂きました。研究内容を始め，進学，生活面での相談まで御指導，御助力頂きました。研究方針については本当に親身に御教示頂きました。心より御礼申し上げます。

龍谷大学 理工学部 情報メディア学科 三好研究室の皆様には，研究に対する多くの御意見，御助力，御指導を頂きました。心より御礼申し上げます。



## 参考文献

- [1] 福岡 佑介. ネットワーク家電と標準インタフェースとの動的結び付けによるマルチベンダホームネットワークシステム構築手法. 奈良先端科学技術大学院大学情報科学研究科卒業論文. Feb. 2008.
- [2] 伊原 誠人. 榊原 弘記. 湯浅 直弘. 中村 匡秀. 松本 健一. ホームネットワークシステムにおけるサービス開発へのペルソナシナリオ法の適用と評価. 奈良先端科学技術大学院大学情報科学研究科. 社団法人 電子情報通信学会.
- [3] パッタラ・リーラーブルット. 中村 匡秀. 井垣 宏. 松本 健一. 菊野 亨. ホームネットワークシステムにおけるサービス競合の分類と解消について. 大阪大学大学院情報科学研究科. 奈良先端科学技術大学院大学情報科学研究科. 社団法人 電子情報通信学会.
- [4] 福田 将之. 井垣 宏. 中村 匡秀. ホームネットワークシステムにおけるリアルタイムな家電制御サービスの実現. 神戸大学大学院工学研究科情報知能学専攻. 社団法人 電子情報通信学会.
- [5] 井垣 宏. 長江 洋子. 山田 松江. 中村 匡秀. 松本 健一. 青山 幹雄. ユーザの移動に伴う機器と環境の状態引き継ぎを実現するホームネットワークシステムの提案. 南山大学数理情報学部. 奈良先端科学技術大学院大学情報科学研究科. 社団法人 電子情報通信学会.
- [6] 永原 崇範. 鹿島 拓也. 猿渡 俊介. 川原 圭博. 南 正輝. 森川 博之. 青山 友紀. 篠田 庄司. ユビキタス環境に向けたセンサネットワークアプリケーション構築支援のための開発用モジュール U3 (U-cube) の設計と実装. 中央大学大学院. 東京大学大学院. 社団法人 電子情報通信学会.
- [7] NEC Business Solution. IT Square. 第8回 企業業務や生活を支援するセンサーネットワーク. ユビキタス eye -ユビキタス テクノトレンド-. 2004.
- [8] 村上 剛司. 長谷川 勉. 木室 義彦. 千田 陽介. 家永 貴史. 有田 大作. 倉爪 亮. 日本ロボット学会誌. Vol. 26 No. 2. page192-199. 2008.
- [9] 椎尾 一郎. 日常生活のユビキタスコンピューティング. お茶の水女子大学理学部. センシングネットワーク: 要素技術編. 人工知能学会誌 Vol. 23 No. 5. Sep. 2008.
- [10] 金 明哲. Rと自己組織化マップ. 同志社大学文化情報学部. フリーソフトによるデータ解析・マイニング. Jan. 2006.

- [11] 澁谷 慧一郎, 遠山 元道. 自己組織化マップを利用する分類済み階層の自動設定. 慶應義塾大学理工学部情報工学科.
- [12] 亀井 俊之. 自己組織化マップを用いた画像検索エンジンの構築に関する研究. 神戸大学工学部情報知能工学科卒業論文. Feb. 1997.
- [13] 古川 徹生. SOM 集合をマップする SOM: 自己組織化マップから自己組織化ホモトピーへ. 九州工業大学大学院生命体工学研究科.
- [14] 島田 敬士, 谷口 倫一郎. 密度可変型自己組織化マップによる追加学習法. 九州大学大学院システム情報科学府. 社団法人 電子情報通信学会. Dec. 2006.
- [15] 伊藤 冬子, 廣安 知之, 三木 光範. 自己組織化マップ(Self-Organizing Maps)の基礎. Intelligent Systems Design Laboratory Research Reports. Jan. 2006.
- [16] [田中 雅博](#). 自己組織化マップ (SOM : Self Organizing Maps)
- [17] 徳高 平蔵, 大北 正昭, 藤村 喜久朗. 自己組織化マップとその応用. シュプリンガー・ジャパン株式会社. Jul. 2007.
- [18] 徳高 平蔵, 岸田 悟, 藤村 喜久朗. 自己組織化マップの応用. 海文堂出版株式会社. Feb. 1999.

## 付録

# ex はファイル名。 他ファイル使用時はリネーム。

# Initialize

echo "Execute randinit ..."

./randinit -din ex.dat -cout exi.cod -xdim 30 -ydim 25 -topol hexa -neigh bubble -rand 123 #マップの参照ベクトルの初期化. xdim : x 軸 30 ydim : y 軸 25 -topol hexa : 6 角格子型 -neigh bubble : 近傍関数をステップ関数 乱数の種を 123

# Learning Step

echo "Execute vsom ..."

./vsom -din ex.dat -cin exi.cod -cout ex2.cod -rlen 7000 -alpha 0.04 -radius 6 -rand 0 #マップの学習. -rlen : 学習回数 -alpha : 更新係数の初期値(学習率) -radius : 初期近傍半径 乱数は不定

./vsom -din ex.dat -cin exi.cod -cout ex2.cod -rlen 10000 -alpha 0.02 -radius 3 -rand 0

# Labeling

echo "Execute vcal ..."

./vcal -din ex.dat -cin exi.cod -cout exiv.cod

./vcal -din ex.dat -cin ex2.cod -cout ex2v.cod

# umat : 参照ベクトルの可視化

# Create EPS file of U-Matrix map EPS ファイルの出力

echo "Execute umat ..."

./umat -cin exiv.cod -eps -o exiv.eps

./umat -cin ex2v.cod -eps -o ex2v.eps

# Create PS file of U-Marix map PS ファイルの出力

./umat -cin exiv.cod -ps -o exiv.ps

./umat -cin ex2v.cod -ps -o ex2v.ps

# Create Sammon's map サモンマップの出力. 規程のファイル名の規則に従い出力

echo "Execute sammon ..."

./sammon -cin ex2v.cod -cout ex2v.sam -rlen 100 -eps

# Create Plane map 要素のフィルタリング

# plane 1: is small

# plane 4: is night

# plane 16: eat grass

echo "Execute planes ..."

./planes -cin ex2v.cod -plane 3

./planes -cin ex2v.cod -plane 4

./planes -cin ex2v.cod -plane 16