BioAIM: Bio-inspired Autonomous Infrastructure Monitoring

Bo Ryu[1] Nadeesha Ranasinghe[1] Wei-Min Shen[1] Kurt Turck[2] Michael Muccio[2]

***Abstract*— The Bio-inspired Autonomous Infrastructure Monitoring (BioAIM) system detects anomalous behavior during the deployment and maintenance of a wireless communication network formed autonomously by unmanned airborne nodes. A node may experience anomalous or unexpected behavior in the presence of hardware/software faults/failures, or external influence (e.g. natural weather phenomena, enemy threats). This system autonomously detects, reasons with (e.g. differentiates an anomaly from natural interference), and alerts a human operator of anomalies at runtime via a communication network formed by the Bio-inspired Artificial Intelligence Reconfiguration (BioAIR) system.**

In particular, BioAIM learns and builds a prediction model which describes how data from relevant sensors should change when a behavior executes under normal circumstances. Surprises occur when there are discrepancies between what is predicted and what is observed. BioAIM identifies a dynamic set of states from the prediction model and learns a structured model similar to a Markov Chain in order to quantify the magnitude of a surprise or divergence from the norm using a special similarity metric. While in operation BioAIM monitors the sensor data by testing the applicable models for each valid behavior at regular time intervals, and informs the operator when a similarity metric deviates from the acceptable threshold.

***Index Terms*— Adaptive systems, Cognition, Command and control systems, Communication networks, Intelligent control, Fault detection, Cyber Security, Fault tolerant systems, Unmanned Aerial Vehicles.**

I. Introduction

In contrast to traditional UAV operation where a single operator remotely controls an individual UAV, the BioAIR [1] system controls an entire swarm of UAVs in a distributed manner as summarized in section IV. The advantage of this distributed system is that it is robust to faults and failures, i.e. there is no single point of failure that the damage repair mechanisms cannot recover from. However, the lack of centralization makes it difficult to identify if the overall system has encountered any unforeseen or anomalous circumstances. This requires learning the expected or normal behavior of the BioAIR autonomous system and its underlying autopilot control systems, and identifying any deviations from the norm or anomalies. In addition to identifying anomalies, providing recommendations to correct the anomaly would enable a single operator to monitor multiple BioAIR swarms simultaneously. For example, in Figure 1 on the left side depicts a normal BioAIR network formed by a chain of nodes flying in between the origin and destination, while the right side depicts an anomaly caused by a malfunctioning node highlighted in red. Another words, the red node is not supposed to be there.

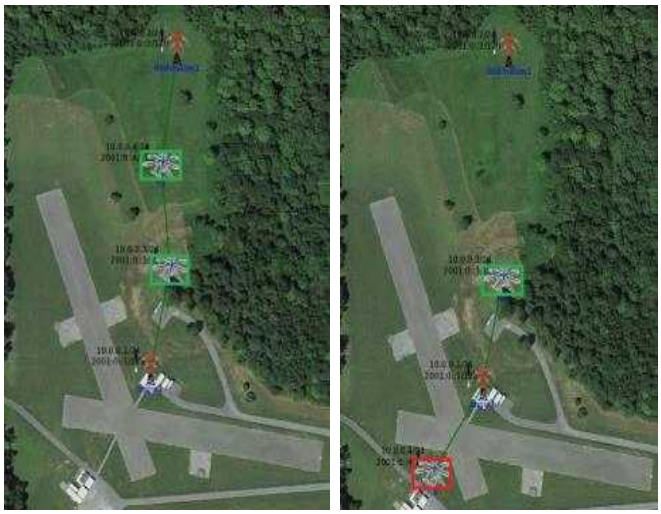

Figure 1: Normal BioAIR network vs. anomalous node in network

There are three central challenges to design such a decision support system. The first challenge is determining what constitutes as an anomalous behavior; a significant deviation from the normal behavior of the BioAIR system under monitoring. Due to the intrinsic nature of autonomy it is often non-trivial to distinguish undesirable/abnormal system changes from the transient behaviors of the underlying autonomous principles. The second challenge is developing an algorithm to learn the norm, and detect the anomaly (either defined a priori or learned). The third challenge is to identify the potential causes of the anomaly and to provide timely recommendations to support decision making.

The first challenge of defining normalcy in an autonomous system calls for a comparison mechanism between the current behavior of a system and its expected behavior. Given specifications of system behavior, such as models of normal or anomalous BioAIR behavior the comparison mechanism should be able to return a metric identifying the degree of deviation. This would consequently facilitate an ideal solution which would use the similarity metric to provide an alert to the human operator when a significant deviation is detected, while allowing the proposed autonomous system to deal with minor deviations.

The second challenge arises from the fact that it is not feasible to expect intricate details of the BioAIR system, the environment and the underlying autopilot systems to be completely visible to the decision support system. As such, it is inherently difficult to precisely characterize the expected or normal behavior in the face of dynamic missions and operational contexts. Therefore, the decision support system

should be able to acquire a basic model of the expected behavior, and improve it or learn, while training under controlled conditions and in operational environments.

The third challenge is to provide a formal mechanism for building a higher level language that makes anomaly detection technology relevant and useful to a human operator who must manage remote autonomous entities on an intermittent, asynchronous basis in emerging autonomous systems. This means that the decision support system must facilitate efficient information transaction mechanisms between the operator and the underlying autonomous systems. Hence, the creation of a user interface layer and associated dialog processes is essential. The proposed system must report which underlying behaviors are operating as normal, and which are suspected of being anomalous. An additional feature could include displaying a set of feasible actions to be taken or recommendations to be considered in the current situation.

The BioAIM solution presented in this paper was designed to address all of these challenges and was successfully tested in several simulated BioAIR swarm UAV deployments. The rest of this document is organized as follows. Section II summarizes the related work. Section III details the experimental setup. Section IV summarizes the BioAIR algorithm. Section V details the BioAIM algorithm and other components. Some experimental results are presented in Section VI. Finally, Section VII highlights potential future work and concludes the paper.

II. Related Work

BioAIM is an extension of the Surprise-Based Learning (SBL) algorithm introduced by Ranasinghe and Shen in 2008. The main innovation of this extension is the use of multiple agents, which make local decisions within a node (e.g. on each UAV), and global decisions at a centralized location (e.g. operator's server) based on communicated information (when it is available). The basic concept of SBL was first proposed by Shen and Simon in 1989, and later formalized as Complementary Discrimination Learning (CDL) [2]. SBL has improved CDL's model learning process to be more precise, and supports the resource limitations of embedded agents.

In contrast to most Reinforcement Learning techniques [3], SBL is able to transfer its learned knowledge between assets. Unlike most Neural Network and Genetic Algorithm solutions, SBL is able to learn offline and continue learning throughout the lifetime of an asset. At present, Deep Learning [4] and Google's Neural Turing Machines [5] in particular have gained a lot of interest in unsupervised learning applications, yet these black-box techniques are unable to expose their reasoning to aid remote management.

III. Experimental Setup

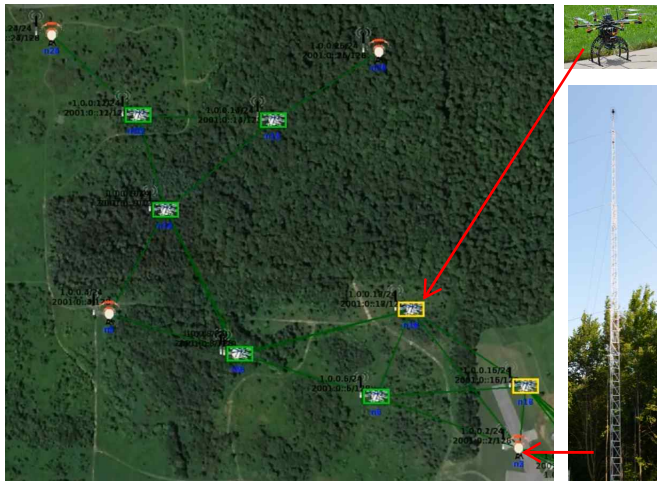

Figure 2: CORE WiFi Network Emulator

In these experiments BioAIR mobile nodes where modelled based on the performance of real unmanned rotary-wing hexarotor vehicles provided by AFRL. Each ground-based destination or origin node was represented by an ODROID minicomputer equipped with a WiFi card. All tests were performed in simulation with hardware-in-the-loop using the Navy's Common Open Research Emulator (CORE) [6] network emulator as shown in Figure 2.

IV. BioAIR Summary

The base case for forming a communications network is establishing the connectivity between two sites. An origin and a destination are networked together by forming a chain or a tentacle which is comprised of several nodes. Nodes can be launched from anywhere at varying time intervals with some a priori knowledge about the location of an origin and a destination. Once connectivity is established between a set of nodes, the received signal quality is used to improve the overall performance of the network rather than the straight-line distance between them. It is important to note that typically the signal quality has exponential falloff with distance, which can change due to environmental conditions.

The idea behind this algorithm is to form tentacles from one or more origins to one or more destinations by growing them in a biological way through the accumulation of nodes as shown in Figure 3. In basic terms, initially all nodes will fly towards an origin. When a node encounters a tentacle or an origin it will fly towards the destination, and at the edge of its communication range to the existing tentacle or origin, it will hold position if it is rotary-wing or a circular pattern if it is fixed-wing, thus extending the tentacle towards the destination. When the tentacle reaches the destination, any extra nodes will construct new tentacles or reinforce existing nodes.

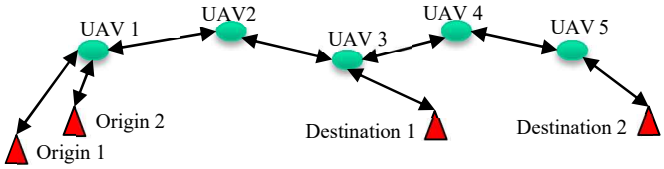

Figure 3: A tentacle connecting origins and destinations

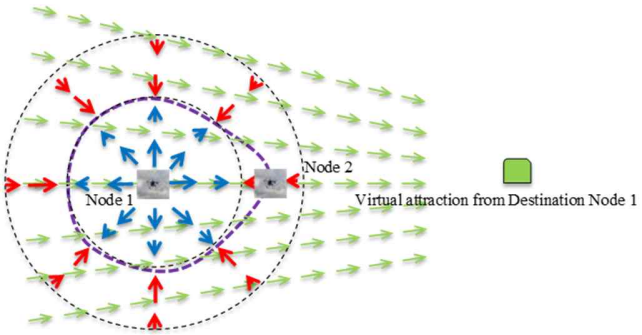

Figure 4: Fields guiding node 2 towards destination

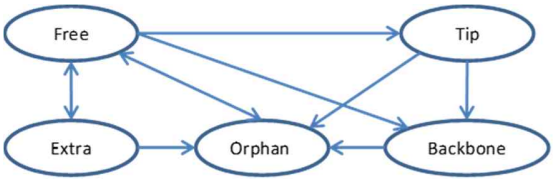

Figure 5: Valid role transitions for a node

Figure 6: Valid state transitions for a tentacle

The BioAIR algorithm stops a node at the edge of its communications range to a neighbor by converting the received signal quality into a field strength, which corresponds to being attracted or repelled by the nearest neighbor. This field indicates that when the nodes are close to each other they repel to prevent collisions, when they are outside the acceptable range they attract, and when they are completely out of range there is no signal or field. At the onset, the destination's location is disclosed to each node so as to compute a virtual attraction field towards it even when it is out of range.

Figure 4 visualizes the fields interacting between two adjacent nodes and the destination's virtual attraction with a fixed magnitude of 1.0. The result of summing these fields is an elliptical trajectory marked in purple. This guides nodes to an equilibrium point that exists closer to the destination. Given noisy readings and the fact that a node may switch to flying in a circular pattern when it enters the equilibrium point, there is no possibility for a node to stop at the equilibrium point that exists further away from the destination.

In order to accomplish the construction and subsequent maintenance of tentacles, each node will take one of the following roles: "orphan", "free", "tip", "backbone" or "extra". The BioAIR algorithm strictly dictates the valid role transitions as shown in Figure 5. Similarly, a tentacle could be in one of the following states reflecting its progress towards connecting an origin to a destination: "forming", "complete", "next destination", "damaged" or "reinforcing". Figure 6 depicts valid tentacle state transitions.

None of the above information is provided to BioAIM directly; instead it is expected to learn the overall behavior by observing sensor data during normal BioAIR operation.

V. BioAIM System

Much of BioAIM is built on SBL, with key differences in application specific assumptions and implementations. In particular, BioAIM is specifically designed for autonomous monitoring of the BioAIR system. BioAIM learns the relationship between actions/behaviors and sensors autonomously. The learned knowledge is stored in a prediction model comprised of several prediction rules. A prediction rule describes how data from relevant sensors should change when an action or behavior is executed. Each prediction rule contains a set of conditions, an action/behavior and a set of predictions, defined as follows:

Prediction Rule ≡

Conditions → Action/Behavior → Predictions (1)

Condition ≡

(Entity, Attribute, Comparison Operator, Value) (2)

Prediction ≡

(Entity, Attribute, Change Indicator, Value) (3)

It states that when the action is occurring, the conditions (2) are logic sentences describing the state of the observed sensor data at the current time, while the predictions (3) are logic sentences that describe the expected change in the sensor data at the next sampled time. For example, when a node starts flying towards the destination, its distance to the destination should decrease. This action/behavior can be represented as a prediction rule (Global Positioning System, Distance To Destination, !=, 0) → Fly → (Global Positioning System, Distance To Destination, ↓, X), where "↓"is a change indicator for decrease, and X is a value. This human readable format enables us to understand and debug each node's learned knowledge.

BioAIM monitors activities of individual nodes as well as the networked system as a whole. In particular, each node executes an instance of a local BioAIM agent, while the origin executes an instance of the global BioAIM agent. The local agent is responsible for monitoring the node's BioAIR data and any additional locally available sensor data to ensure that the node is performing as expected. In contrast, the global agent periodically gathers data from all BioAIR nodes in communication range via tentacles to ensure that the overall behavior of the swarm is performing as expected. The origin is the ideal node to host the global agent.

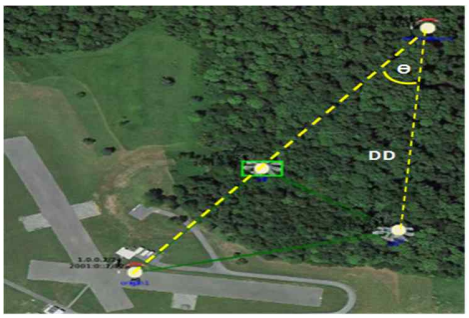

Figure 7: Desired tentacle formation behavior parameters

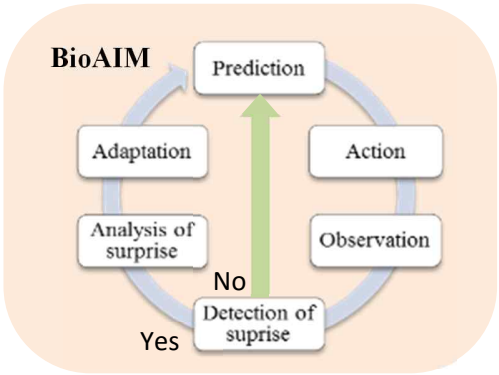

Figure 8: BioAIM process borrowed from Surprise-Based Learning

BioAIM learns the expected behavior during a priori training from successful BioAIR examples. The BioAIR data communicated to BioAIM includes information such as time, node's unique ID, node's role and tentacle's state. Additionally, as depicted in Figure 7 BioAIM computes the distance from a node to the destination (DD), the neighbor's role (NR; highest if any), and an angle Θ formed between a node and a straight line from the origin to the destination. The above information is sufficient to validate the complex BioAIR transitions, while enforcing a human intuitive notion of roughly where the nodes should be placed, which will be detailed in an example at the end of this section.

BioAIM makes observations at fixed intervals by recording all the sensor data on each node and transmitting it to a centralized node running BioAIM. This interval can be as short as short processing and communication times permit (i.e. every few seconds), or as long as it takes for significant changes to occur onboard a node (i.e. every few hours). When learning a prediction model from scratch there is no predicted outcome for the action being executed, else if a model has been learned there is a prediction describing expected observation at the next time interval. When the action is executing, a new observation is made at the next time interval. If there was no prediction, BioAIM compares the past and present observations with a set of predefined comparison operators (e.g. presence, absence, greater-than, less-than & equal-to) and updates its prediction model by creating a set of prediction rules that capture any changes. Given a prediction, BioAIM uses these comparison operators to check for any discrepancies or surprises between the expected and current observations. If there are no surprises the cycle repeats, else the prediction model is updated through surprise analysis and rule maintenance to improve future predictions as shown in Figure 8.

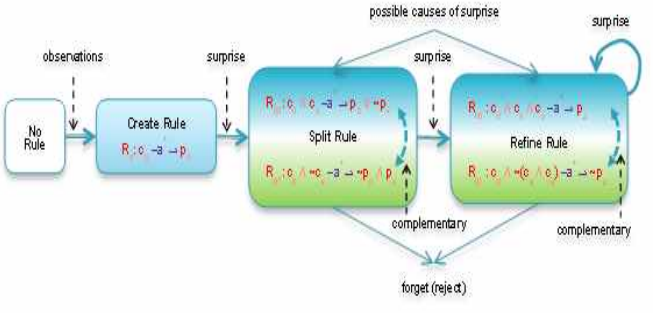

Figure 9: Life cycle of a prediction rule

Surprise analysis attempts to identify the cause of each surprise by comparing two similar observations in the past as described in [7]. BioAIM assumes there will be sufficient data to identify a discernable correlation for a surprise, yet in the event of encountering a hidden cause BioAIM records the entire observation to improve its prediction accuracy. Rule maintenance uses these potential causes to update the model accordingly.

Rule maintenance is comprised of three components responsible for creating new rules, splitting an existing rule once, and refining them thereafter as shown in Figure 9. In BioAIM, when a rule is surprised for the first time, it is split into two complementary "sibling" rules by appending a possible cause from surprise analysis. One sibling rule is specialized and the other sibling rule is generalized, and together they capture both the original and surprised results. However, if the surprised rule already has a sibling, then both sibling rules are refined together based on the identified causes of the surprise so that the revised rules may predict more successfully in the future. Of course, not all revised rules are improving, and it is possible that the revised sibling rules would describe a contradiction or encounter a large number of surprises consecutively in the execution. Such sibling rules are marked as obsolete and become inactive in predicting and further learning. Precise details of this complex process stemming from SBL can be found in [8].

Mathematically, a prediction rule can be represented as a conditional probability P(H|C,A), where H is the prediction, C the condition and A the action. This probability represents the transition probability between states in a structured model such as a Hidden Markov Model (HMM) or Automata. In a learned structured model, each state may contain a set of related prediction rules which often fire together in a sequence of training data. Hence, BioAIM forms a structural model by learning prediction rules and grouping the ones that fired together into states. Therefore, BioAIM is a "structural model learning" technique which can construct a predictive and structured model from the continuous actions and sensors. This is a novel feature rarely seen among the related state-of-the-art learning algorithms. The learned structured model helps distinguish between discrepancies or surprises caused by noisy data and completely abnormal events based on probabilistic matching, as was demonstrated by SBL in the DARPA Mind's Eye project's autonomous video surveillance application [7].

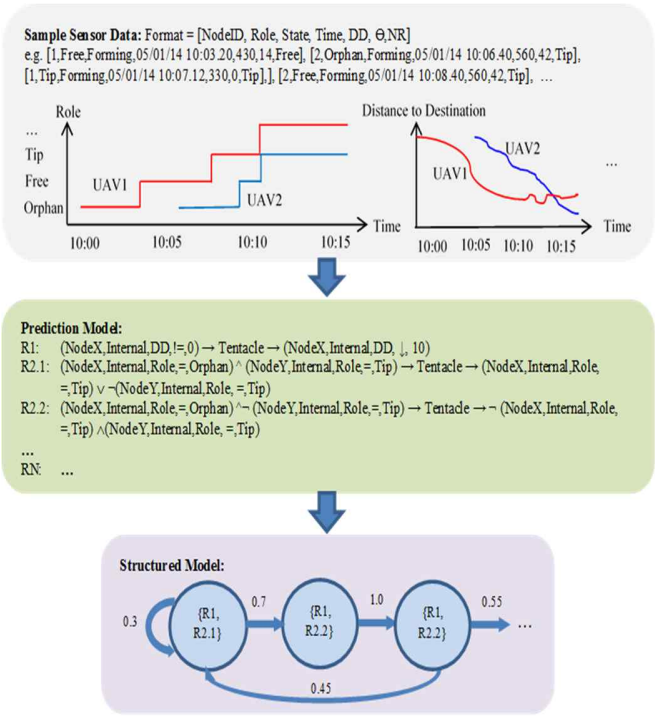

Figure 10: Learning a prediction model and structured model from sensor data

By using the predictions of one rule to satisfy the conditions of another rule, it is possible to string a sequence of prediction rules and forecast the predicted observation at a future point in time. Surprises occur when there are discrepancies between the prediction and the observation, yet some surprises are more important to detect and adapt to than others. For example, sensor noise may cause surprises, but they are less important than complete sensor failures. To quantify the magnitude of a surprise or divergence from the norm, BioAIM learns structured models when the prediction model for a given action or behavior yields no further surprises.

Figure 10 depicts an example of learning a structured model from a learned prediction model. This structured model returns the validity of the observed state or state sequence with respect to the predicted state or state sequence, as a number between 0.0 and 1.0 using a special similarity metric. This similarity metric is designed to be more accurate than forward-backward log-likelihood estimation in the presence of noisy or missing data at comparing a sequence of observations against the learned sequence of state transitions.

During the training phase BioAIM learns a prediction model for each valid behavior on each node from synthetically generated data created by hand based abstractions of the real data or very clean sensor data. The requirement for this abstraction is to eliminate any noise in real sensor data, which ensures that BioAIM learns an accurate prediction model devoid of errors introduced by noise. Note that extracting a good prediction model from noisy data would require a large number of varied training examples, resulting in much longer training durations. Once a prediction model is learned it is applied to sequences of real sensor data from one or more successful demonstrations of the behavior so as to learn the corresponding structured model. Training with real data ensures that the structured model captures the statistical probabilities of being in a particular state and transitioning to another. When in operation BioAIM monitors the sensor data by testing the applicable models at regular time intervals, and informs the operator when the similarity metrics deviates from acceptable thresholds for valid behaviors. These thresholds are recorded by training on several successful and unsuccessful variants of possible BioAIR deployments.

At runtime sensor data messages are periodically received at the origin from all BioAIR networked nodes. These messages at minimum contain the transmitting node's timestamp, unique identifier, role, tentacle state, calculated distance to destination, neighbor's role, and angle Θ formed between a node and a straight line from the origin to the destination. The graphs in Figure 10 help visualize some sample data to depict the complex relationships between certain elements. For example the graph on the left depicts how the roles of 2 nodes changes with respect to each other over a period of time, while the graph on the right depicts how the distance to destination of each node changes over time. This makes it clear that a relationship between the role and distance could also be plotted, yet none of these relationships are presented to BioAIM. Instead BioAIM is provided with the data and is expected to learn the relationships autonomously.

BioAIM samples the data periodically and learns a prediction model similar to the one depicted in Figure 10 following the processes visualized in Figure 8 and Figure 9. The prediction model effectively identifies and captures the relationships described earlier in the form of prediction rules. Notice rule R1 captures some of the distance relationship, while rules R2.1 and R2.2 capture the role relationships. These rules are predictive because they can forecast the expected observation of executing a behavior based on a few previously observed patterns, even in precise situations which may not have been observed before. For example, the data predicts how the roles transition in the presence of 2 nodes, but BioAIM can correctly predict how the roles transition when 3 nodes are participating as well. Therefore, during the training phase BioAIM will continue to improve the learned prediction model until there are no further surprises or the surprises are within predefined acceptable thresholds.

When the learned prediction model is acceptable, data from various successful training scenarios are applied to the prediction model for the purpose of extracting states and state transitions as explained earlier. For example in the given data state one is identified by rules R1 and R2.1 firing simultaneously, while the state two occurs only when rules R1 and R2.2 fire together. So this means that if rules R1 and R2.1 are currently observed the next observation could be the same state one with a probability of 0.3, or it could be state two with a probability of 0.7. However, if state one was observed after state two, then this would be an anomaly based on the trained data. Similarly, if R2.1 and R2.2 fired together it would define an unknown state, which would also be flagged as an anomaly and alerted to the operator. The similarity metric calculation quantifies the likelihood of a state transition or a sequence of state transitions being valid (close to 1.0) or anomalous (close to 0.0).

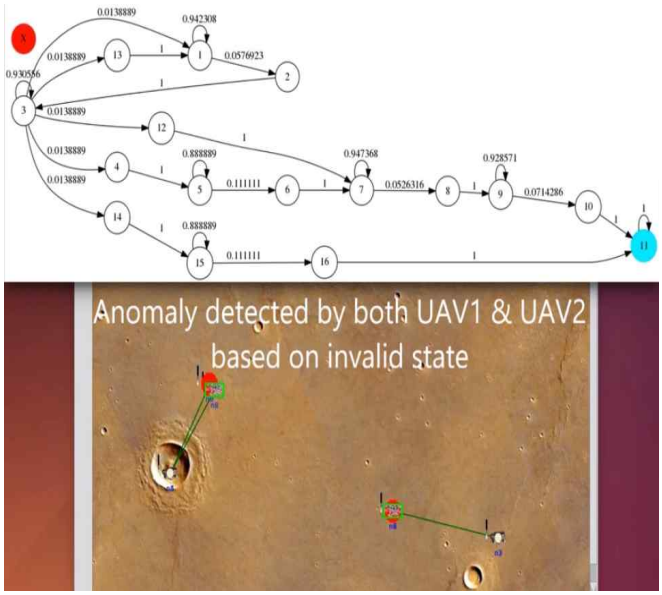

Figure 11: BioMonitor reporting an anomaly

The BioMonitor is a graphical user interface which provides BioAIM feedback to one or more operators. Exposing an appropriate amount of information helps minimize the burden on an operator. If too much information is revealed to the operator when an anomaly occurs, interpreting it to reach the best decision may cause a backlog of events. Similarly, if too little information is revealed the operator may not be able to reach the best decision. The current prototype of the GUI is shown in Figure 11. This screenshot shows the detection of an anomaly which caused a tentacle to sever. The top half of the image contains a graphical representation of the structured model learned by BioAIM with an anomalous state marked in red as "X". This anomaly is reported to the operator as shown in the bottom half of the image.

All of these features make BioAIM ideal for autonomously detecting and alerting a human operator of anomalies at runtime in a swarm of mobile airborne nodes under BioAIR control.

VI. Experimental Results

BioAIM was evaluated on several BioAIR scenarios, where it learned models from synthetic data. In particular data from 10 different scenarios, each comprised of 5 nodes with over 150 sampled time intervals was subjected to statistical analysis. In each scenario, the nodes where allowed to construct a tentacle under normal conditions for 50% of the time, then one or more anomalies such as damaging a node and hijacking nodes (forcefully flying them away in a random direction) was performed to establish if BioAIM would report the presence of anomalies. The number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) detections were recorded for each time interval. The following metrics were calculated:

Precision = TP / (TP + FP) = **1.000**
Recall = TP / (TP + FN) = **0.675**
F1 Score = (2*Precision*Recall)/(Precision+Recall) = **0.806**
Mathews Correlation Coefficient (MCC) = (TP*TN)–(FP*FN)/√(TP+FP)(TP+FN)(TN+FP)(TN+FN)
MCC = **0.722**

The 100% precision indicates that whenever BioAIM detected an anomaly, it was undoubtedly an anomaly, yet the 68% recall indicates that the algorithm missed 32% of true anomalies. The reason for the missed anomalies is primarily due to the delay over several time intervals to establish that the data is indeed anomalous rather than noisy. Another reason is that the synthetic data used during training was insufficient to capture all anomalies experienced during testing. From a practical perspective the high F1 score and MCC indicate that BioAIM can autonomously detect most anomalies and aid a human operator in monitoring a swarm of autonomous UAVs.

VII. Conclusion & Future Work

This paper presents the BioAIM system for autonomously learning the valid behaviors of a swarm of airborne nodes, detecting any anomalous behavior at runtime, and alerting an operator of the anomaly with sufficient information for decision support. In particular, BioAIM learns a prediction model which describes how data from relevant sensors should change when a behavior executes under normal circumstances. It autonomously identifies a dynamic set of states from the prediction model and learns a structured model, which helps quantify the magnitude of divergence from the norm. When in operation BioAIM monitors the sensor data by testing the applicable models for each valid behavior at regular time intervals, and informs the operator of any significant anomaly (i.e. divergence from the norm). In future, BioAIM can be improved to handle known anomalies such as bad weather with predefined contingencies, thereby enabling autonomous adaptation to threats without human intervention.

Acknowledgment

The authors would like to thank the members of the Air Force Research Laboratory for supporting this research and field experiments.

References

[1] B. Ryu, N. Ranasinghe, W. Shen, K. Turck, M. Muccio, "BioAIR: Biologically inspired Artificially Intelligent Reconfiguration", Submitted to IEEE MILCOM Conference, 2015.
[2] W. Shen, "Autonomous Learning From The Environment", New York, W.H. Freeman & Company, 1994.
[3] K. Doya., K. Samejima, K. Katagiri, M. Kawato, "Multiple Model-based Reinforcement Learning", Journal of Neural Computation, 2002.
[4] G. E. Hinton, S. Osindero, and Y. Teh, A fast learning algorithm for deep belief nets, Neural Computation, 18, pp 1527-1554.
[5] A. Graves, G. Wayne and I. Danihelka, Neural Turing Machines, Neural and Evolutionary Computing, 2014.
[6] J. Ahrenholz, "Comparison of CORE Network Emulation Platforms", IEEE MILCOM Conference, 2010, pp.864-869.
[7] N. Ranasinghe and W-M. Shen, "Autonomous Surveillance Tolerant to Interference", Towards Autonomous Robotic Systems conference, TAROS 2012, August 2012.
[8] N. Ranasinghe, "Learning to Detect and Adapt to Unpredicted Changes", Doctoral Dissertation, University of Southern California, August 2012.