

Neural Network Applications in Sensor Fusion

Shantanu Sharma and M.S. Apurva
{shsharma,apurvams}@cse.iitk.ac.in

Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur-208016, INDIA

Guided By:

Dr. R. K. Ghosh

rkg@cse.iitk.ac.in

Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur-208016, INDIA

November 22, 2003

Abstract

Wireless Sensor Networks have emerged as a new information-gathering paradigm based on the collaborative effort of a large number of sensing nodes. This paper describes the application of neural network technology in the problem domain of sensor data fusion. The first section of this paper introduces and reviews the problems presented by sensor fusion. The second section provides the background on neural-network and sensor data fusion. The subsequent section discusses the domains where neural-network is applied for sensor data fusion varying as widely as intelligent waste-water management to military surveillance. We provide a model for wide-area surveillance using Neural Network based Sensor Data Fusion. We also discuss how fuzzy modifications to artificial neural networks can improve the confidence level of sensor fusion.

1 Introduction

Wireless Sensor Networks have emerged as a new information-gathering paradigm based on the collaborative effort of a large number of sensing nodes. In such networks, nodes deployed in a remote environment must self-configure without any a priori information about the network topology or global view. Nodes will act in response to environmental events and relay collected and possibly aggregated information through the dynamically formed multi-hop wireless network in accordance with desired system functionality.

These networks can form the basis for many types of smart environments such as smart hospitals, battlefields, earthquake response systems, and learning environments. A set of applications, such as biomedicine, hazardous environment exploration, environmental monitoring, military tracking and reconnaissance surveillance are the key motivations for the recent research efforts in this area.

Compact, portable, and inexpensive systems capable of quickly identifying contaminants in the field are of great importance when monitoring the environment. One approach is to combine a sensor array with a neural network. One advantage of this approach is that most of the intense computation takes place during the training process. Once the neural network is trained for a particular task, operation consists of propagating the data through the neural network.

The quantity and complexity of the data collected by sensor arrays can make conventional analysis of data difficult. ANNs, which have been used to analyze complex data and for pattern recognition, could be a better choice for sensor data analysis. A common approach in sensor analysis is to build an array of sensors, where each sensor in the array is designed to respond to a specific analysis.

2 Background

We divide this section into two parts, providing necessary background on *Sensor Data Fusion* and *Neural Networks* respectively:

2.1 Sensor Data Fusion

The task of sensor data fusion involves integration of numerous data streams, originating from separate sensors, into a consistent model that represents the pertinent highlevel features of the tactical environment and then to present an assessment of their significance. In the context of a typical modern surveillance environment, the data to be "fused" might consist of surveillance, reconnaissance and intelligence information as well operational and environmental data. A problem central to many data fusion systems is the need for rapid acquisition and interpretation of the information.

In a potentially hostile situation the time taken to perform this assessment is severely limited and a rapid and accurate response is vital. The task of data fusion in a military and surveillance applications is complicated by the additional problem of deception. As with conventional systems the incoming data may be subject to noise, interference, ambiguity and contradiction but in military and surveillance applications the data may be subject to deliberate deception by an enemy. This may take a number of different forms which a robust system must be capable of handling in an "intelligent" way.

2.2 Neural Networks

Artificial Neural Networks (ANNs) is an abstract simulation of a real nervous system that contains a collection of neuron units communicating with each other via axon connections. Such a model bears a strong resemblance to axons and dendrites in a nervous system. The McCulloch-Pitts model of neuron describes it as a binary device with each neuron has a fixed threshold logic. This model was derived from the works of John von Neumann, Marvin Minsky, Frank Rosenblatt, and many others.

Hebb postulated, in his classical book *The Organization of Behavior*, that the neurons were appropriately interconnected by self-organization and that "an existing pathway strengthens the connections between the neurons". He proposed that the connectivity of the brain is continually changing as an organism learns different functional tasks, and that cells assemblies are created by such changes. By embedding a vast number of simple neurons in an interactive nervous system, it is possible to provide computational power for very sophisticated information processing. 1

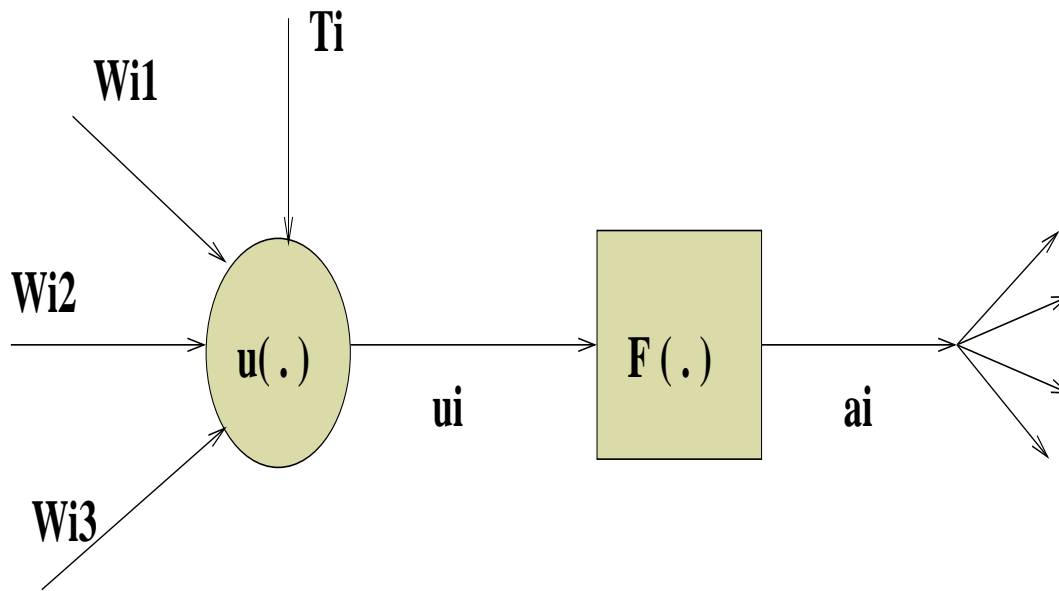


Figure 1: Illustrating Neural Network Model

2.2.1 Supervised and Unsupervised Neural Networks

Neural networks are typically classified in terms of their corresponding training algorithms: fixed-weights networks, unsupervised networks, and supervised networks. There is no learning required for the fixed-weight networks, so a learning mode is supervised or unsupervised.

- **Supervised Learning Rules:** Supervised learning networks have been the mainstream of neural model development. The training data consist of many pairs of input/output training patterns. Therefore, the learning will benefit from the assistance of the teacher. Given a new training pattern, say, $(m + 1)^{th}$, the weights may be updated as follows:

$$w_{ij}^{(m+1)} = w_{ij}^{(m)} + \delta w_{ij}^{(m)}$$

The following schematic illustrates the supervised learning approach:
2

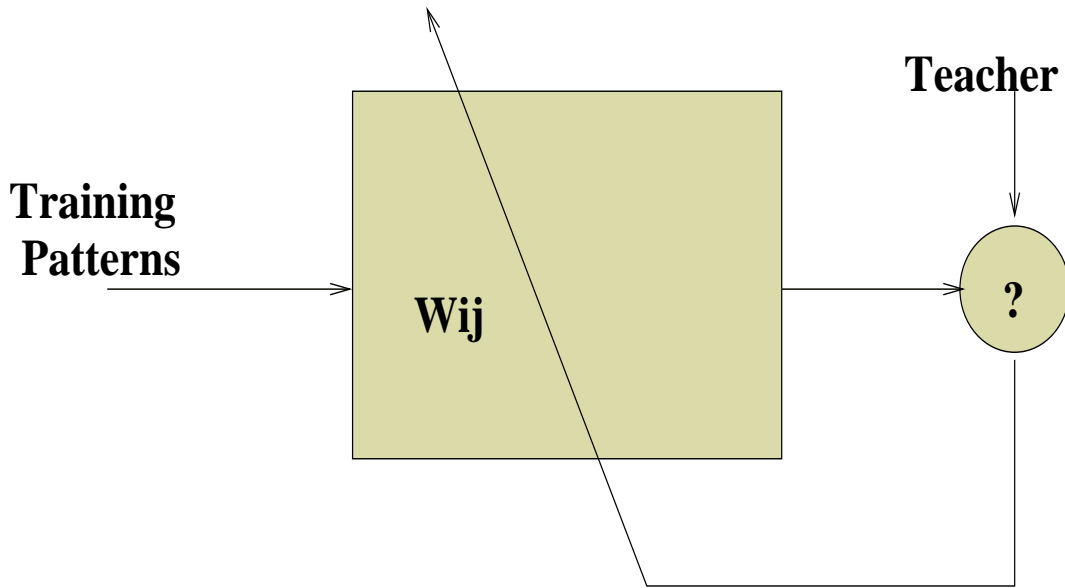


Figure 2: Illustrating Supervised Neural Network

Figure:

- **Unsupervised Learning Rules:** For an unsupervised learning rule, the training set consist of input training patterns only. Therefore, the network is trained without benefit of any teacher. The network learns to adapt based on the experiences collected through the previous training patterns. Typical examples are the Hebbian learning rule, and the competitive learning rule.

A simple version of Hebbian learning rule is that when unit i and unit j are simultaneously excited, the strength of the connection between them increases in proportion to the product of their activations.

As an example of competitive learning, if a new pattern is determined to belong to a previously recognized cluster, then the inclusion of the new pattern into that cluster will affect the representation (e.g., the centroid) of the cluster. This will in turn change the weights characterizing the classification network. If the new pattern is determined to belong to none of the previously recognized cluster, then the structure and the weights of the neural network will be adjusted to accommodate the new class (cluster).

Here is a typical schema of an unsupervised system, Figure: 3

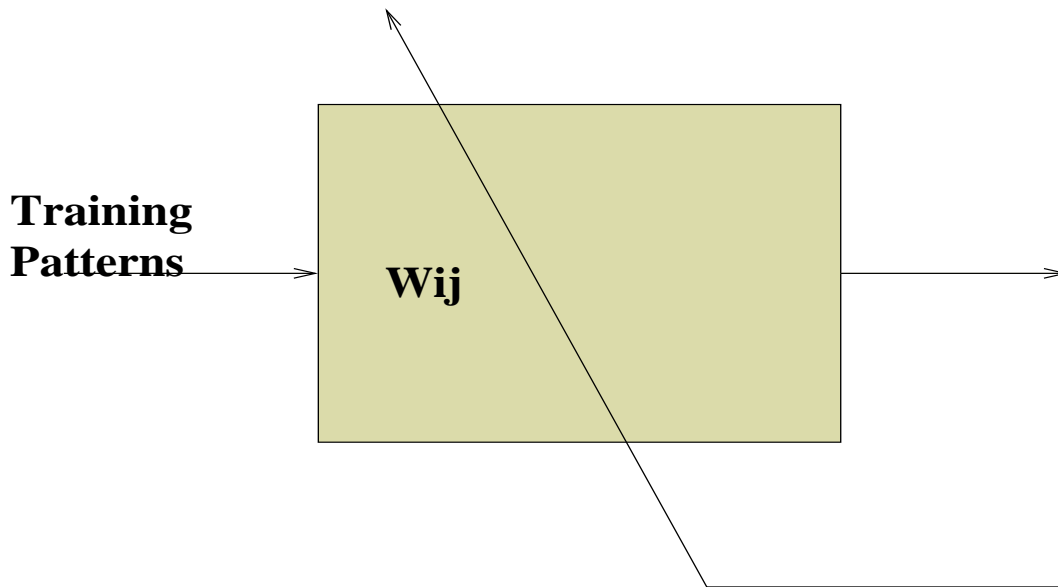


Figure 3: Illustrating Unsupervised Neural Network

A fuzzified neural network approach [1] based on a function of failed sensor is also found to be effective.

3 Applications of Neural Networks in Sensor Fusion

Neural Networks have been vastly applied in the field of Sensor Fusion. In this section we discuss various applications of Neural Networks in Sensor Data Fusion:

3.1 Autonomous Mobile Robots

In an autonomous mobile robot, it is essential that the robot represents its local environment. The local environment representation can be used for achieving lowlevel obstacle avoidance, for maintaining a global representation of environment and for determining the position of the robot in such a global representation. The robot typically acquires such a representation of its local environment by the use of its sensors.

However, measurements made by sensor are uncertain, erroneous and incomplete and the outputs of multiple sensors must be fused to obtain a more reliable representation. The key issue in sensor fusion is the conversion of the sensor data to a common “*sensor language*” before the actual fusion is performed. This conversion involves modelling of the sensors’ error characteristics.

At first the sensor’s performance was modelled with standard mathematical models but it has been generally acknowledged that these models are not sufficiently accurate. Therefore, several algorithms were devised to learn these models in a calibration phase. Neural network approach is found to be effective for learning of inverse sensor models. The main advantages of this approach are that the sensor model remains adaptive both to changes in the error characteristics of the sensor and to changes in the environment, which also influence the performance of a sensor.

The network is trained by using supervised approach. The main problem is how the learning samples for the network can be obtained. The network

can be trained with learning samples that do not specify the desired output of the network directly, but instead contain outputs whose expected values equal the desired outputs of the network. Such samples can indeed be provided for by the robot.

3.2 Intrusion detection

Intrusion detection schemes can be classified into two categories: misuse and anomaly intrusion detection. Misuse refers to known attacks that exploit the known vulnerabilities of the system. Anomaly means unusual activity in general that could indicate an intrusion. If the observed activity of a user deviates from the expected behavior, an anomaly is said to occur. Misuse detection can be very powerful on those attacks that have been programmed in to the detection system. However, it is not possible to anticipate all the different attacks that could occur, and even the attempt is laborious.

Some kind of anomaly detection is ultimately necessary. One problem with anomaly detection is that it is likely to raise many false alarms. Unusual but legitimate use may sometimes be considered anomalous. The challenge is to develop a model of legitimate behavior that would accept novel legitimate use. It is difficult to build such a model for the same reason that it is hard to build a comprehensive misuse detection system: it is not possible to anticipate all possible variations of such behavior. The task can be made tractable in three ways:

1. Instead of general legitimate use, the behavior of individual users in a particular system can be modelled. The task of characterizing regular patterns in the behavior of an individual user is an easier task than trying to do it for all users simultaneously.
2. The patterns of behavior can be learned for examples of legitimate use, instead of having to describe them by hand-coding possible behaviors.
3. Detecting an intrusion real-time, as the user is typing commands, is very difficult because the order of commands can vary a lot. In many

cases it is enough to recognize that the distribution of commands over the entire login session, or even the entire day, differs from the usual.

3.2.1 The NNID System

The NNID anomaly intrusion detection system is based on identifying a legitimate user based on the distribution of commands she or he executes. This is justifiable because different users tend to exhibit different behavior, depending on their needs of the system. Some use the system to send and receive e-mail only, and do not require services such as programming and compilation. Some engage in all kinds of activities including editing, programming, e-mail, Web browsing, and so on. However, even two users that do the same thing may not use the same application program. For example, some may prefer the vi editor to emacs, favor pine over elm as their mail utility program, or use gcc more often than cc to compile C programs. Also, the frequency with which a command is used varies from user to user.

The set of commands used and their frequency, therefore, constitutes a print of the user, reflecting the task performed and the choice of application programs, and it should be possible to identify the user based on this information. It should be noted that this approach works even if some users have aliases set up as shorthand for long commands they use frequently, because the audit log records the actual commands executed by the system. Users privacy is not violated, since the arguments to a command do not need to be recorded. That is, we may know that a user sends e-mail five times a day, but we do not need to know to whom the mail is addressed. Building NNID for a particular computer system consists of the following three phases:

1. Collecting training data: Obtain the audit logs for each user for a period of several days. For each day and user, form a vector that represents how often the user executed each command.
2. Training: Train the neural network to identify the user based on these command distribution vectors.
3. Performance: Let the network identify the user for each new command distribution vector. If the networks suggestion is different from the actual user, or if the network does not have a clear suggestion, signal an

anomaly. The particular implementation of NNID and the environment where it was tested is described in the next section.

3.2.2 Experiments

The NNID system was built and tested on a machine that serves a particular research group at the Department of Electrical and Computer Engineering at the University of Texas at Austin. This machine has 10 total users; some are regular users, with several other users logging in intermittently. Data was collected on this system for 12 days, resulting in 89 user-days. Instead of trying to optimize the selection of features (commands) for the input, we decided to simply use a set of 100 most common commands in the logs, and let the network figure out what information was important and what superfluous. Intelligent selection of features might improve the results some but the current approach is easy to implement and proves the point.

In order to introduce more overlap between input vectors, and therefore better generalization, the number of times a command was used was divided into intervals. There were 11 intervals, non-linearly spaced, so that the representation is more accurate at lower frequencies where it is most important. The first interval meant the command was never used; the second that it was used once or twice, and so on until the last interval where the command was used more than 500 times. The intervals were represented by values from 0.0 to 1.0 in 0.1 increments. These values, one for each command, were then concatenated into a 100-dimensional command distribution vector (also called user vector below) to be used as input to the neural network. The standard three-layer back-propagation architecture was chosen for the neural network.

The idea was to get results on the most standard and general architecture so that the feasibility of the approach could be demonstrated and the results would be easily replicable. More sophisticated architectures could be used and they would probably lead to slightly better results. The input layer consisted of 100 units, representing the user vector; the hidden layer had 30 units and the output layer 10 units, one for each user.

3.3 Multisensor Surveillance

In realistic surveillance scenarios, it is impossible for a single sensor to see all areas at once, or to visually track a moving object for a long period of time. Objects become occluded by trees and buildings and sensors themselves have limited fields of view. A promising solution to this problem is to use a network of video sensors to cooperatively monitor all objects within an extended area and seamlessly track individual objects that cannot be viewed continuously by a single sensor alone. Some of the technical challenges within this approach are to:

1. Actively control sensors to cooperatively track multiple moving objects;
2. Fuse information from multiple sensors into scene-level object representations;
3. Monitor the scene for events and activities that should trigger further processing or operator involvement; and
4. Provide human users with a high-level interface for dynamic scene visualization and system tasking.

3.3.1 Object Type Classification

Bottom-up motion detection and tracking algorithms (which do not try to fit a priori models to image data) view objects in the scene as moving blobs of pixels. The neural network is a standard three-layer network, trained using the backpropagation algorithm. Input features to the network are measured directly from the image blob and camera settings: blob dispersedness (perimeter area); blob area; blob aspect ratio and camera zoom value. There are four output classes: single human; human group; vehicle and clutter. This neural network classification approach is fairly effective for single image frames; however, one of the advantages of video is its temporal component.

To exploit this, classification is performed on each blob as it is tracked through the sequence of frames. The classification results for each frame are kept in a histogram and at each time step, the most likely class label for the blob is chosen based on all classifications that have been made for it.

3.4 Forecasting Rainfall and Floods

Advances in remote sensing tools range from weather radar to satellite observations. Extensive systems for rainfall estimation, like the NEXt generation RADar (NEXRAD) are in place in countries like the US. Weather radar measure quantities like reflectivity at spatially distributed scales, which in turn relate to rainfall rates. Satellites measure brightness temperatures, which are used for cloud classification and estimation of rainfall rates. Ground based measurement systems like rain gages measure point rainfall. These sensors collect vast amounts of information.

The power of information technology is leveraged to efficiently acquire, store, retrieve, and use this information. Methods based on Artificial Neural Networks (ANN) have been used by several researchers in recent years in the area of precipitation estimation and forecasting. These are complex data dictated tools that have been shown to act as universal function approximators, and converge faster than other traditional approximators. ANN based tools have shown promise in recent years for precipitation estimation and forecasting.

Accurate quantitative forecasting of rainfall for basins with a short response time is essential to predict streamflow and flash floods. Neural networks are used to develop a Quantitative Precipitation Forecasting (QPF) model that highly improved forecasting skill at specific locations in Pennsylvania, using both Numerical Weather Prediction (NWP) output and rainfall and radiosonde data. Besides using radiosonde and rainfall data, the model also uses the satellite-derived characteristics of storm systems such as tropical cyclones, mesoscale convective complex systems and convective cloud clusters as input.

The convective classification and tracking system (CCATS) is used to identify and quantify storm properties such as life time, area, eccentricity, and track. As in standard expert prediction systems, the fundamental structure of the neural network model was learned from the hydroclimatology of the relationships between weather system, rainfall production and streamflow

response in the study area. Here, the present results from the application of the Quantitative Flood Forecasting (QFF) model to forecast floods in 4 small watersheds along the leeward side of the Appalachian mountains in the mid-Atlantic states. Threat scores consistently above 0.6 and close to 0.8–0.9 were obtained for 18 hour lead-time forecasts, and skill scores up to 60% were obtained for the 24 hour lead-time forecasts.

3.5 Hazardous waste management

The objectives are

- Designing and applying cost-effective technologies for site restoration and remediation.
- Developing practices for monitoring, preventing, reducing, and cleaning up air emissions, wastewater discharges, and hazardous wastes and obtaining appropriate permits or exemptions.
- Conducting monitoring, risk assessment, and chemical analyses to meet regulatory requirements.
- Developing and implementing strategies to minimize waste and prevent pollution.
- Designing and implementing cost-effective ways to manage hazardous and nonhazardous waste streams.
- Implementing ways to mitigate potential impacts to natural and cultural resources.

3.5.1 Smart Pump and Treat

Conventional flow-and-transport models are generally quite slow to predict the effectiveness of some groundwater remediation pumping strategies. Thus, we have been using a smart pump-and-treat approach, that is, employing artificial neural networks (ANNs) trained to predict time, effectiveness, and cost data, then harnessed to search for strategies that balance timely and effective cleanup with minimum cost. In one training simulation, we analyzed 28 locations (areas thought to contain groundwater contamination and

required to be cleaned up within 50 years) to identify the lowest cost subset that was as effective as the full 28-location set. Analyses showed that treating 8 to 13 locations could meet our containment and removal goals and cost less than 35% of treating all 28 locations. In the current simulation, we are using a grid of 225 pumping sites to evaluate strategies involving 50 extraction and 10 injection pumps. The use of ANNs is crucial to find low-cost alternatives to reduce the time and maximize the extent of cleanup in five-year management periods.

3.6 Proximity Sensor for Surface Modelling

Sensor integration represents an active area of investigation in the field of robotics and industrial automation. Proximity sensors seem to be quite attractive for their acceptable cost-to-performance ratio, as compared to that of more expensive sensing techniques, e.g., vision or laser range finding. Neural network based joint interpretation of UltraSonic and InfraRed measurements provided by a composite proximity sensor, can be used to extract geometrical and morphological features of a flat target. Among proximity sensors, ultrasonic (US) and infrared (IR) detectors are particularly interesting in real-life applications, as one of their most interesting features is that IR reflecting behaviors have well known characteristics of complementarity, fusion of data provided by different proximity sensors is therefore a crucial point to partly overcome their limitations.

A composite sensor integrates into the same device an US range finder: the couple of emitting and receiving capsules and an IR detector: the light emitting diode and its coupled phototransistor. The spatial information provided by the US for interpreting the signal coming from the IR sensor can be used to estimate the spectral reflectivity, known as the “color”, of a flat target, in view of supporting the navigation of a semi autonomous vehicle, as a sort of “label recognizer”. An approach based on neural networks allows a faster design of the sensor fusion system as well as better performance, both in terms of lower measurement error, lower classification error probability and fewer measurements needed for making a decision.

3.6.1 Neural Network Application on Proximity Data

The problem to be addressed in order to interpret the data coming from the IR subsystem is to develop a model of the IR output signal v_{IR} to express its dependence on v_{US} and on the target color. A methodology for recognizing the target color from the sensing device outputs has to be developed. A neural approach is used for both modelling and classification problem. By means of an exhaustive test of the IR output, a nearly linear dependence of the IR measurement v_{IR} on $1/d^2$. A proportionality coefficient α ($0 < \alpha \leq 1$) is associated to each color, with value 1 representing *WHITE* target.

The integration of the data provided by the US and IR subsystem consists of using the US measurements concerning d and θ to make a prediction of $z_W(d, \theta) = m_d(d).m_\theta(\theta)$. In the case of a *WHITE* target: the ratio $\alpha = v_{IR}/z_W$ is then an estimate of α and is fed to the classifier. The efficiency of this approximation depends on different factors: the accuracy of the model, the internal noise of the IR sensor and the noise in the US measurements. A complex and oversized structure of the network increases the risk of overfitting and loss of generalization. In practice the network would be prone to training the noise more than the real signal. Thus a more practical solution is to make the classification process take into account the aforementioned uncertainties in a probabilistic sense, where possible inconsistency of the IR and US data were taken into account in an heuristic fashion.

Therefore, in proximity sensor data analysis, neural networks are applied as multilayer perceptrons having one hidden layer trained with the available experimental data. Network *Classifier* is a single layer network trained to classify the different colors. The classifier is a onehot decoder with as many outputs as classes and samples measured with different target colors are used for its training. After training, the network output delivers a continuous value in each component of the output vector, which can be interpreted as being proportional to the class probability and used for attributing the input pattern to one of the classes.

Figure: 4 illustrates the neural network application for Proximity Sensor Data Analysis.

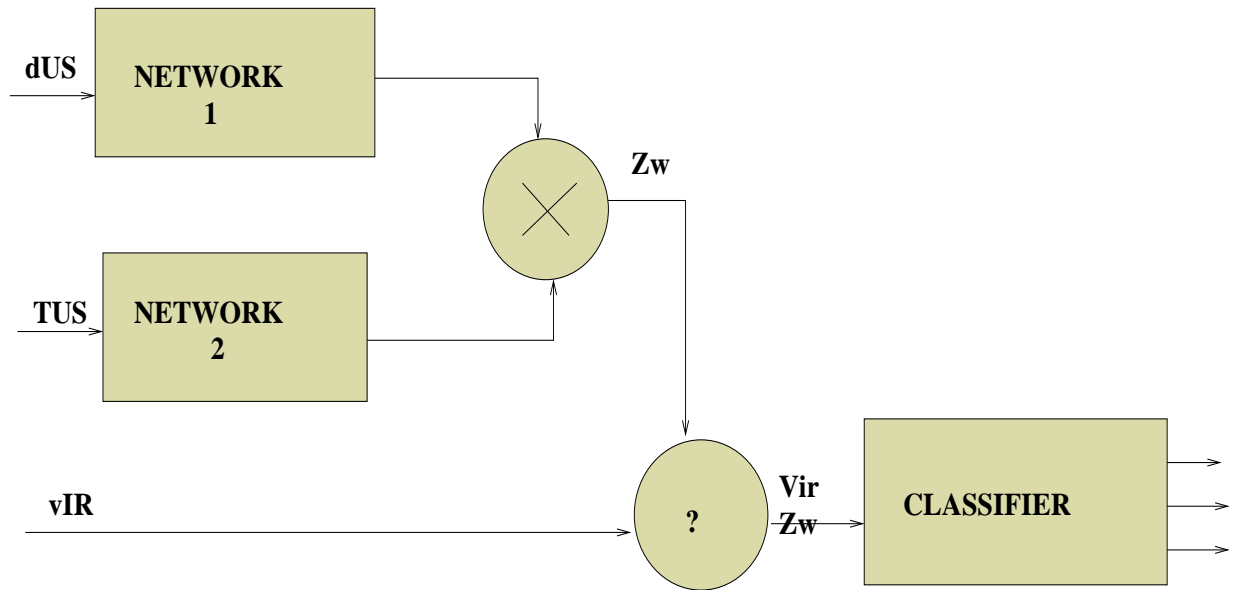


Figure 4: Neural Network Application on Proximity Data Analysis

4 Fuzzification of Neural Network for Sensor Fusion

One of the major problems associated with the application of neural network for sensor fusion is the inability of the NN's to cope with sensor failure. Training of the neural network is based on the assumption that sensors are operational and deliver reasonable data quality. However, in many real-life scenarios this assumption is not true. Preliminary experimental data shows that for a suite of nine sensors, the failure of only one sensor reduces the correct target identification confidence level from near 100% to slightly lower than 30%.

The imminent solution to this problem is to provide a set of neural network realizations, with each realization corresponding to a specific sensor (or

sensors) failure. This method is mostly applicable to the failure of a single sensor. Each sensor should be tested to assess sensor functionality by comparing current sensor reading against a sensor data "cookbook". Disconnecting a particular sensor from the sensor suite will be followed by the selection of the corresponding NN weight set. In this case, the most suitable NN will always be selected to obtain the best confidence level. Once the data is collected and pre-processed, each sensor is tested in-situ to determine the sensor's functionality.

The sensor functionality is determined by comparing the current sensor data with that sensor's data cookbook. The global test results of the sensor functionality are fed into the fuzzy adaptive weight generator, which selects one of the fuzzified sets of interconnected weights. A particular set of weights so selected is transferred to the decision making NN for the final recall. Preliminary results show 60% improvement in the confidence level comparing to standard fixed NN solutions.

5 A Proposed Surveillance Model: NeuroServ

In this section we propose a model for mobile surveillance in a wide-area military base using neural network based sensor data fusion. Automated feature extraction from sensorial data is of key importance in military surveillance. We categorize sensors are categorized according to the information they provide. Features can be classified into buildings, fields and water-bodies.

The features can be characterized using texture i.e. spatial frequency of intensity patterns as observed by the mobile sensing robots. The autonomous sensing-robots are spread across the surveillance site (for example International Borders) and monitor temperature, relative humidity, moving objects and also providing optical inspection capability. Location Servers (LC) and Mobile Switching Centers are setup to coordinate the locations of individual robots.

5.1 Location Management in NeuroServ

In a large-scale scenario, each robot (R_i : $i=1$ to n) acts as a Mobile Host (MH) in the mobile network. A key issue is of location management of the mobile robots since aggregation of many robots near fewer Location Servers or Mobile Switching Centers (MSCs) leads to loss of information in other regions.

So, the proposed location management strategy is to monitor the location of individual robots using regional directories (RDs) with and notifying the robot whenever it crosses from one cell to another so that the robot can return to original cell. The merits of this mechanism are two-fold: The robot can sense in a wider area thereby providing a higher resolution optical inspection and the approach becomes widely scalable for bigger networks and larger number of robots.

5.2 Feature Extraction in NeuroServ

We propose neural network technology for extracting features. Sensors record contrast, image-bitmap and send it to the expert-system using a multi-hop mobile network. The expert-system has data regarding sensor characteristics and scans the observed data for any abnormality. This implements decision-level sensor data fusion. A neural network trained on previous data is used to construct a classification-system over the observed data.

5.3 Merits of NeuroServ Model

The merits of the proposed NeuroServ model are as follows:

- Images with low resolution or poor contrast are correctly classified as buildings/plains/water-bodies because of other fusion of data from other sensors.
- High spatial scalability of the architecture over wide-scale military bases such as international borders because of use of distributed location managements and mobile robots.

- Cost effectiveness because of usage of heterogenous mobile robots. Heterogenous robots monitoring image or temperature data can be easily manufactured in bulk and provide complimentary data to the expert systems.
- Real-time data analysis because of abstraction of data-analysis from data-recording. Expert systems having high computing powers can be spatially very distant from the surveillance-site and communicate with the mobile robots over the mobile network.
- Supervised learning can improve the accuracy of model over time.

6 Comparison of Neural Network with Other Approaches

In 1998, the U.S. Defense Advanced Research Projects Agency (DARPA) initiated an evaluation of its intrusion detection research projects.¹ To date, it is the most comprehensive scientific study known for comparing the performance of different intrusion detection systems (IDSs). MIT's Lincoln Laboratory set up a private controlled network environment for generating and distributing sniffed network data and audit data recorded on host machines. Network traffic was synthesized to replicate normal traffic as well as attacks seen on example military installations. Because all the data was generated, the laboratory has a priori knowledge of which data is normal and which is attack data. The simulated network represented thousands of internal Unix hosts and hundreds of users. Network traffic was generated to represent the following types of services: http, smtp, POP3, FTP, IRC, telnet, X, SQL/telnet, DNS, finger, SNMP, and time. This corpus of data is the most comprehensive set known to be generated for the purpose of evaluating intrusion detection systems and represents a significant advancement in the scientific community for independently and scientifically evaluating the performance of any given intrusion detection system. TCP/IP data was collected using a network sniffer and host machine audit data was collected using Sun Microsystem's Solaris Basic Security Module (BSM). In addition, dumps of the file system from one of the Solaris hosts were provided. This data was distributed to participating project sites in two phases: training data and test data. The training data is data labeled as normal or attack

and is used by the participating sites to train their respective intrusion detection systems. Once trained, the test data is distributed to participating sites in unlabeled form. That is, the participating sites do not know a priori which data in the test data is normal or attack. The data is analyzed off-line by the participating sites to determine which sessions are normal and which constitute intrusions. The results were sent back to MIT's Lincoln Labs for evaluation. The attacks were divided into four categories: denial of service, probing/surveillance, remote to local, and user to root attacks. Denial of service attacks attempt to render a system or service unusable to legitimate users. Probing/surveillance attacks attempt to map out system vulnerabilities and usually serve as a launching point for future attacks. Remote to local attacks attempt to gain local account privilege from a remote and unauthorized account or system. User to root attacks attempt to elevate the privilege of a local user to root (or super user) privilege. There were a total of 114 attacks in 2 weeks of test data including 11 types of DoS attacks, 6 types of probing/surveillance attacks, 14 types of remote to local attacks, 7 types of user to root attacks, and multiple instances of all types of attacks. The attacks in the test data were also categorized as old versus new and clear versus stealthy. An attack is labeled as old if it appeared in the training data and new if it did not. When an attempt was made to veil an attack, it was labeled as stealthy, otherwise it was labeled as clear. The reason we present this evaluation study is because we believe it to represent the true state of the art in intrusion detection research. As such, it represents the foundation of more than 10 years of intrusion detection research upon which all future work in intrusion detection should improve. From this study, we can learn the strengths of current intrusion detection approaches, and more importantly, their weaknesses. Rather than identifying which systems performed well and which did not, we simply summarize the results of the overall best combination system. Lincoln Laboratory reported that if the best performing systems against all four categories of attacks were combined into a single system, then roughly between 60 to 70 percent of the attacks would have been detected with a false positive rate of lower than 0.01%, or lower than 10 false alarms a day. This result summarizes the combination of best systems against all of the attacks simulated in the data. It shows that even in the best case scenario over 30% undetected. However, the good news is that the false alarm rate is acceptably low enough that the techniques can scale well to large sites with lots of traffic. Further analysis showed that most of the systems reliably detected old attacks that occurred within the

training data with low false alarm rates. These results apply primarily to the network-based intrusion detection systems that processed the TCP/IP data. This result is encouraging, but not too surprising since most of the evaluated systems were network-based misuse detection systems.

7 Conclusion

Neural Networks are exhaustively applied in Sensor Data Fusion. Autonomous Mobile Robots develop a local environment representation using sensorial data. Misuse and anomaly intrusion detection can be performed by neural network, learning from user behaviors. Neural Network approaches are applied in multisensor surveillance to integrate data from spatially distant sources. Neural Networks are extensively used in weather forecasting, hazardous waste management and proximity sensing for surface modelling. Fuzzy approaches for neural networks shows significant improvements in confidence level of sensor data over fixed neural networks.

References

- [1] Jeongdal Kim Tomasz Jansson Andrew Kostrzewski, Di Hyn Kim and Gajendra Savant. Fuzzified neural network for similar/dissimilar sensor fusion.
- [2] Tim Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4):99–105, 2000.
- [3] Armin Scholl Christian Becker. A survey on problems and methods in generalized assembly line balancing.
- [4] F. Cremer, J. Schavemaker, E. den Breejen, and K. Schutte. Detection of anti-personnel land-mines using sensor-fusion techniques, 1999.
- [5] Frank Cremer, Eric den Breejen, and Klamer Schutte. Sensor data fusion for anti-personnel land-mine.
- [6] J.H. Jarvis D.H. Jarvis. Holonic diagnosis for an automotive assembly line.

- [7] J.G.M. Schavemaker E. den Breejen F. Cremer, K Schutte. A comparison of decision-level sensor-fusion methods for anti-personnel detection.
- [8] Seth Freeman and Joel Branch. Host-based intrusion detection using user signatures.
- [9] L Ronnie M Johansson and Ning Xiong. Perception management - an emerging concept for information fusion. 4(3):231–234, 2003.
- [10] Matthew Lees. Adaptive flood warning and river management.
- [11] llnl. Environmental restoration, protection, and waste management.
- [12] Bing Ma. *Parametric And Nonparametric Approaches For Multisensor Data Fusion*. PhD thesis, The University of Michigan, 2001.
- [13] Hironobu Fujiyoshi Takeo Kanade Robert T. Collins, Alan J. Lipton. Algorithms for cooperative multisensor surveillance.
- [14] Jake Ryan, Meng-Jang Lin, and Risto Miikkulainen. Intrusion detection with neural networks. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.
- [15] Robert Hurriion John Edwards Stewart Robinson, Thanos Alifantis. Modelling and improving human decision making with simulation.
- [16] Diego Zamboni. Doing intrusion detection using embedded sensors. Technical Report 2000-21, CERIAS, Purdue University, 2000.