

# Integration and Implementation of Secured IP Based Surveillance Networks

Charles C. Castello<sup>1</sup>, Jeffrey Fan<sup>1</sup>, Te-Shun Chou<sup>2</sup>, Hong-Ming Kuo<sup>3</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, Florida International University, Miami, Florida, USA

<sup>2</sup>Department of Technology Systems, East Carolina University, Greenville, North Carolina, USA

<sup>3</sup>Identification and Security Technology Center, Industrial Technology Research Institute, Hsinchu, Taiwan

**Abstract** — This paper presents a method of integration and implementation of transmitting video and audio data from multiple Internet Protocol (IP) surveillance cameras in a wireless sensor network to a centralized management unit (central node) using Real Time Streaming Protocol (RTSP). The wireless network is based on “Star” topology by using the IEEE 802.11 series of standards in Wireless Local Area Networks (WLAN). Security for wireless communications between the sensors and central node is achieved in real-time through the proposed uncertainty classification of network traffic with little and negligible run-time overhead for hacker or threat detection. A correlation-based feature selection algorithm is utilized in the central node for improved intrusion detection speed and accuracy by eliminating redundant and irrelevant features in the feature space. Uncertainty problems are solved using the fuzzy belief k-Nearest Neighbours (k-NN) intrusion detector, which incorporates the fuzzy clustering technique with the Dempster-Shafer theory. Also, the detection process is made more efficient using the k-NN technique. Experimental results show up to 94% detection rate of potential threats and intrusions, which indicate a significant performance increase compared with three other classifiers.

## I. INTRODUCTION

Recent years have shown that sensor networks have become extremely important in a number of facets in the modern world where applications include defense, manufacturing, security, weather forecasting, and many more [1]-[3]. Defense functions include the control and monitoring of *Unmanned Aerial Vehicles (UAV)* and *Unmanned Ground Vehicles (UGV)* from a single location for the purposes of surveillance and reconnaissance. Manufacturing functionality consists of monitoring stages of the manufacturing processes. Security includes video and audio capturing from multiple nodes, and weather forecasting entails recording of weather conditions from multiple locations to a single point. Other applications include the monitoring of power stations, oil pumping stations, etc... in remote locations where communication infrastructures are nonexistent.

There are many wireless communication technologies that fit the requirements needed for a sensor network. Attributes such as availability, complexity, cost, range, reliability, security and standards are all taken into consideration. However, with the utilization of any wireless network, added security measures are needed to combat potential vulnerability to threats including hacker activities that may be considered intrusions to the network. This is particularly important to military and security applications that transfer sensitive materials.

To accommodate added security needs, an *Intrusion Detection System (IDS)* is utilized to secure data transfers between nodes in a sensor network by examining network activity for inappropriate behaviour in real-time with little or no run-time overhead. The proposed *IDS* technique is meant to support other security measures by identifying potentially harmful activities. For example, end-to-end encryption techniques could be utilized to protect data being transferred while the proposed *IDS* technique could be used to identify potential hacker activities. In the event of a potential attack, personnel could be notified for further investigation to determine any possible damage to the sensor network, stolen data, attack origin, etc.

In order to design an *IDS*, large data sets of network traffic are collected for examination and training. These data sets include a great deal of traffic records with a number of various features including length of connection, type of protocol, network service, and so on. The collected data sets are used to train misuse detection techniques in order to identify potential security threats. The problem with this is that some included features in data sets are irrelevant with poor prediction ability to the class (e.g. normal or attack) and some features may be considered redundant due to high inter-correlation with other features [4]. Therefore, feature analysis is an important step in *IDS* in order to keep the more relevant features. This will not only decrease computational times, but also increase detection accuracies. Other problems in collected data sets are uncertainty because of limited amounts of information about intrusive activities and ambiguity in data sets. Thus, resolution of uncertainty problems is another important step in *IDS*.

The contribution of this paper is to design and develop a secure wireless sensor network by using *IEEE 802.11* wireless standard. The sensor network will stream audio and video data from multiple *IP* cameras to a *central node* using *RTSP* [5]. Since, it is Internet based, the network is vulnerable to potential hacker or other threat activities. In the proposed design framework, the *central node* will have the ability to access any of these sources through software developed in Microsoft Visual Studio. An *IDS* is employed in the *central node* to handle the security of network traffic in real-time after initial training. A novel uncertainty classification is designed and implemented for the *IDS* which combine a *correlation-based feature selection algorithm* [6] and *fuzzy belief k-NN intrusion detector* [7]. The correlation-based feature selection algorithm is utilized to improve both the intrusion detection speed and accuracy. Features with poor prediction ability to signatures of attack and features inter-correlated with one or more other features are considered unnecessary. Such features are removed and only indispensable information about the original feature space remains.

The *fuzzy belief k-NN intrusion detector* uses the *fuzzy clustering technique* [8][9] in combination with the *Dempster-Shafer theory* [10][11] to resolve uncertainty problems caused by limited and ambiguous information during the decision process of deciding whether or not activities are considered normal or attack. Also, the *k-NN* technique [12] is employed to speed up the detection process. Evaluation of the proposed *IDS* is approached using the intrusion detection benchmark data set *DARPA KDD99* [13] with both normal and attack classes. Results for the proposed *IDS* show favourable detection rates for all four attack types compared with the outcomes of *k-NN* [12], *Fuzzy k-NN* [14], and *Evidence-Theoretic k-NN* [15] classifiers.

This paper begins with a review on previous research in section II followed by the approach for uncertainty classification in section III. The implementation examples of the sensor network and *IDS* are discussed in section IV and lastly, the paper is concluded in section V.

## II. PREVIOUS RESEARCH

There has been much research in the field of *IDS* for sensor networks. Work done in [16] uses hidden *Markov models* in order to detect abnormal transitions in measurements from sensor networks. The main issue with this method is the focus on individual sensor nodes rather than the sensor network infrastructure. Research performed in [17][18] discuss the challenges for intrusion detection in ad-hoc networks. An anomaly detection method is proposed but a detailed solution and implementation is not furnished.

Other types of research focus on *IDS* schemes for ad-hoc networks. The investigation in [19] utilizes a clustering algorithm to build a normal traffic behavioural model and uses said model to detect abnormal traffic patterns. A key advantage to this approach is the ability to detect unknown types of attacks though testing was limited using known attacks. On the other hand, research in [20] uses prior knowledge of attacks with a signature-based detection of attack traffic. The primary focus of this work is how the number of nodes affects the accuracy of detecting attacks. The investigation in [21] proposes a specification-based *IDS* which utilizes the behaviour of ad-hoc routing protocols. However, detailed state information records are required and no evaluation is given for this approach.

Lastly, research performed in [22] applies an anomaly detection technique for intrusion detection in ad-hoc networks called *cross-feature analysis* which determines the correlation between features that appear in normal traffic. This technique has the ability to detect *black hole attacks* and *packet dropping attacks* in simulated ad-hoc networks. A potential drawback to this method is that the detection model can contain a large number of rules which are based on different combinations of feature values. When dealing with sensor networks, the size of the detection model may increase computational time requirements to test the model on incoming traffic due to memory requirements.

When comparing the proposed *IDS* and the previous works of research, there are many advantages, namely the increased performance due to the elimination of redundant and irrelevant features in the feature set using *correlation-based feature*

*selection algorithm*, in-depth evaluation using *DARPA KDD99* which includes 39 attack types falling into four categories, and where experimental results show up to a 94% detection rate when using *fuzzy belief k-NN intrusion detector*.

## III. UNCERTAINTY CLASSIFICATION

The wireless communications are established through an *IP* configuration in the proposed *RTSP* wireless sensor network implementation. Hence, communications are vulnerable to potential threats that may be considered an intrusion to the network. To combat this, a novel uncertainty classification is implemented to improve the security of the wireless network by identifying potential hacker activities in real-time with little or no run-time overhead.

In the proposed uncertainty classification, the *correlation-based feature selection algorithm* is combined with the *fuzzy belief k-NN intrusion detector*. The prior technique decreases computational time and increases accuracy by eliminating redundant and irrelevant features in the data set. The latter technique solves uncertainty problems using *fuzzy clustering technique* with *Dempster-Shafer theory*. The *k-NN* technique is used to make uncertainty problem solving more efficient. When applying these techniques to the wireless sensor network, a training phase is needed to determine the significant features and decision rules for the *feature selection algorithm* and *fuzzy belief k-NN intrusion detector* respectively where the accuracy of the *IDS* increases with increased amounts of training datasets. Once the training phase is complete, the proposed *IDS* can monitor and identify potentially harmful network activities in real-time based on the learned behaviours. Theory behind the training phase categorization process for the *correlation-based feature selection algorithm* and *fuzzy belief k-NN intrusion detector* will now be discussed in the proceeding subsections.

### A. Correlation-Based Feature Selection Algorithm

*Correlation-based feature selection algorithm* is used to eliminate features included in collected data sets of network traffic which have poor prediction abilities in terms of potential attacks and redundant information compared with other features. This is done utilizing the feature selection algorithm based on information-theoretical measure described in Fig. 1. The main tool used in this algorithm is *symmetric uncertainty* [23] which finds the strength of correlation between features and target classes using *entropy*, a measure of the amount of uncertainty. *Symmetric uncertainty* is calculated as

$$SU(Y; X) = 2 \left[ \frac{I(Y; X)}{H(X) + H(Y)} \right] \quad (1)$$

where

$$H(Y) = - \sum_i p(y_i) \log_2 p(y_i) \quad (2)$$

$$H(Y|X) = - \sum_j p(x_j) \sum_i p(y_i|x_j) \log_2 p(y_i|x_j) \quad (3)$$

$$I(Y; X) = H(Y) - H(Y|X) \quad (4)$$

The  $Y$  variable represents the *entropy* before and after observing values of another *entropy* variable  $X$  where  $p(y_i)$  is the prior probabilities for all values of random variable  $Y$  and  $p(y_i|x_j)$  is the conditional probability of  $y_i$  given  $x_j$ . By representing classes with variable  $Y$  and features with variable  $X$ , *entropy* is equal to 0 when there is no uncertainty that all members of a feature belong to the same class. On the other hand, when *entropy* is equal to 1, members of a feature are totally random to a class making the range of *entropy* between 0 and 1.  $I(Y; X)$  is the *information gain* or *mutual information* where the given variable is measured based on how well it separates instances into another variable. It is symmetrical where the information gained about  $Y$  after observing  $X$  is equal to the information gained about  $X$  after observing  $Y$ .

For the algorithm shown in Fig. 1, there are two parts that achieve the reduction of the feature space. The first part, shown in lines 1-5, removes irrelevant features with poor prediction ability of the target class. The second part, lines 6-12, eliminates redundant features that are inter-correlated with other features. The variable  $w$  is used to adjust the discriminative power of mutual information performed on feature-to-feature and feature-to-class to the same level where  $w$  is equal to the mean of summation of feature-to-class information divided by the mean of summation of feature-to-feature information. By multiplying  $w$  to each feature-to-class measure, both feature-to-class and feature-to-feature reach to the same important rank [6]. The remaining features are considered significant information pertaining to predicting potential attacks with better accuracy and less computational time.

---

```

1 // Remove irrelevant features
2 Input original data set  $D$  that includes
  features  $X$  and target class  $Y$ 
3 For each feature  $X_i$ 
  Calculate mutual information  $SU(Y; X_i)$ 
4 Sort  $SU(Y; X_i)$  in descending order
5 Put  $X_j$  whose  $SU(Y; X_j) > 0$  into relevant
  feature set  $R_{XY}$ 
6 // Remove redundant features
7 Input relevant feature set  $R_{XY}$ 
8 For each feature  $X_j$ 
  Calculate pairwise mutual information
   $SU(X_j; X_k) \forall j \neq k$ 
9  $S_{XX} = \Sigma(SU(X_j; X_k))$ 
10 Calculate means  $\mu_R$  and  $\mu_S$  of  $R_{XY}$  and  $S_{XX}$ ,
  respectively.  $w = \mu_S / \mu_R$ 
11  $R = w \cdot R_{XY} - S_{XX}$ 
12 Select  $X_j$  whose  $R > 0$  into final set  $F$ 

```

---

Fig. 1 Feature Selection Algorithm [6]

### B. Fuzzy Belief $k$ -NN Intrusion Detector

The *fuzzy belief  $k$ -NN Intrusion Detector* is utilized to classify user activities as either normal or attack activities. This is done by using the *fuzzy clustering technique* with *Dempster-Shafer theory* in order to solve uncertainty problems and the  *$k$ -NN technique* to reduce the computational time.

In order to detect intrusions, all traffic connections are assigned to either normal or attack classes, given a training set composed of  $N$  network traffic connections where each traffic connection has  $n$  distinct features with positive numerical values. The training set is represented as  $T$  with the training connection as  $x$ , and the set of features in each connection as  $F$ . The class set is represented as  $L$  which includes a number of possible classes,  $p$ .

To classify ambiguous training connections as either normal or attack activities, the *fuzzy  $c$ -Means clustering technique* is used to deal with the uncertainty. The results that occur after clustering are a set of cluster centers  $C$  and a membership partition matrix  $U$ .  $p$  decision rules are built from a vector or connection of  $U$  where the membership grades are treated intuitively to be degrees of confidence that a connection belongs to a particular class. Each connection consists of a number of feature values  $F$ , a class label  $l$ , and a confidence value  $\alpha$ .

$$R_U = \{r_U\} \text{ where } r_U : \langle F_i, l_j \rangle, \alpha_{ij} \quad (5)$$

where  $i$  and  $j$  represent the connection and class numbers respectively. The confidence value and correspondent membership grades are proportional where the connection belongs to a certain class.

A number of  $p$  rules can also be generated from the cluster centers  $C$ , where in each rule, the antecedent part includes  $n$  values of a cluster center and a corresponding class label.

$$R_C = \{r_C\} \text{ where } r_C : \langle c_j, l_j \rangle, \alpha = 1 \quad (6)$$

where the degree of confidence,  $\alpha$ , is equal to 1 meaning there is full confidence that the cluster center belongs to the partitioned class.

Now, let's assume  $v$  is an incoming connection which needs a classification. In order to classify the connection, the *Dempster-Shafer theory* is utilized to measure evidence obtained from the sets of  $p$  decision rules previously ascertained. Depending on the distance between  $v$  and the decision rule, the degree certainty on which class a connection belongs is affected. A large distance implies rules have little influence on  $v$  and with a small distance,  $v$  belongs to the same class of the rule.

Now, the  *$k$ -NN technique* is applied to find the most informative  $k$  nearest training connections of  $v$ . By using these  $k$  connections, corresponding decision rules are found. To differentiate the degrees of importance for each rule, the *weighted  $k$ -NN technique* [24] is used by assigning a weighted value,  $w$ . By adapting *Dempster-Shafer theory*, the degree of belief is quantified by a mass function, denoted as  $m$ ,

$$m(l_q) = w \cdot \alpha \quad (7)$$

where  $q$  represents the class number. Another factor besides the degree that  $v$  belongs to a certain class is the belief should also be designated to the frame with every class label. According to the *Dempster-Shafer theory*, the summation of all mass functions inferred from one training connection is equal to 1. Thus, the belief belonging to the frame is equal to one minus the summation of beliefs for all single classes.

$$m(L) = 1 - \sum_{i=1}^p m_i(l_q) \quad (8)$$

When there are multiple mass functions with the same class label,  $m_1(A)$  and  $m_2(B)$ , these functions can be combined using *Dempster Rule of Combination*, which fuses multiple mass functions into a single belief function  $m_{12}(C)$ . The combined result is known as the orthogonal sum of  $m_1$  and  $m_2$  and is noted as  $m = m_1 \oplus m_2$ .

$$m_{12}(C) = \begin{cases} 0 & \text{if } A = 0 \\ \frac{\sum_{A \cap B = C} m_1(A) \cdot m_2(B)}{\sum_{A \cap B \neq 0} m_1(A) \cdot m_2(B)} & \text{if } 0 \neq A \subseteq \Omega \end{cases} \quad (9)$$

After all mass functions with equal class labels have been combined into *fused mass functions*, final decisions can be made by introducing the *pignistic probability function*

$$Bp(l_q) = m(l_q) + \frac{m(L)}{p} \quad (10)$$

The function quantifies the belief into individual classes with pignistic probability distribution. In order to make the optimal decision,  $v$  is assigned to a class with the highest pignistic probability [7].

#### IV. IMPLEMENTATION EXAMPLES

In this section, the sensor network application in “Star” topology was developed using *Microsoft Visual Basic 2005 Express Edition* and *Visual Basic .NET* [25][26]. Simulation and testing for the uncertainty classification was done using *C++* and *TANAGRA* [27]. Other software used was *Apple Quicktime* for displaying *RTSP* feeds. All development was done in *Windows XP Professional*.

The *IP* camera applied in the experiment was the *TRENDnet* model *TV-IP212W*. This camera is able to send two-way audio and one-way video through wired or wireless networks using *LAN 10/100Mbps* and *802.11* devices. The camera also has the capability of advanced security features including *Wireless Encryption Protocol (WEP)*, *Wi-Fi Protected Access-Phase Shift Keying (WPA-PSK)* and *WPA2-PSK* encryption modes, *MPEG-4* and *MJPEG* video formats, and acts as a stand-alone server for direct access to audio and video. One of the more important features assisting in choosing this camera was the *RTSP*

capabilities. Very few *IP* cameras at the moment utilize *RTSP*. Some companies that offer a wide variety of *RTSP* compliant cameras are *Axis* and *TRENDnet*. Companies such as *ACTi*, *Panasonic*, *Sony*, and *Vivotek* are currently introducing *RTSP* compliant cameras.

In the following subsections, basic steps are discussed on how to configure an ad-hoc network utilizing *IP* cameras followed by steps taken to setup single and multi-sensor network applications. Finally, intrusion detection is implemented, tested, and detection performance is presented with comparisons showing the proposed *IDS* outperforming three other classifiers with a maximum detection rate of 94%.

##### A. Ad-Hoc Network Setup

In order to setup a wireless ad-hoc network, a wireless *802.11* device must first be installed on the development computer and *central node* of the sensor network. Setup of ad-hoc networks is trivial and similar on both systems.

##### B. IP Camera Setup

Configuration for the *IP* camera will now be discussed. Initial configuration for this camera must be done using the camera’s *LAN 10/100Mbps* Ethernet connection. In order to configure the *IP* camera correctly, the default *IP* address must first be found using the included software called “IPSetup.” Once the *IP* address has been found, the camera’s interface can be accessed using an internet browser where setting alterations take place.

TABLE I  
CAMERA IP ADDRESSES

Camera	IP Address
1	192.168.1.7
2	192.168.1.6
3	192.168.1.5
4	192.168.1.4

Similar setups are required for multiple *IP* cameras in the sensor network. The main difference would be the *IP* addresses applied to each camera. Table - I shows the *IP* addresses applied in the implemented sensor network.

##### C. Single Sensor Setup

A software application for a sensor network with a single camera, where many different software controls were used including *Quicktime*, will now be discussed. All methods and properties of controls can be adjusted through the *properties window* [25]. The sizing properties will affect the size of the *Quicktime* control during the application’s start and during the *RTSP* media feed. The “URL” property determines the *RTSP* data stream source.

There is needed coding for the “Stop”, “Pause”, “Play”, and “Unmute” buttons along with the “File” menu in the application. The “Stop”, “Pause”, and “Play” buttons are very straight forward where the “Stop()”, “Pause()”, and “Play()” methods are utilized from the *QuickTime* control. The “Unmute” button is coded to change viewable text on the button to “Mute” when

the user clicks the button. This will also unmute sound on the *QuickTime* control. When clicked again, the text on the button is changed back to “Unmute” and the audio is muted. This is done using an If-Else statement [26] along with the “Text()” property of the button and the “AudioMute()” property of the *QuickTime* control. Lastly, the “File” top menu consists of an “Exit” command where the “Exit()” method of the application is utilized in the code.

In order to test the application and data streaming between the *central node* and camera, the ad-hoc network must be running properly. During application usage, press the “Play” button and media streaming should begin in a few seconds after the initialization process has completed.

#### D. Multi-Sensor Setup

A sensor network application with multiple sensors will now be discussed, which is very similar to the single sensor variation. The “(Name)” property for each control should respectively match in the code and names given to controls must be unique. Also, the “URL” property for each *QuickTime* control must be changed according to the specifications. The finished product to access multiple sensors from a *central node* is shown in Fig. 2.



Fig. 2 Multiple Sensor Application

#### E. Intrusion Detection

In the course of experimentation, the approach was trained and evaluated using the intrusion detection benchmark data set *DARPA KDD99*, which not only includes a large quantity of network traffic but also a wide variety of attacks. Each connection is represented with 41 features plus a label of either normal or a type of attack. A total of 39 attack types are included and fall into four main classes: *Denial of Service (DoS)*, *Probe*, *User to Root (U2R)*, and *Remote to Local (R2L)*.

In the first stage of experimentation, four sets of data are generated according to the normal class and four categories of attack. In each data set, connections with the same attack category and all normal connections are included. The feature

selection algorithm is run and 12, 12, 5, and 7 from the 41 original features are extracted for use of detecting *DoS*, *Probe*, *U2R*, and *R2L* attacks, respectively.

Next, experiments of intrusion detection are performed using the selected features from the previous stage. To minimize the inaccuracy and variation factor of experimental results, 10 trials are performed in every detection task. In each trial, only a very small amount of connections are randomly selected from training and testing sets. It not only speeds up the classification process but also simulates the uncertainty caused by lack of network traffic information. The four training sets have 545 *DoS* attacks, 213 *Probe* attacks, 52 *U2R* attacks, and 99 *R2L* attacks, respectively and each set has 878 normal connections. The four testing sets have 235 *DoS* attacks, 268 *Probe* attacks, 215 *U2R* attacks, and 291 *R2L* attacks, respectively and each set has 479 normal connections.

Procedures of training and testing are performed in each trial. In the training phase, a *fuzzy belief k-NN detector* is constructed using the limited and ambiguous training data. Testing data are then fed into the trained detector to identify intrusions in the testing phase. The performance is evaluated using distinct numbers of  $k$  nearest neighbour, *False Positive Rate (FPR)* and *Detection Rate (DR)*.

Table - II shows the averaged rates with  $k$  ranging from 1 to 10 for the proposed *IDS* as well as the *k-NN*, *Fuzzy k-NN*, and *Evidence-Theoretic k-NN* classifiers. Since *DoS* and *Probe* attacks usually have frequent sequential patterns that are different from normal connections, both can be easily separated from normal activities thus achieving low *FPR* and high *DR* for the proposed approach. On the contrary, *U2R* and *R2L* attacks do not have any intrusion, only frequent sequential patterns. Both are embedded in the data portions of the packets and normally involve only a single connection. Also, many attacks in *Normal-U2R* and *Normal-R2L* testing sets do not exist in the corresponding training sets.

Past research results [28][29][30] have indicated that it is difficult to achieve satisfactory detection accuracy while detecting *U2R* and *R2L* attacks, which *DR* are up to 30%. However, *DR* of 81.22% and 68.10% are achieved in detecting *U2R* and *R2L* attacks, respectively. Also, comparing the results with three other classifiers shows a large improvement in *DR* for all four attack types. This shows that the proposed approach is capable of handling network traffic data which contain degrees of uncertain information.

#### V. CONCLUSION

An IP-based wireless sensor network has been analysed and developed using *RTSP* to stream incoming data from multiple sensors to a central location in “Star” topology. Security was handled to prevent Internet intrusion for the sensor network using *shared network authentication scheme*, *WEP*, and the proposed uncertainty classification which combines *correlation-based feature selection algorithm* with *fuzzy belief k-NN intrusion detector* in real-time with little and negligible overhead after the initial training sequence. The application was developed and tested using *Microsoft Visual Basic 2005 Express Edition* in conjunction with *Visual Basic .NET*, *Apple Quicktime*,

TABLE II  
DETECTION PERFORMANCE AND COMPARISON

	<i>Normal-DoS</i>		<i>Normal-Probe</i>		<i>Normal-U2R</i>		<i>Normal-R2L</i>	
	FPR	DR	FPR	DR	FPR	DR	FPR	DR
<i>k-NN</i>	0.64%	91.74%	0.67%	76.66%	0.18%	13.11%	0.36%	18.49%
<i>Fuzzy k-NN</i>	0.74%	93.12%	0.82%	78.44%	0.27%	15.25%	0.35%	20.36%
<i>Evidence-Theoretic K-NN</i>	0.73%	92.98%	1.27%	83.31%	0.30%	16.87%	0.40%	21.01%
<i>Proposed Intrusion Detection</i>	9.73%	94.57%	0.25%	65.71%	9.75%	81.22%	9.63%	68.10%

TANAGRA, and C++. Essential knowledge, steps, and considerations were discussed to give the tools necessary to develop a secure sensor network, which can be tailored to fit the needs of the readers. Also, there are many other features and capabilities that are implied that can be added to the application, such as video and audio recording, motion estimation, two-way audio, and much more. Future research considering this work could take other aspects into consideration, such as improved efficiency using processor, battery, and bandwidth resources in wireless sensor networks that utilize the proposed IDS.

No matter what information is transferred in the wireless network, the uncertainty classification will protect data communicated between different nodes by identifying potentially harmful activity, where experimental results show up to a 94% detection rate of potential threats. The power to bring secure information from multiple sensors to a centralized location is immense which makes this a very important area of study.

#### REFERENCES

- [1] S. Oh, I. Hwang, and S. Sastry, "Distributed multitarget tracking and identity management," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 1, pp. 12–29, Jan.-Feb. 2008.
- [2] Y.B. Tao, H. Ding, and Y. Xiong, "A novel peer-to-peer distributed sensor network framework based on IP sensor for telemonitoring," *Assembly Automation*, vol. 26, no. 2, 2006.
- [3] F.J. Pierce and T.V. Elliott, "Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington," *Computers and Electronics in Agriculture*, vol. 61, no. 1, pp. 32–43, Apr. 2008.
- [4] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [5] H. Schulzrinne, A. Rao, and R. Lanphier. (1998) *The Internet Engineering Task Force (IETF)*. [Online]. Available: <http://www.ietf.org/rfc/rfc2326.txt>
- [6] T. S. Chou, K. K. Yen, J. Luo, N. Pissinou, and K. Makki, "Correlation-based feature selection for intrusion detection design," *IEEE Military Communications Conference*, Orlando, FL, Oct. 2007.
- [7] T. S. Chou and K. K. Yen, "Fuzzy belief k-nearest neighbors anomaly detection of user to root and remote to local attacks," *8th Annual IEEE SMC Information Assurance Workshop*, pp. 207–213, West Point, NY, Jun. 2007.
- [8] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [9] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, pp. 32–57, 1973.
- [10] A.P. Dempster, "A generalization of Bayesian inference," *Journal of the Royal Statistical Society, Series B*, vol. 30, pp. 205–247, 1968.
- [11] G. Shafer, *A Mathematical Theory of Evidence*, Princeton, University Press, Princeton, NJ, 1976.
- [12] E. Fix and J. L. Hodges, "Discriminatory analysis: nonparametric discrimination: consistency properties," *Report Number 4*, Project Number 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
- [13] KDD99 Archive: *The Fifth International Conference on Knowledge Discovery and Data Mining*.
- [14] M. Keller, M.R. Gray, and J.A. Givens Jr., "A fuzzy k-nearest neighbor algorithms," *Transactions on Systems, Man and Cybernetics*, vol. 15, no. 4, pp. 580–585, 1985.
- [15] T. Denoeux, "A k-nearest neighbor classification rule based on Dempster-Shafer theory," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 804–813, May 1995.
- [16] S. Doumit and D. Agrawal, "Self-organized criticality & stochastic learning based intrusion detection system for wireless sensor networks," *Proceedings of the IEEE Military Communications Conference (MILCOM 2003)*, vol. 22, no. 1, pp. 609–614, Oct. 2003.
- [17] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT 2003)*, pp. 368–373, Jan. 2003.
- [18] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad-hoc networks," *IEEE Wireless Communications*, pp. 48–60, Feb. 2004.
- [19] C.E. Loo, M.Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for routing attacks in sensor networks," *International Journal of Distributed Networks*, vol. 2, no. 4, pp. 313–332, Dec. 2006.
- [20] F. Anjum, D. Subhadrabandhu, and S. Sarkar, "Intrusion detection for wireless ad-hoc networks," *Proceedings of the IEEE Vehicular Technology Conference, Wireless Security Symposium*, pp. 2152–2156, Oct. 2003.
- [21] C. Tseng, P. Balasubramanyam, C. Ko, R. Limprasitiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV," *Proceedings of the 1st ACM workshop on security of ad hoc and sensor networks*, pp. 125–134, 2003.
- [22] Y. Huang, W. Fan, W. Lee, and P. Yu, "Cross-feature analysis for detecting ad-hoc routing anomalies," *Proceedings of the 23rd International Conference on Distributed Computing Systems*, 2002.
- [23] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.
- [24] S.A. Dudani, "The distance-weighted K-NN rule," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 6, no. 4, pp. 325–327, 1976.
- [25] B. Sempf, D. Xie, J. Greenwood, R. Harrop, C. Kwong, J. Machacek, B. Bischof, J. Reid, and K. Cheda, *Pro Visual Studio.NET*, D. Shakeshaft, Ed. Heidelberg, Germany: Springer-Verlag, 2004.
- [26] M. Halvorson, *Microsoft Visual Basic 2005 Step by Step*, B. Ryan and M.V. Tschudi-Sutton, Ed. Redmond, Washington, United States: Microsoft Press, 2006.
- [27] TANAGRA: <http://eric.univ-lyon2.fr/~ricco/tanagra/>
- [28] I. Levin, "KDD-99 classifier learning contest LLSoft's results overview," *ACM SIGKDD Explorations Newsletter*, Vol. 1, no. 2, pp. 67–75, Jan. 2000.
- [29] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "On the capability of an SOM based intrusion detection system," *Proceedings of the International Joint Conference on Neural Networks*, Vol. 3, pp. 1808–1813, Jul. 2003.
- [30] M. Sabhnani and G. Serpen, "Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context," *Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications*, Las Vegas, NV, pp. 209–215, Jun. 2003.