INL/CON-07-12266
PREPRINT

Intelligent Control in Automation Based on Wireless Traffic Analysis

2007 International Joint Conference on Neural Networks

Kurt Derr
Milos Manic

August 2007

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance

Intelligent Control in Automation Based on Wireless Traffic Analysis

Kurt Derr
Idaho National Laboratory
2525 Freemont Avenue
Idaho Falls, ID 83415, USA
derr5843@uidaho.edu
Kurt.Derr@inl.gov

Milos Manic
Department of Computer Science
University of Idaho at Idaho Falls
1776 Science Center Dr., Ste.306
Idaho Falls, ID 83402, USA
misko@uidaho.edu

Abstract – Wireless technology is a central component of many factory automation infrastructures in both the commercial and government sectors, providing connectivity among various components in industrial realms (distributed sensors, machines, mobile process controllers). However wireless technologies provide more threats to computer security than wired environments. The advantageous features of Bluetooth technology resulted in Bluetooth units shipments climbing to five million per week at the end of 2005 [1, 2]. This is why the real-time interpretation and understanding of Bluetooth traffic behavior is critical in both maintaining the integrity of computer systems and increasing the efficient use of this technology in control type applications. Although neuro-fuzzy approaches have been applied to wireless 802.11 behavior analysis in the past, a significantly different Bluetooth protocol framework has not been extensively explored using this technology. This paper presents a new neuro-fuzzy traffic analysis algorithm of this still new territory of Bluetooth traffic. Further enhancements of this algorithm are presented along with the comparison against the traditional, numerical approach. Through test examples, interesting Bluetooth traffic behavior characteristics were captured, and the comparative elegance of this computationally inexpensive approach was demonstrated. This analysis can be used to provide directions for future development and use of this prevailing technology in various control type applications, as well as making the use of it more secure.

Keywords: Bluetooth traffic analysis, intrusion detection, computer security, neuro-fuzzy controller, classification, data mining, fuzzy control.

1. INTRODUCTION

Bluetooth (BT) is a low-power-consumption and short-range wireless technology for Personal Area Networks (PANs). BT connects personal electronic devices, such as laptops, mobile phones, digital cameras, audio equipments, and printers, without the clutter of cables. The number of BT products on the market doubled from 2003 to 2004 to an installed base of over 250 million growing to about 500 million units by the end of 2005 [1, 2]. Paired with this is the notable increase in consumer awareness of Bluetooth wireless technology. The astonishing fact that these products have a shipping rate of over 5 million per week [1, 2], gives rise to a vital issue of better understanding of traffic patterns of this technology. This understanding can further enable better use of the technology with respect to a variety of control applications as well as securing the traffic in mission critical applications. Wireless technology in factory automation has advantages in reducing overall costs through easier installation, engineering and reduced failure rate in operation [3]. Wireless can also increase productivity and ease maintenance by introducing mobility, flexibility and fast control network access for additional devices. Examples include sensors, actuators, various robotics and factory applications, etc.

Many manufacturers such as SENA [4], offer industrial Bluetooth products such as Bluetooth serial adapters, Bluetooth serial modules and bluetooth access points. These products are designed to replace RS-232 serial cables with wireless connectivity, ideal for industrial applications that require data logging, wireless sensor control and monitoring in production facilities.

Unfortunately, potential security risks that exist with Bluetooth as with other wireless technologies cannot be ignored, especially when it comes to mission critical factory automation scenarios, such are military productions. A malicious user might use Bluetooth's ability to connect with other devices to create a targeted attack. An attacker could connect to a factory Intranet by spoofing the host identity to circumvent an access control and assume the address and hostname of the trusted host. Another example would be to infect Bluetooth devices (which can typically happen in a crowded public place such as an airport or subway/train stations) with malicious code that seeks out specific information and spreads its malware to other Bluetooth devices, which may then be used in a factory environment and continue spreading malicious code [5].

While 802.11 types of wireless traffic data patterns have been widely researched for various control type applications, Bluetooth traffic with its unique features still remains an uncharted territory. Understanding traffic at the baseband, link management, and logical link control and adaptation (L2CAP) protocol layers of Bluetooth is unknown because specialized hardware and software are required to examine what the detailed traffic patterns look like. Real-time interpretation of BT traffic will set foundations for better and predictive capabilities, better knowledge of anomalies and signatures in mobile devices, data intrusion detection, and directions for the development of more efficient and secure future Bluetooth technology. While research with regards to Neuro-Fuzzy Controllers (NFCs) has been done in the past, the authors decided to present a new, simple yet effective, computationally inexpensive neuro-fuzzy approach for traffic data patterns understanding and predicting of real-time threats.

2. BACKGROUND

The understanding of Bluetooth wireless traffic data patterns is essential in various control type applications such as factory automation, as well as maintaining a secure Bluetooth communication. In this paper we present a new neuro-fuzzy algorithm for adaptive-predictive analysis of Bluetooth traffic data patterns. To the best of our knowledge neuro-fuzzy techniques have not being used for BT intrusion detection and prevention.

The Bluetooth serial link protocol is a popular profile for factory automation because it allows devices to connect to one another without the use of wires, replacing cables that traditionally connect equipment using the RS-232/RS-485 standard. Radio-Frequency identification (RFID) is used in complex factory automation projects requiring auto-identification of work-in-progress and supply chain management. Handheld RFID reader/writer devices now are capable of industrial Bluetooth connectivity to laptop, desktop, or PDA devices [16]. Wireless battery free sensors based on the Bluetooth radio are being used to eliminate sensor wire from factory production equipment [17]. Since the use of wireless technologies in a factory automation environment poses greater risks [18] than the use of wired technologies, constant monitoring of wireless traffic is essential as a secondary line of defense.

Wireless technologies, such as Bluetooth, WiFi, and ZigBee, present additional security challenges over wired networks because the communications are based on radio frequency and not limited to a wired segment of a network. Some examples of wireless security issues/vulnerabilities are eavesdropping, pairing attacks, location tracking, and cipher attacks.

Computer security typically relies on a challenging problem of profiling patterns of usage. Security itself includes fuzziness. For example, given a quantitative measurement, a range value or an interval can be used to denote a normal value. Any values falling outside the interval are considered anomalous to the same degree regardless of their different distances to the interval. Also, values inside the interval will be viewed as normal to the same degree. Fuzzy logic is important to computer security because introducing fuzziness to these quantitative features helps to smooth out the abrupt separation.

Bluetooth devices communicate with one another through applications that are defined as Bluetooth profiles. Some examples of BT profiles include cordless telephone, intercom, device synchronization, headset, local area network access, fax, file transfer, serial link, and printing, representing a wide gamut of Bluetooth technology uses. Each of these profiles provides additional features and functionality in the Bluetooth protocol stack. The likelihood of flaws in the implementation of this technology increases with the complexity of the protocol stack.

Intrusion detection systems represent an established technology in the area of computer security. Intrusion detection systems involve a range of technologies that are used in the detection, reporting, and correlation of different types of events. Intrusion detection may help mitigate risk by providing administrators or users with information on attempted or actual security events. As the ubiquity of wireless technologies increases in automation and control systems environments, intrusion detection becomes a necessary secondary line of defense to protect critical infrastructure control and manufacturing systems.

Artificial Neural Networks have been largely researched for intrusion detection for their inherent ability to effectively model network traffic [6, 7]. Neural networks can make decisions quickly and facilitate real-time detection and gain experience over time. The ability to classify network activity based on incomplete data is another advantage of using neural networks [8]. The main disadvantages of neural network based intrusion detection systems are the training requirements of the network and the "black box" nature of the network. Approaches that combine both neural and fuzzy may overcome these limitations by providing the ability to learn and to make decisions based on rules that overcome the black-box limitation of neural networks. The neural network portion of the system adapts to new network intrusions over time, overcoming the problems of strictly rule-based intrusion detection systems.

Fuzzy logic has been largely researched for intrusion detection for its inherent ability to describe uncertain, gray areas of system behavior in intrusion detection. Fuzzy control schemes have shown to be effective in reducing excessive data loss for high bandwidth applications [9]. Fuzzy clustering schemes have been used for intrusion detection in wired environments. Process, system, and network state information have been captured and analyzed using the Fuzzy c-Medoids algorithm to detect anomalous behavior [10, 11, 12, 13, 14, 15].

The paper is organized as follows. Section 3 discusses the proposed adaptive-predictive neuro-fuzzy controller, section 4 describes the Traffic Pattern Intelligent Control Algorithm (TPICA), section 5 presents test examples and comparisons with similar applications in the MATLAB environment, and section 6 presents our conclusions.

3. ADAPTIVE-PREDICTIVE NEURO FUZZY CONTROLLER (APNFC)

Several neural-fuzzy models [19, 20] are noted in the literature; such as the Adaptive Neuro Fuzzy Inference System (ANFIS) [21], Neuro Fuzzy Controller (NEFCON) [22], Fuzzy Net (FUN) [23], and Self Constructing Neural Fuzzy Inference System (SONFIN) [24]. The integration of neural and fuzzy achieves synergistic effects over an architecture based strictly on neural or fuzzy approaches.

The two major phases of the presented APNFC are 1) data mining of the fuzzy inference system, and 2) mapping of extracted knowledge to a fuzzy realm, i.e. fuzzy controller design (the algorithm is explained in details in following sections).

The first (data mining) phase of the NFC algorithm is performed via an effective and computationally inexpensive neural network algorithm (Figure 1). Some knowledge of BT traffic (natural groupings of data elements) is created as a result of this phase. This phase provides an understanding of

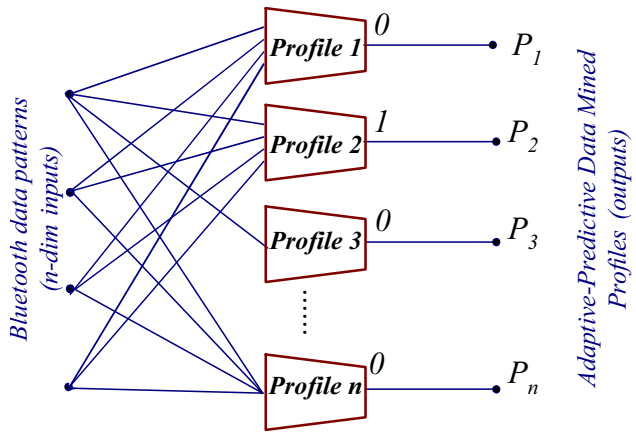

Fig. 1. Architecture for FCAN Knowledge Extraction

complex, multidimensional BT traffic data patterns, embellished in groupings of similar patterns, i.e. Bluetooth profiles. The data mining phase represents an ongoing process and enables adaptive, dynamic, and real-time learning capability of this system.

The second phase of NFC algorithm targets a procedure of automatic fuzzy rule induction. This phase entails mapping of each profile discovered in the previous phase to *m* number of fuzzy class descriptors (one for each of *m*-dimensions of profile data space). A fuzzy class descriptor entails an ensemble of fuzzy sets describing the certain profile for one dimension. Next, fuzzy sets are automatically combined into fuzzy rules. Each profile's center of gravity (COG) is compared against the peaks of discovered fuzzy sets. When a match is discovered with a certain fuzzy set, that set becomes a part of the AND predicate of a fuzzy rule. Final rules are in form of:

IF fuzzy_set1=set_name AND fuzzy_set2=set_name..AND...
THEN Cluster = cluster_number

The final fuzzy controller is executed in min-max Zadeh fashion. For the sake of simplicity, singleton weighting assignment presented in this paper is done under the assumption of a uniform data pattern distribution in one profile.

In the test examples section of this paper, the presented approach is compared against a numerical/statistical approach applied in the MATLAB environment.

The presented APNFC approach is the foundation for the derivation of the presented TPICA algorithm (*Traffic Pattern Intelligent Control Algorithm*). The TPICA algorithm accordingly to the APNFC approach consists of two phases. In the first, knowledge extraction phase, the authors present a simple yet effective *Forming Clusters As Needed* (FCAN) neural network algorithm. For the second phase of TPICA, the authors present the *Direct Mapping Fuzzy Logic Controller* (FLC) *Design*.

3.1. FCAN - Simple Knowledge Extraction Algorithm

This is the first phase of Neuro-Fuzzy Controller. This phase is executed by a simple yet adaptive unsupervised clustering algorithm based on the idea of *Forming Clusters As Needed* (FCAN). A single layer neural network is based on the following weight update formula:

$$\mathbf{W}_k = \frac{IPF \cdot \mathbf{W}_k + \alpha \mathbf{X}}{m+1} \quad (1)$$

where *IPF* is an importance factor, based on the number of patterns already belonging to a cluster *k*, and alpha is a weight constant defining the importance of input pattern *X*. The weight set for a cluster *k* is therefore based on a previous weight vector, number of belonging patterns, and a newly added pattern to that cluster. The attracting radius is based on a Euclidian Distance (*ED*):

$$ED = \sqrt[2]{\sum_{i=1}^{m} (x_i - w_i)^2} \quad (2)$$

between input pattern *x* and an m-dimensional cluster identifying neuron with weights w_i,

As a result, a set of clusters is identified:

$$C = \{C_i \mid i = 1,2,...n\} \quad (3)$$

where *n* is the number of clusters recognized. Then, a *2*-tuple containing a center of gravity and radius of a cluster is associated with each cluster:

$$\forall C_i, \quad (COG_i, R_i), \quad i = 1,2,...,n \quad (4)$$

where the center of gravity is an *m*-dimensional point in data space:

$$COG_i = \{x_{ij} \mid i = 1,2,...,n, \ and \ \ j = 1,2,...,m\} \quad (5)$$

where *n* is the number of clusters and *m* is the space dimensionality. A neural network architecture used for FCAN knowledge extraction is illustrated by Fig.1.

Unlike other algorithms, such as *k*-means and *c*-means [25, 26, 27], neither the seeded number of clusters nor cluster seeds are required. In this way a biased clustering is avoided, i.e. the algorithm produces clusters based on data only, and not based on initial parameters.

Traditional problems such as an unpredictive number of iterations and different results after each run are alleviated in this approach. This simple, computationally inexpensive algorithm requires a single iteration only. Comparative unsupervised clustering algorithms require iterations until stabilizing into a final detection arrangement (Kohonen WTA algorithm) [28, 29]. Typically, clustering algorithms depend on an order of patterns applied, causing a different result with each run. Given a same data set, this algorithm with weight constant alpha set to 1 will produce the same result each time for every run of the algorithm.

This algorithm typically detects convex shape spaces only, where the radius intensity is driven by the furthest pattern belonging to a cluster. However, through recursive application of the FCAN algorithm, convex sub areas of the initial cluster are being detected, constantly reducing the space detected versus the space occupied by patterns (see Figure 13). The union of the outer edges of these subclusters demonstrates that an arbitrary shape can be detected. This makes the FCAN algorithm very suitable for hardware implementation – the same implemented algorithm can be applied over and over, depending on the required detection resolution.

The only fixed parameter is the maximum cluster radius value. However, having a certain understanding of the measured data, this parameter can be heuristically determined. For example, a number of parameters in Bluetooth technology have a maximum upper limit; e.g., 79 RF channels, 1600 hops/second for frequency hopping, 7 active slaves in a piconet, 16 packet types, and 65,567 individual bits in an L2CAP packet.

3.2 Direct Mapping Fuzzy Logic Controller (FLC) Design

This is the second phase of Neuro-Fuzzy Controller. In this phase, knowledge extraction is done via neuron to fuzzy mapping. Fuzzy logic controller design is based on two factors: the shape of detected clusters and the weighting of inner cluster space. As explained in the previous section, the algorithm can perform detection of an arbitrary shaped cluster. With regards to weighting, the authors will first show the approach based on uniform arrangement of patterns. Then the improved approach that directly reflects the inner behavior embellished by the similarity of patterns within each cluster will be presented. Finally, a complexity comparison with the numerical approach will be presented.

In this phase a mapping from cluster to fuzzy space is performed:

$$S_M : S_c \rightarrow S_F \tag{6}$$

This is mapping from *m*-dimensional cluster space (S_C) to 2-dimensional fuzzy space (S_F). This dramatic dimensionality reduction offers significant computational complexity reduction while preserving the system data intricacy. Patterns classified as belonging to more than one cluster preserve this characteristic through the overlap of computationally simple fuzzy sets.

For each cluster, there exists a following mapping:

$$C_M : C_i \rightarrow F_{CD(j)} \tag{7}$$

where $i=1,2,...,n$, and $j=1,2,...,m$. Each cluster is mapped to *m* fuzzy class descriptors F_{CD}, where *m* is the data space dimensionality. For example, for a single cluster in 3-dimensional space, three fuzzy class descriptors would exist: $F_{CD(x)}$, $F_{CD(y)}$, $F_{CD(z)}$, for each of *x, y, z* dimensions.

Each fuzzy descriptor is further decomposed into *n* fuzzy sets (*FS*), where *n* is the number of identified clusters, as illustrated by Figure 13:

$$F_{CD(j)} = \{FS_j \mid j = 1,2,...,n\} \tag{8}$$

For weighting, an approach similar to a Zadeh or Takagi-Sugeno controller [30] is used. Singletons are created using a normalized weighting approach. The approach goes as follows. For each cluster C_i exists a radius R_i, and number of symmetric weight areas *N*:

$$R_i = \{r_i \mid i = 1,2,...,N\} \tag{9}$$

where weights are determined as:

$$w_j = w = \frac{1}{N}, \ \sum w_j = 1 \tag{10}$$

Then, each contour *j* is defined by the space defined by the difference between radii:

$$r_j - r_{j-1} = \frac{R_i}{N} \tag{11}$$

where weight contour w_{con} is determined as follows:

$$w_{con} = w \cdot j, \ \ j = 1,2,...,N \tag{12}$$

With equidistant and proportionally weighted contours, this approach provides a instantaneous response, assuming a uniform Bluetooth traffic data pattern arrangement. However, for certain applications, this level of precision may not be sufficient.

The initial approach is further enhanced by the recursive application of the same algorithm. This way, a refined cluster resolution is achieved *(*Figure 2, 6). Given sufficient number of recursive iterations, this approach ultimately results in cluster recognition to a pattern which is crucial for high precision military applications, such as missile guidance. By enhancing the initially simplistic approach with equidistant and proportionally weighted contours, a recursive approach provides realistic traffic behavior detection based on density of contours.

4. THE TPICA ALGORITHM

The pseudo code for TPICA (Traffic Pattern Intelligent Control Algorithm) goes as follows.

Step 1:

Perform the FCAN algorithm. For all patterns, update the network weights. The result of this step is a finite number of convex clusters with a variable radius associated.

Step2:

Step2.1: For *n* number of clusters perform:

Step2.2: For *m* number of dimensions perform:

Perform a mapping of a cluster to one dimension according to an established protocol. For each dimension, each cluster is mapped into an ensemble of fuzzy sets (Eq.8). For the sake of simplicity in this paper, a uniform pattern distribution within a cluster is assumed. As a proof of concept, a simple uniform fuzzy set assignment is performed (as shown in Section 5).

The result of this step is a group of fuzzy sets (Fuzzy Class Descriptor), resulting from a single cluster mapping to a single dimension.

Go to Step2.2. (next dimension)

Step2.3: Perform grid weight assignment based on a chosen protocol. Again, as a proof of concept and with the uniform pattern distribution within a cluster assumption, a set of symmetric concentric contours is used as a weight assignment protocol.

Go to Step2.1. (next cluster)

Step3:

Perform grid weight assignment based on a chosen protocol. For the sake of simplicity, in this paper, a uniform pattern

distribution within a cluster is assumed. Hence a set of symmetric concentric contours is used as a weight assignment protocol.

Step4:

Run the APNFC controller for a newly observed pattern. Go back to **Step 1** (adapt the existing knowledge with newly observed data).

5. TEST EXAMPLES

A frame represents a packet transmitted in Bluetooth between master and slave devices. As previously noted there are a maximum of 79 Bluetooth frequency channels. The number of channels used by an application may depend on the quality of service (QoS), packet size requirements, and the type of traffic represented by the application.

5.1 Datasets

Each Bluetooth packet may encapsulate data and control information from multiple layers in the protocol stack; e.g., baseband, link management, L2CAP, RFCOMM, OBEX, and OPP. Attributes for these layers include: baseband - role, channel, clock, flow, type, am_addr, L2CAP_flow, logical link ID, seqn, arqn, and payload length; link management - role, address, op_code, and transaction ID; L2CAP - role, address, pdu length, channel id, code, identifier, command length, protocol, and source channel id. Mobile devices from various manufacturers may use different layers of the protocol stack for the same application. Therefore a Bluetooth dataset captured from sending and receiving images between specific manufacturer mobile devices may have some variation in the protocol stack when compared to mobile devices from other manufacturers.

The training and testing data sets used in this experiment are based on the file transfer profile/protocol. Other profiles, such as personal area networking will have different bandwidth requirements and may utilize fewer channels. Frame number and channel number are captured for a series of files transferred between Bluetooth master and slave devices. Frame and channel number are basic attributes that will be present in the protocol exchange between Bluetooth devices regardless of manufacturer. Data and control information from the baseband, link management, L2CAP, RFCOMM, OBEX, and OPP may be included in the data capture as well.

5.2 Testing

Figure 2 shows the top level run of the TPICA algorithm of the frame and channel data. The cluster inner space at this point is weighted based on a weighting of concentric contours.

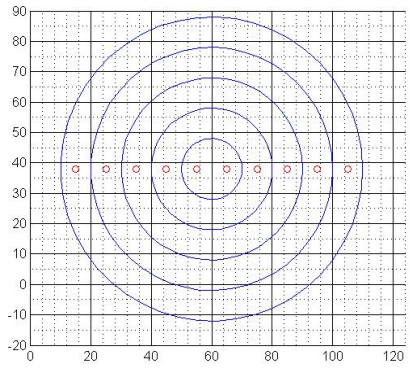

Figure 2. . Top Level Profiling of Baseband Frame and Channel Number Fuzzy Sets

At this level, fuzzy sets are evenly distributed with regards to concentric circles. A mapping is performed based on the scheme described by Figure 13. The initial fuzzy set is being decomposed according to a concentric scheme into five evenly distributed fuzzy sets (Figure 3). Greater fidelity can be achieved by increasing the number of fuzzy sets.

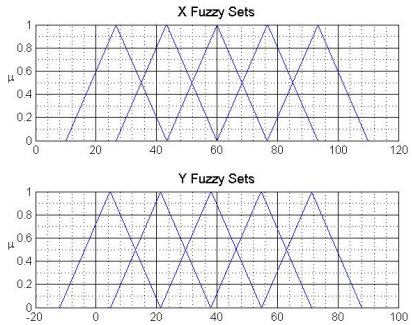

Figure 3. Fuzzy Sets for Top-Level Profile

Profiling at this level refers to a complete set of data. A comparison of neural network output with the FCAN output is given at Figure 4.

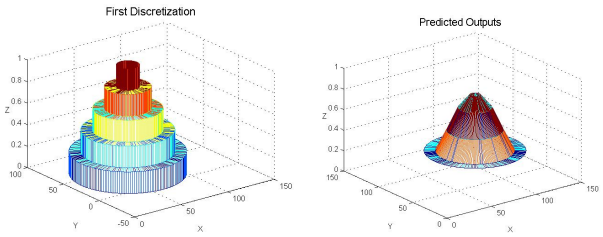

Figure 4. Testing of TPICA on Profile 1 Data

The second level discretization (profiling) of Bluetooth data traffic is shown in Figure 5, with the outputs of the FCAN algorithm shown at Figure 6. These results are based on the first part of the algorithm (FCAN) only (the Z axis represents the output values based on the neural network profiling).

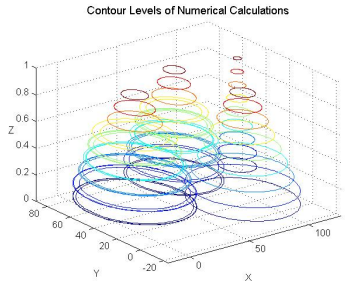

Figure 5. Output of FCAN Algorithm for Top-Level Profile

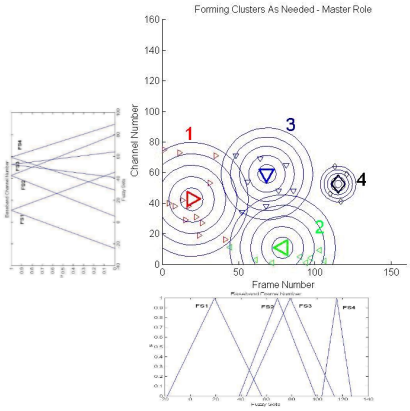

Figure 6. . Second Level Profiling – Four Profiles Detected

A second level profiling results in four profiles as illustrated by Figure 6. These clusters are furthered mapped to four fuzzy sets, for each dimension. Figure 7 illustrates the output of the first phase of the TPICA algorithm.

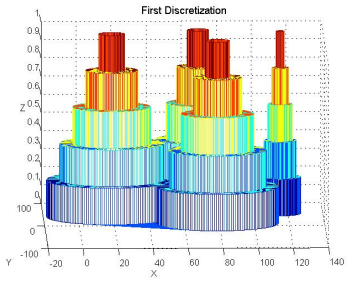

Figure 7. Output of FCAN Algorithm for Second-Level Profiles

The predicted outputs from the TPICA algorithm for the second level profiling as well as the result of a numerical, traditional approach to profiling are illustrated by Figure 8. This approach is based purely on averaging of neighboring levels from Figure 6. Although numerically more expensive (the range between two levels needs to be averaged with sufficient number of points), the surface obtained in a traditional approach is not as smooth as the surface obtained with a fuzzy approach.

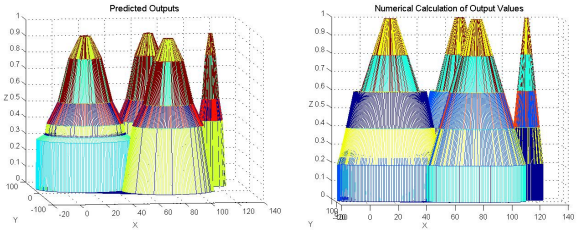

Figure 8. Second Level Profiling of Four Clusters Based on TPICA and Traditional (Numerical) Approach

The outputs of the first and final phase of the TPICA algorithm resulting from testing of these four profiles are on the left hand side of Figure 8. The resulting surface is smooth resulting from a fuzzy controller providing continuity of the output surface.

In two dimensional space, the fuzzy approximation depicted by Figure 9 occurs.

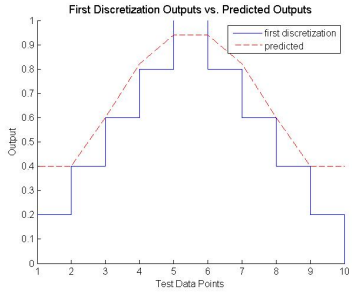

Figure 9. Two-Dimensional Illustration of Fuzzy Approximation of Neural Clustering

The fuzzy logic controller execution is illustrated by Figure 9 and Eq.13:

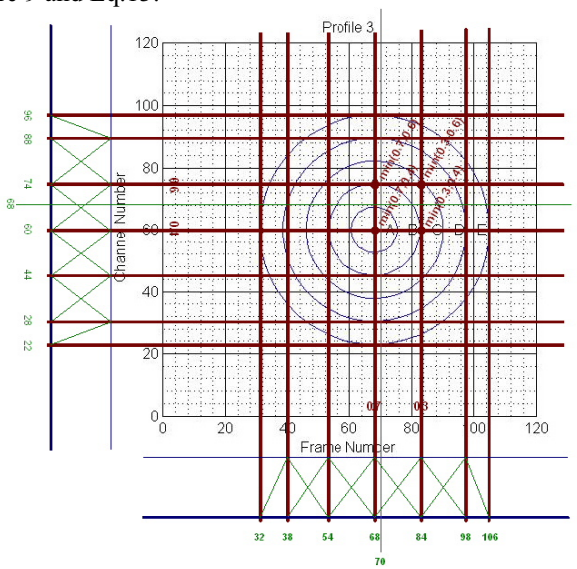

Fig. 10. Fuzzy Sets from Previous Step Being Decomposed into Fuzzy Class Descriptors – Example for Profile 3

$$(70,68) \xrightarrow{predicted} \sim 0.8 \qquad (13)$$

$$70 \rightarrow 0.7_{(68)} \,\&\, 0.3_{(84)};\ 68 \rightarrow 0.6_{(74)} \,\&\, 0.4_{(60)}$$

$$out = \frac{0.4 \cdot 1_A + 0.6 \cdot 0.7_{B-C} + 0.3 \cdot 0.6_C + 0.3 \cdot 0.7_{B-C}}{0.4 + 0.6 + 0.3 + 0.3} = 0.75625$$

Further enhancement of the controller is achieved through recursive application of the TPICA algorithm to each profile. Figure 8 shows the first discretization produced by applying the FCAN algorithm to the top-level profile (cluster 1), resulting in four sub clusters.

Similarly, to a second level fuzzy set decomposition, a third level fuzzy set decomposition can be performed (Figure 11):

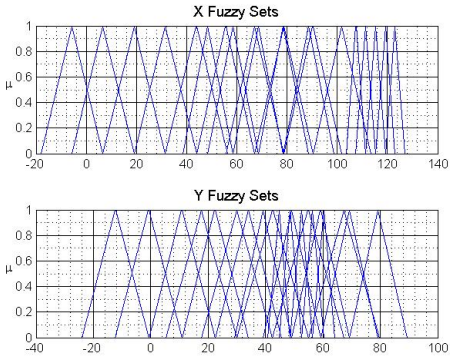

Figure 11. Fuzzy Sets for Second-Level Profiles

This is a result of subprofiling of each of the four profiles from the second level decomposition. As a result of automatic rule extraction phase, the following knowledge base (fuzzy rule base) is data mined:

R1: IF FS1 = Frame_FS1 AND FS2 = Channel_FS2, THEN CLUSTER = Cluster_1

R1: IF FS1 = Frame_FS3 AND FS2 = Channel_FS1, THEN CLUSTER = Cluster_2

R1: IF FS1 = Frame_FS2 AND FS2 = Channel_FS4, THEN CLUSTER = Cluster_3

R4: IF FS1 = Frame_FS4 AND FS2 = Channel_FS3, THEN CLUSTER = Cluster_4

This is the initial cut. Later on, each rule will be decomposed into k^2 rules, k being the number of concentric contours, i.e. areas with the same weight within a cluster. Each profile's center of gravity (COG) is compared against the peaks of discovered fuzzy sets. When a match is discovered with a certain fuzzy set, that set becomes a part of an AND predicate of a fuzzy rule in creation.

The process of decomposing rules on three levels is illustrated by example in Figure 12. The top level cluster (first profile) is mapped to a FS_1 fuzzy set. Based on the next level profiling (cluster 1.2), FS_1 is decomposed into new three fuzzy sets (2nd level). In the 3rd level profiling, $FS_{1.2}$ is further decomposed into three fuzzy sets, where the central one reflects the central profile of the 3rd level.

Through further recursive application, each fuzzy set is decomposed into an ensemble of fuzzy sets, i.e. fuzzy class descriptors.

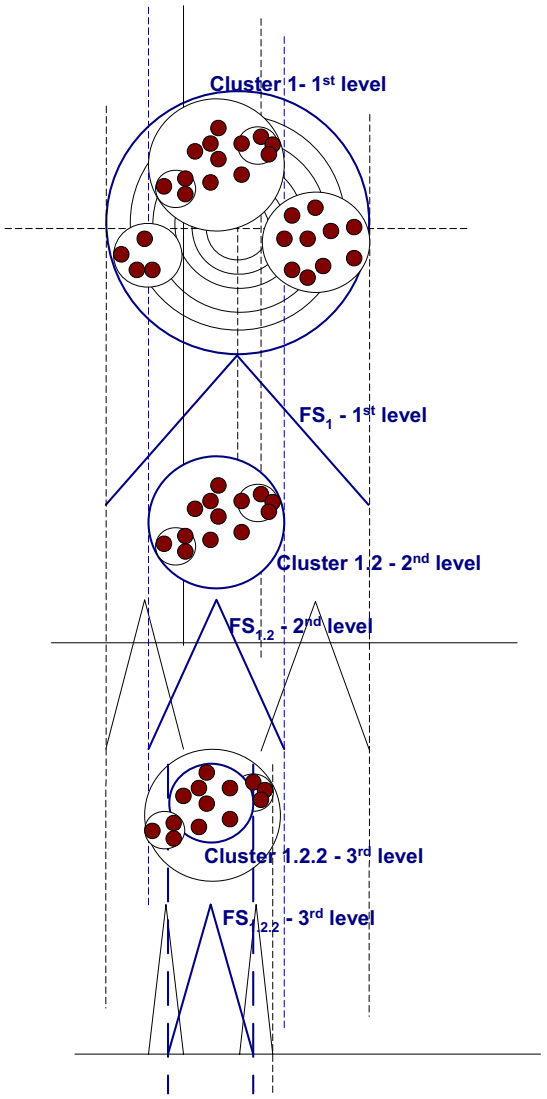

Figure 12. Illustration of Three Levels of Fuzzy Set Decomposition

6. CONCLUSION

While neuro-fuzzy controller approaches have been researched for wireless traffic, the new territory of Bluetooth protocols have not been investigated yet. This paper presents a TPICA algorithm, simplistic and elegant recursive application of computationally inexpensive adaptive neuro-fuzzy controller, used for the predictive traffic analysis of the ubiquitous new Bluetooth technology.

For the purpose of testing the data set referring the Bluetooth file transfer protocol was chosen for further analysis. The presented TPICA algorithm revealed 4 typical, general profiles. Analysis of the first profile (cluster 1) revealed that file transfer protocol typically occurs with frames in the range 1-50, that occur on base band channels 10 to 79. This means that during the initialization period, channels are evenly used.

As the packet exchange continues, channels used for ftp communication tend to localize into two or three clusters. The data can be further analyzed for other protocols and protocol parameters.

For real time response, this controller can be trained and designed off-line. For the execution phase the response is instantaneous - therefore making it suitable for real time applications with the previous data history available.

The algorithm can be extended to a multidimensional space (without a visualization option). Future work will also include different strategies for cluster profiling. Recursive sub-clustering can discover sub-profiles, where each can be mapped to a new fuzzy set. To avoid extensive overlap, near fuzzy sets can be combined together based on a degree of their similarity. Profiling (first phase of TPICA algorithm), can be investigated using different algorithms, such as a Counter-Propagation Network (CPN). Type-2 instead of type-1 fuzzy sets will also be investigated for the profile weighting.

7. REFERENCES

[1] "Bluetooth SIG Announces Best of CES Contest Winners", Bluetooth SIG (Special Interest Group) Press Release, URL from January 9, 2006: http://www.bluetooth.com/Bluetooth/Press/SIG/Bluetooth_SIG_Announces_Best_of_CES_Contest_Winners.htm,

[2] "Bluetooth Shipments Climb to Five Million Per Week", Bluetooth SIG (Special Interest Group) Press Release, URLfrom May 24, 2005: http://bluetooth.com/Bluetooth/Press/SIG/Bluetooth_Shipments_Climb_to_Five_Million_Per_Week.htm.

[3] Steigmann, R., Endresen, J., "Wireless networking seeks role in factory automation", http://wireless.industrial-networking.com/articles/articledisplay.asp?id=1357, URL from May 2007.

[4] SENA Technologies, "Industrial Automation", http://www.sena.com/solutions/factory_automation/, URL from May 2007.

[5] R. McMillan, "'BlueBag' PC sniffs out Bluetooth flaws",IDG News Service, URL from June 7, 2006: http://www.infoworld.com/article/06/06/07/79045_HNbluebag_1.html,,

[6] E. Casilari, A. Alfaro, A. Reyes, A. Diaz-Estrella, and F. Sandoval, 1998, "Neural modeling of Ethernet traffic over ATM networks," EANN '98, June, Gibraltar, pp. 304-307.

[7] A. Aussem, A. Mahul, and R. Marie, 2000, "Queueing Network Modeling with Distributed Neural Networks for Service Quality Estimation in B-ISDN Networks," Proceedings IEEEINNS- ENNS International Joint Conference on Neural Networks, Como, Italy, pp. 392-397.

[8] J. Cannady, Artifical Neural Networks for Misuse Detection, National Information Systems Security Conference, October 1998.

[9] H. Kazemian, L. Meng, A Fuzzy Control Scheme for Video Transmission in Bluetooth Wireless, Elsevier Science, 176 (2006), 1266 – 1289.

[10] H. Kazemian, L. Meng, Neuro-Fuzzy Control for MPEG Video Transmission Over Bluetooth, IEEE Transactions on Systems, Man, and Cybernetics, Volume 36, Number 6, November 2006

[11] H. Shah, J. Undercoffer, A. Joshi, Fuzzy Clustering for Intrusion Detection, IEEE International Conference on Fuzzy Systems, May 2003, pp. 1274-1278

[12] I. Suliman, J. Lehtomaki, I. Oppermann, Fuzzy-Based Intra-Piconet Scheduling in Bluetooth, Proceedings of the 2005 Finnish Signal Processing Symposium, August 25, 2005, pp. 54-57

[13] L. Cheng, I. Marsic, Fuzzy Reasoning for Wireless Awareness, International Journal of Wireless Information Networks, Vol. 8, No. 1, 2001

[14] H. Jin, J. Sun, H. Chen, Z. Han, A Fuzzy Data Mining Based Intrusion Detection Model, IEEE International Workshop on Future Trends of Distributed Computing Systems, May 2004

[15] John Dickerson, Julie Dickerson, Fuzzy Network Profiling for Intrusion Detection, Proceedings of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society, Atlanta, July, 301-306, 2000.

[16] "Escort Memory Systems Goes Mobile With the Element RFID Bluetooth Reader", http://news.thomasnet.com/fullstory/513190, March 16, 2007

[17] "Briefing: ABB's Trend with Wireless Automation Applications", http://www.arcweb.com/txtlstvw.aspx?LstID=c316d114-e794-4706-80fa-90750c261247&PrintPreview, October 9, 2006

[18] "Automation World", http://www.automationworld.com/view-3154, May 2007

[19] D. Nauck, R. Kruse, A Neuro Fuzzy Controller Learning by Fuzzy Error Propagation, Proceedings of the Conference of North American Fuzzy Information Processing Society, NAFIPS 1992, pp. 388-397.

[20] J. Mendel, Type-2 Fuzzy Sets and Systems: An Overview, IEEE Computational Intelligence, Vol. 2 No. 1, February 2007

[21] R. Jang, Neuro-Fuzzy Modeling: Architectures, Analyses and Applications, PhD Thesis, University of California, Berkeley, July 1992.

[22] D. Nauck and R. Kruse, NEFCON-I: An X-Window Based Simulator for Neural Fuzzy Controllers. In Proceedings of the IEEE International Conference on Neural Networks, Orlando, pp. 1638-1643, 1994.

[23] S.M. Sulzberger, N.N. Tschicholg-Gurman, S.J. Vestli, FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks, In Proceedings of IEEE Conference on Neural Networks, San Francisco, pp. 312-316, 1993.

[24] J.C. Feng and L.C. Teng, An Online Self Constructing Neural Fuzzy Inference Network and its Applications, IEEE Transactions on Fuzzy Systems, Vol 6, No.1, pp. 12-32, 1998.

[25] C. Bishop, Pattern Recognition and Machine Learning, Springer Science LLC., 2006, pp. 423-459.

[26] E. Cox, Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration, Morgan Kaufmann Publishers, 2005, pp. 207-263.

[27] J. Han, M. Kamber, Data Mining, Morgan Kaufmann Publishers, 2006, pp. 335-391.

[28] Kohonen, T. (1982) Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43:59-69.

[29] Kohonen, T. (1988) Self-Organization and Associative Memory, 2nd Ed. New York, Springer-Verlag.

[30] N. Mastorakis, General Fuzzy Systems as Extensions of the Takagi-Sugeno Methodology, WSEAS Transactions on Systems, April 2004, pp. 795 - 800