

Support Vector Machine and Probability Neural Networks in a Device-free Passive Localisation (DfPL) Scenario

Gabriel Deak, Kevin Curran, *Senior Member, IEEE*, Joan Condell, Daniel Deak, and Piotr Kiedrowski

Abstract—The holy grail of tracking people indoors is being able to locate them when they are not carrying any wireless tracking devices. The aim is to be able to track people just through their physical body interfering with a standard wireless network that would be in most peoples home. The human body contains about 70% water which attenuates the wireless signal reacting as an absorber. The changes in the signal along with prior fingerprinting of a physical location allow identification of a person's location. This paper is focused on taking the principle of Device-free Passive Localisation (DfPL) and applying it to be able to actually distinguish if there is more than one person in the environment. In order to solve this problem, we tested a Support Vector Machine (SVM) classifier with kernel functions such as Linear, Quadratic, Polynomial, Gaussian Radial Basis Function (RBF) and Multilayer Perceptron (MLP), and a Probabilistic Neural Network (PNN) in order to detect movement based on changes in the wireless signal strength.

Index Terms—Device-free Passive Localisation, Support Vector Machine, Neural Networks, Wireless Sensor Networks.

I. INTRODUCTION

Indoor location estimation is a crucial component in many applications. Location estimation is important for many scenarios such as asset tracking, health care, location based network access, games, manufacturing, government, logistics, industry, shopping, security, tour guides, and conference guides. Various localisation systems that can estimate the position of a person or object exist. One can select the system which offers the accuracy and precision required for a specific application.

Indoor localisation systems can be classified into active and passive systems. Location tracking techniques for active localisation require the tracked people to participate actively. The second class known as passive localisation is based on monitoring changes of characteristics dependent on people's presence in an environment. By participating actively, we mean that a person carries an electronic device which sends information to a positioning system helping it to infer that person's position. In some cases the electronic devices can also process recorded data and send the results for further processing to an application server running the localisation

algorithm. In the passive localisation case, the position is estimated based on the variance of a measured signal or video process. Thus the tracked person is not carrying any electronic devices to infer the user's position.

This work is focused on solving an extremely difficult task that is multi-occupancy detection in a passive localisation scenario. Thus the following sections will analyse one of the techniques used to deploy indoor passive localisation systems. Various DfPL systems will be presented as an introduction to indoor passive localisation. Various techniques such as Ultra-wideband (UWB), Physical Contact, Differential Air Pressure, Computer Vision, and Device-free Passive Localisation (DfPL) have been used in indoor passive localisation.

Ultra-wideband (UWB) is one of the first techniques used to deploy passive localisation systems [1]. Through-the-wall surveillance or through-wall imaging (TWI) are used to denote UWB passive systems [2], [3]. This technique has been recently used for both static and motion detection. UWB passive localisation is considered to be an extension to a technique called radio tomographic imaging due to its similarity to the medical tomographic imaging. Through-wall imaging refers to the ability of detecting and monitoring objects or people through buildings walls. This can be very useful to law enforcement agencies and can have many applications in military and civil scenarios [4]. UWB has the advantage of being able to penetrate walls. Various implementations of UWB technique have been proposed. A UWB system has the following two main components: transmitters and receivers. Short pulses are sent by a pulse generator via a horn antenna [5]. The receivers wait and monitor echoes from various objects or people.

TileTrack represents a low cost two-dimensional location estimation system based on physical contact [6]. Changes in the capacitance between transmitting and receiving electrodes (plate electrodes or wire electrodes) are monitored. The system is based on 9 floor tiles with one transmitting electrode for each tile. Each tile is 60 cm by 60 cm square-shaped made from thick chip-board with thin steel coating. The prototype used to deploy the TileTrack technique has a square tracking area with a size of 3 x 3 tiles.

AirBus estimates location based on indoors airflow disruption caused by human movement [7]. An air pressure sensor is placed within the central heating, ventilation, and air conditioning (HVAC) unit. The sensor detects pressure variations. AirBus can correctly identify an open or closed door 80% of the cases with HVAC in operation and 68% with HVAC unit switched off.

Gabriel Deak, Kevin Curran and Joan Condell are with School of Computing and Intelligent Systems, Faculty of Computing and Engineering, University of Ulster, Derry, N. Ireland, BT48 7JL, UK (e-mail: Deak-G@email.ulster.ac.uk)

Daniel Deak is with S.C. Centrul de Calcul Info98 S.A., 2 Timisoara Street, 332015, Petrosani, Romania

Piotr Kiedrowski is with Institute of Telecommunication, University of Technology and Life Science, ul. Kaliskiego 7, 85-789 Bydgoszcz, Poland

Computer vision can be considered as a DfPL system because the tracked people are not carrying any electronic devices or tags [8]. The EasyLiving project [9] is a computer vision based system which aims to transform any environment in a smart environment dependent on location information. Possible applications include switching on/off devices near to the users location, monitoring peoples behaviour and many others. The system architecture consists of three PCs (Personal Computers) and two sets of colour cameras. Each camera is connected to one PC, while the third PC is used for running the person tracker algorithms. Video processing algorithms are used to separate and track people. The system was tested with a maximum of three people in the environment. The possibility of obstructions depends on the behaviour and the number of persons.

The Device-free Passive Localisation (DfPL) [10], [11] is based on monitoring the variances of the signal strength in a wireless network. The human body contains about 70% water and it is known that waters resonance frequency is 2.4 GHz. The frequency of the most common wireless networks is 2.4 GHz, thus the human body behaves as an absorber attenuating the wireless signal [2], [4], [12]–[15]. This technique is the focus of our research and the remainder of the paper is based on DfPL using Wireless Sensor Networks (WSNs).

The paper is organised as follows: Section II introduces Support Vector Machine Classification with various kernel functions, Section III discusses Probabilistic Neural Network technique, Section IV presents the test bed and motion detection technique using the classifier introduced in Section II. Section IV concludes the paper.

II. SUPORT VECTOR MACHINE (SVM) CLASSIFICATION

SVM is a supervised learning method for data analysis, pattern recognition, classification and regression analysis. SVM uses training vectors, pairs of inputs-outputs, to build a model that is used afterwards to predict classes that new data belongs to. For our tests we used two Matlab functions *svmtrain* and *svmclassify* defined by:

$$\begin{aligned} SVMStruct &= svmtrain(Training, Targets, Name, Value) \\ PredClass &= svmclassify(SVMStruct, TestData) \end{aligned} \quad (1)$$

Training and *Targets* represent the input-output pairs used for training. *Name-Value* pair specify optional arguments. *svmclassify* uses the obtained model to classify new data. One can find more details about SVM classification in Matlab including all the optional parameters in [16]. We tested various kernel functions as follows:

$$SVMStruct = svmtrain(Training, Targets, 'kernel_function', 'polynomial') \quad (2)$$

where '*kernel_functions*' is an optional argument and '*polynomial*' represents the type of kernel function used for training. The training process can use kernel functions such as Linear, Quadratic, Polynomial, Gaussian RBF and MLP. *SVMStruct* represent the model obtained after training. This is a structure

containing information about the trained SVM classifier. A field of interest in this structure/model is *GroupNames* which returns the predicted classes for the data represented by *TestData*, a parameter of *svmclassify* function.

Section III of this paper presents results obtained with SVM classifier in a DfPL scenario.

III. PROBABILITY NEURAL NETWORK (PNN)

The main application area of Probability Neural Networks (PNN) is pattern classification. PNN is based on Bayes theory which requires probability density functions (PDF). The PDFs are constructed using Parzen Windows [17]. PNN uses a supervised learning process and develops distribution functions within a pattern layer. The distribution functions are responsible for estimating the likeliness for an input vector. Further, priori probabilities (relative frequencies) are used to group learned patterns and to determine the class a given input vector belongs to. The input vector is classified based on the shortest Euclidian distance between inputs and distribution functions specific to a class. PNN architecture consists of three layers: input layer, pattern layer and output layer. A fourth layer can be used in order to normalise the input vector if this was not normalised previously.

Parzen estimator approximates the Probability Distribution Functions (PDF) for each class. The Parzen estimator can accurately classify inputs if the training set is large enough. The distribution functions are estimated in the pattern layer where a neural Bayes classifier is implemented. Each input vector of a training set has a processing element allocated in the pattern layer and the output should have an equal number of elements. Otherwise the network can poorly classify the input vectors. Based on the training set, the patterns are programmed in the pattern layer. One output will be generated for each input vector with the highest match between inputs and the programmed patterns. Otherwise no output will be generated.

The training of a PNN is simpler compared to the feedforward back propagation network. The pattern layer can be very large if there are various values that classify each category. In order to solve a problem where we need to classify the input vectors in two classes A and B, we consider the Bayes rule that classifies an input belonging to class A is considered as:

$$P_A C_A f_A(x) > P_B C_B f_B(x) \quad (3)$$

where, P_A is the priori probability of instances of patterns in class A, C_A is the cost associated with classifying vectors and $f_A(x)$ is the PDF of class A.

The PDF is estimated in a Probability Neural Network by the equation:

$$f_A(x) = \frac{1}{(2\pi)^{n/2} \sigma^n} \frac{1}{m_n} \sum_{i=1}^{m_A} \exp\left[-2 \frac{(x - x_A)^T (x - x_{Ai})}{\sigma^2}\right] \quad (4)$$

where x_{Ai} represents i_{th} training pattern from class A, n is the dimension of input vectors and σ is the smoothing parameter (standard deviations of Gaussian distribution)

Considering the previous problem where we try to assign an input vector to one of two classes, A or B, the architecture of the PNN is showed in Figure [fig:Probability-Neural-Network].

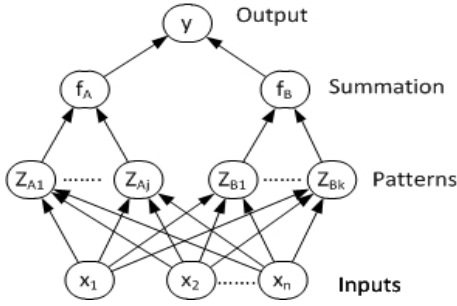


Fig. 1. Probability Neural Network classifier

Thus PNN can be used for classification of patterns. The structure in Figure [fig:Probability-Neural-Network] is an example of probabilistic neural networks that can be used to classify inputs such as WiFi signal strength. The Received Signal Strength Indicator (RSSI), used in wireless communications to represent signal strength, can be classified for example into one of the two classes: motion or no motion.

IV. EVALUATION

This section presents the experiment we conducted in order to detect motion in a DfPL scenario. First, the test bed shown in Figure 2 will be described and then Support Vector Machine classifier will be used to analyse/classify motion. Finally, we compare the classification errors for various kernel functions in Table I.

The project focuses on deploying a DfPL system on top of a Wireless Sensor Network (WSN). The first step towards implementing such a system is filtering data using a selected smoothing algorithm, SavitzkyGolay smoothing filter in this case. The filtered data is fed to a SVM classifier in order to detect movement. The next step is using classified data, timestamps and links affected as parameters in a decision making algorithm that will compute a person's location or return the number of people detected. The project aims towards multi-occupancy detection in a DfPL scenario.

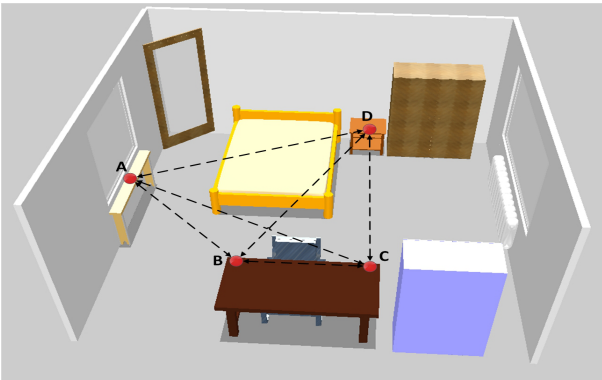


Fig. 2. The test bed with bidirectional link selected.

We have collected the data in a room of size 3.6m by 3.4m. A Wireless Sensor Network (WSN) based on four IEEE 802.14.5 Java Sun SPOT nodes and a base station was deployed in the environment. The data was recorded using a single thread collection over a period of approximately two hours. The data sets were obtained by observing Received Signal Strength Indicator (RSSI) levels when movements took place. We use two data sets containing 800 values (see Figure 3) selected from the two hours recordings. The first data set represents the training data while the second one is the test data. The nodes are broadcasting messages every 200 ms. When the messages are received, Received Signal Strength Indicator (RSSI) is added and then the messages are forwarded to the base station. However, working with a single collection thread can cause delays as the base station collects data from one node at a time.

In the case of four nodes, considering that we collect data from 12 links, we experienced delays in the collection speed. However the delays were not large enough to affect our tests. For larger test beds, multiple collection threads or more than one base station will improve the collection speed.

We have selected one bidirectional link between nodes A and B as shown in Figure 2. Both links are considered to be independent. Figure 3 shows the raw data collected from the selected links. We do not use data collected from all 12 links as the scope of the paper is to classify motion on two selected links (bidirectional communication between nodes A and B). SVM classification will perform in a similar manner on any selected link.

Both data sets are smoothed in order to filter noise. The derivative of the signal is used to normalise the data. Figure 4 shows the smoothing and derivative on one of the links. Data from the second link is processed in a similar manner. It is necessary to normalise the data in order to train and use classifiers.

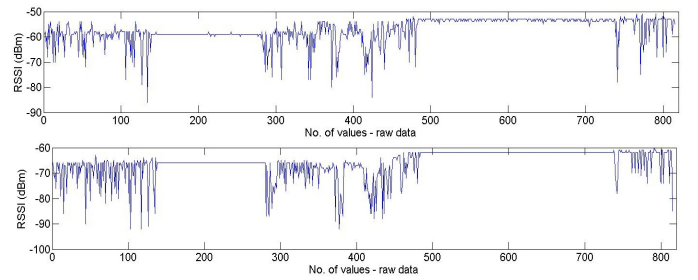


Fig. 3. Raw data from two selected links.

Figure 5 shows the threshold selection considering the normalised data. The value used in this case was ± 2 . Any other value above or below this threshold is considered an event which will be classified as motion. The threshold is dependent on the environment. In very noisy environments we need to modify this threshold. Thus a calibration depending on the level of noise in the environment is required.

Figure 6 shows targets vector and predicted classes using SVM with the Polynomial kernel function. One link is used to train the classifiers while the data recorded on the second link represents the test vector. The targets vector is obtained

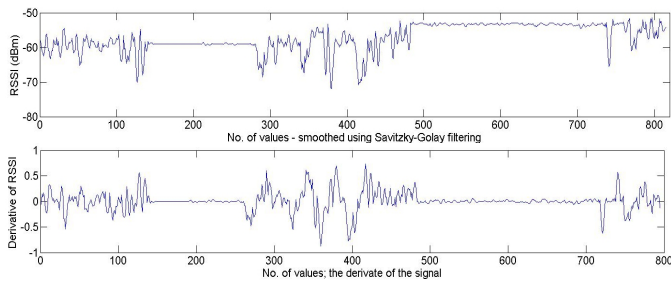


Fig. 4. Smoothing and derivative of one link.

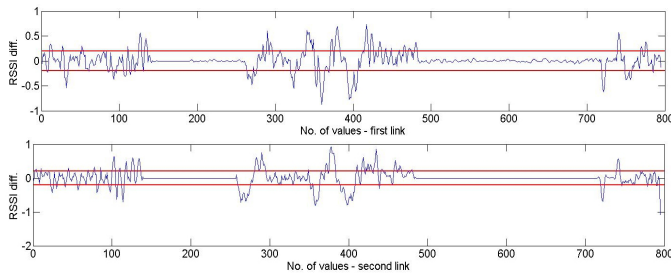


Fig. 5. Threshold selection on the derivative of the data.

by analysing the data based on the threshold chosen above. Afterwards the test data is fed to the classifier and the output is compared with the targets vector. Due to the limited space available, targets and predicted class for other kernel functions will not be added. Figure 6 represents an example of SVM classification. As one may notice the data is classified into two classes: 'No motion' (value 1) and 'Motion' (value 2).

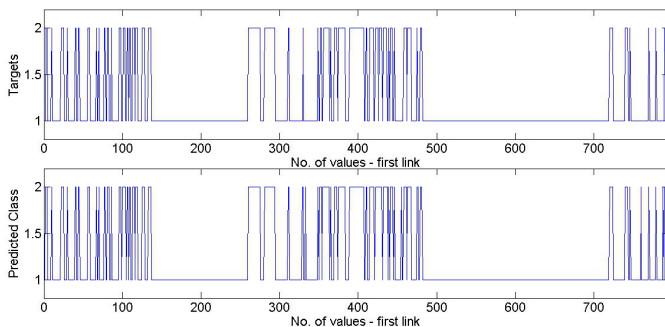


Fig. 6. Targets and predicted classes using SVM with Polynomial kernel function.

Table I shows the errors obtained in the classification process. Considering the number of values we have used for training, we can conclude that the SVM classifier performed well.

The Polynomial kernel function performed better for RSSI measurements compared to other functions and to PNN. The Mean Square Error (MSE) was used to calculate the error between targets and predicted classes. The MSE obtained for SVM with Polynomial function was 0.0201.

V. CONCLUSION

In this paper we presented Support Vector Machine and Probabilistic Neural Network classifiers that enable motion

TABLE I
CLASSIFICATION ERRORS FOR TEST DATA

SVM Kernel Function	Error
Linear	0.1635
Quadratic	0.0226
Polynomial	0.0201
RBF	0.0252
MLP	0.1686
Probabilistic Neural Network	
PNN	0.1736

detection in a DfPL scenario. Various SVM kernel functions such as Linear, Quadratic, Polynomial, Gaussian Radial Basis Function and Multilayer Perceptron, and an implementation of PNN were used to process wireless signal strengths in order to detect motion. The results showed the possibility of using classifiers in order to detect multi-occupancy using DfPL considering the timestamps and links affected by human presence as parameters. The usage of timestamps and links in order to decide upon the number of people in the monitored environment is considered as future work. A person cannot affect wireless links covering different areas in the environment at the same time. We analysed a bidirectional communication between two nodes in the deployed WSN. Further, more complex classifiers will be analysed in order to obtain a high accuracy motion detection.

ACKNOWLEDGEMENT

This work is supported by a Vice-Chancellor's Research Scholarship from the University of Ulster. We would like to thank Oracle for their help in supplying Java Sunspot Development Kits. It is also a pleasure to thank those who helped and supported me. I am grateful to my family for their support. I would particularly like to thank my friends Marian Baciu and Sorin Vreme for their friendship and guidance. This paper is dedicated in loving memory of my father and my friend Iulian Stoi.

REFERENCES

- [1] L. Frazier, "Surveillance through walls and other opaque materials," *IEEE Aerospace and Electronic Systems Magazine*, vol. 11, no. 10, pp. 6 – 9, oct 1996.
- [2] L. Ma, Z. Zhang, and X. Tan, "A novel through-wall imaging method using ultra wideband pulse system," in *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IHH-MSP '06. International Conference on*, dec. 2006, pp. 147 – 150.
- [3] J. Wilson and N. Patwari, "See-through walls: Motion tracking using variance-based radio tomography networks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 5, pp. 612 – 621, may 2011.
- [4] F. Aryanfar and K. Sarabandi, "Through wall imaging at microwave frequencies using space-time focusing," *Antennas and Propagation Society International Symposium, 2004. IEEE*, vol. 3, pp. 3063 – 3066 Vol.3, june 2004.
- [5] E. Gazit, "Improved design of the vivaldi antenna," *Microwaves, Antennas and Propagation, IEE Proceedings H*, vol. 135, no. 2, pp. 89 – 92, apr 1988.
- [6] M. Valtonen, J. Maentausta, and J. Vanhala, "Tiletrack: Capacitive human tracking using floor tiles," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, march 2009, pp. 1 – 10.
- [7] J. Krumm, *Ubiquitous Computing Fundamentals*. CRC Press, 2010.

- [8] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, "Multi-camera multi-person tracking for easyliving," in *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, 2000, pp. 3 – 10.
- [9] Microsoft Research, "Easy Living," <http://www.research.microsoft.com/>, 2011.
- [10] G. Deak, K. Curran, and J. Condell, "Filters for RSSI-based measurements in a Device-free Passive Localisation Scenario," *International Journal on Image Processing & Communications*, vol. 15, no. 1, pp. 23 – 34, 2011.
- [11] —, "History Aware Device-free Passive (DFP) Localisation," *International Journal on Image Processing & Communications*, vol. 16, no. 3 - 4, pp. 21 – 30, 2012.
- [12] M. Moussa and M. Youssef, "Smart cevides for smart environments: Device-free passive detection in real environments," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, march 2009, pp. 1 – 6.
- [13] M. Youssef, M. Mah, and A. Agrawala, "Challenges: device-free passive localization for wireless environments," *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pp. 222 – 229, 2007.
- [14] A. Kosba, A. Abdelkader, and M. Youssef, "Analysis of a device-free passive tracking system in typical wireless environments," in *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, dec. 2009, pp. 1 – 5.
- [15] L.-P. Song, C. Yu, and Q. H. Liu, "Through-wall imaging (twi) by radar: 2-d tomographic results and analyses," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 43, no. 12, pp. 2793 – 2798, dec. 2005.
- [16] Mathworks, "R2012a Documentation, Bioinformatics Toolbox," <http://www.mathworks.co.uk/help/toolbox/bioinfo/>, 2012.
- [17] S. Sumathi and P. Surekha, *Computational Intelligence Paradigms Computational Intelligence Paradigms Theory and Applications*, 2010.