

AUTOMATIC INSECT RECOGNITION USING OPTICAL FLIGHT DYNAMICS MODELED BY KERNEL ADAPTIVE ARMA NETWORK

Kan Li and José C. Príncipe

Computational NeuroEngineering Laboratory, University of Florida, Gainesville, FL 32611 USA

ABSTRACT

Automatic insect recognition (AIR), using noninvasive methods *in situ*, has far-reaching implications in entomology, agriculture, and disease control and prevention. An emerging technology in computational entomology uses flight information captured by laser sensors. Current methods treat these optical signals as static patterns, rather than time series. We propose a novel approach to AIR by evaluating each insect passage as a nonstationary process involving a sequence of pseudo-acoustic frames and modeling the short-term flight dynamics using the kernel adaptive autoregressive-moving average (KAARMA) algorithm. Since flight behavior is both nonlinear and nonstationary in nature, dynamic modeling provides a general framework that fully exploits the transitional and contextual information. Results show KAARMA classifier outperforms the state-of-the-art AIR methods, using support vector machine (SVM), deep-learning autoencoder, and batch learning, in identifying Zika vector mosquito *Aedes aegypti* among five species of flying insects, while using significantly more efficient data representation.

Index Terms— Audio processing, computational entomology, dynamic modeling, kernel adaptive filtering.

1. INTRODUCTION

Insects are vital to our ecosystem. The most diverse group of animals, they permeate all terrestrial environments, sharing and often competing for the same resources with humans. They directly impact agricultural production both positively and negatively. Insects make up a majority of the pollinators responsible for over 35% of the worldwide food-crop production volume, and more than 75% of the leading food crops rely on pollinators for quality and/or yield, with annual market estimate of \$577 billion [1]. Approximately 90% of all wild flowering plant species are pollinator-dependent [1], and the distribution and density of insects act as important bioindicators of environmental stress for terrestrial ecosystems [2].

Insects can also be extremely disruptive. Left uncontrolled, many species feed on cash crops, damage stored foods, and destroy building materials. In the U.S. alone,

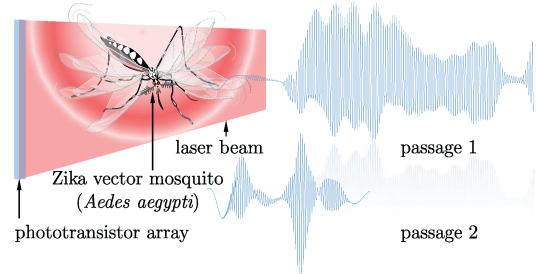


Fig. 1: Optical recordings of Zika vector *Aedes aegypti*.

pesticides were responsible for roughly \$40 billion saved crops [3]. Pesticide expenditures in 2007 reached more than \$39 billion worldwide (\$12 billion in the U.S.) with insecticides accounting for 28% (39%) of the total [4]. Many species of insects are also vectors of diseases and have a profound impact on human and animal health, particularly flying insects in the family *Culicidae* or mosquitoes. Mosquito-borne illnesses include chikungunya, dengue, malaria, West Nile virus, yellow fever, and the recently prevalent Zika fever [5]. The World Health Organization (WHO) estimates that 17% of all infectious diseases are vector borne, responsible for over one million deaths annually, and with over half of the world's population at risk [6]. Birth defects in Brazil have doubled since the Zika epidemic [7]. Due to the lack of vaccines or effective treatment of certain diseases, e.g., Zika, insecticides are used for vector control. However, most methods of applying insecticides, such as aerial spraying for mosquitoes, miss their intended targets and can cause detrimental effects on public health, the environment, and society [3]. For example, behavior changes and colony failures in bees (responsible for almost 80% of all insect pollination) are linked to pesticides [8]. Furthermore, insecticides' effectiveness diminishes over time, as vectors develop increasing resistance [9].

Accurate, automatic, and rapid identification *in situ* is key to combat agricultural pests and disease vectors, and to monitor beneficial insects such as pollinators. Noninvasive and inexpensive intelligent traps are an emerging technology in computational entomology [10, 11]. Flying insects are lured into its entrance using an attractant. Airflow from a fan gently guides it across a laser sensor, consisting of a planar laser source aimed at a phototransistor array (Fig. 1). Fluctuations

This work was supported by DARPA Contracts N66001-10-C-2008 and N66001-15-1-4054.

in light intensity caused by wingbeat occlusions are captured by the phototransistor and analyzed in real-time for classification. Chamber door is opened for positive identification. Otherwise, the insect is released by reversing the fan airflow.

Automatic insect recognition (AIR) is at the core of making intelligent traps a viable solution. Early work on optical flight information examined the wingbeat frequency as the sole discriminating feature for classifying species of fruit flies using stroboscope [12]. More recently, inspired by speech processing, features such as Mel-frequency cepstral coefficients (MFCCs) [13] and linear predictive coding coefficients (LPCs) [14] have been extracted from laser-sensor signals to perform AIR, using machine learning techniques such as support vector machine (SVM), k-nearest neighbors (KNN), decision trees, Gaussian mixture model (GMM), or a combination of algorithms [15, 16]. State-of-the-art results have been reported using deep learning algorithm applied to Mel-spectrum features [17]. Specifically, a class of stacked autoencoder (SAE) [18], trained using maximum correntropy criterion (MCC) [19], coupled with SVM classifier is used.

The major drawback with existing approaches is that insect passages are evaluated as static patterns: rather than analyzing the optical flight information as a time series, it is viewed as a single quasi-stationary acoustic frame. In order to compensate for the variations in signal duration, centering and zero-padding are performed across passages, after filtering, to generate signals of uniform length, with a single vector of cepstral coefficients extracted from an entire passage. However, we see from Fig. 1 that flight recordings within the same species or individual exhibit large variations in duration, with distinct and varying dynamics across passages.

In this paper, we propose a novel approach to AIR by treating each insect passage as a nonstationary process involving a sequence of multiple pseudo-acoustic frames and modeling the short-term flight dynamics using the kernel adaptive autoregressive-moving average (KAARMA) algorithm [20]. KAARMA is an online adaptive algorithm trained with recurrent stochastic gradient descent in reproducing kernel Hilbert spaces (RKHSs) to model spatiotemporal signals using state-space trajectories. It achieves the appropriate memory depth via feedback of its internal states and is trained discriminatively, utilizing the full context of the input sequences. Since flight behavior is both nonlinear and nonstationary in nature, dynamic modeling using KAARMA provides a general framework that fully exploits the transitional and contextual information. Furthermore, it provides native support for sequences of varying length, eliminating the need for zero-padding and centering signals collected from different passages. To the best of our knowledge, this is the first time that insect flight information captured using laser sensors is being modeled as a dynamical system. As a proof of concept, we demonstrate the capabilities of a single multiclass KAARMA network to automatically recognize flying insects using a dataset comprised of three well-known

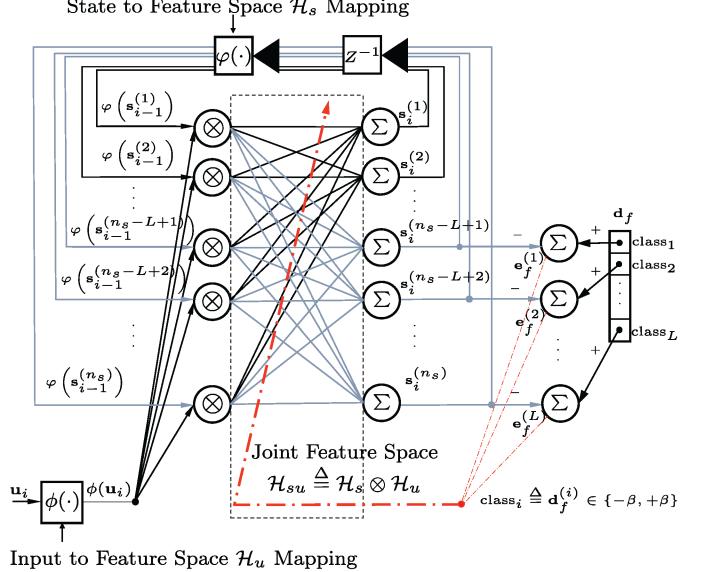


Fig. 2: Block diagram of an L -category KAARMA classifier.

disease vectors and two common fly species. We show that the proposed KAARMA classifier outperforms all previously reported accuracies, including state-of-the-art AIR methods using deep learning autoencoders and batch learning, while using significantly more data-efficient representation.

The remainder of this paper is organized as follows. We briefly review the KAARMA algorithm in Section II. In Section III, we compare the multiclass KAARMA performance with state-of-the-art results. Section IV concludes the paper.

2. MULTICLASS KERNEL ADAPTIVE ARMA ALGORITHM

We briefly describe the KAARMA algorithm. Please refer to [20] for a more in-depth discussion. Let a dynamical system be defined in terms of a general nonlinear state transition function $\mathbf{x}_i = \mathbf{g}(\mathbf{s}_{i-1}, \mathbf{u}_i)$ and an observation function $\mathbf{y}_i = \mathbf{h}(\mathbf{x}_i) = \mathbf{h} \circ \mathbf{g}(\mathbf{s}_{i-1}, \mathbf{u}_i)$, where $\mathbf{u}_i \in \mathbb{R}^{n_u}$ is the input, $\mathbf{x}_i \in \mathbb{R}^{n_x}$ is the hidden state, $\mathbf{y}_i \in \mathbb{R}^{n_y}$ is the output, and $\mathbf{s}_i \triangleq [\mathbf{x}_i, \mathbf{y}_i]^T$ is the augmented state vector. Applying the theory of RKHS, the state-space model (SSM) in the joint RKHS $\mathcal{H}_{su} \triangleq \mathcal{H}_s \otimes \mathcal{H}_u$ can be expressed as the functional weights $\Omega \triangleq \Omega_{\mathcal{H}_{su}} \triangleq \begin{bmatrix} \mathbf{g}(\cdot, \cdot) \\ \mathbf{h} \circ \mathbf{g}(\cdot, \cdot) \end{bmatrix}$, where \otimes is the tensor-product operator. The kernel SSM becomes $\mathbf{s}_i = \Omega^T \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i)$ and $\mathbf{y}_i = \mathbb{I}\mathbf{s}_i$, where $\mathbb{I} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I}_{n_y} \end{bmatrix}$ is a fixed selector matrix, with \mathbf{I}_{n_y} being an $n_y \times n_y$ identity matrix.

Fig. 2 shows a multiclass KAARMA network. The states \mathbf{s}_i are assumed hidden, and a partial deferred desired value or label \mathbf{d}_i may only be observed at the end of a sequence or at the final time index $i = f$. Entries in the L -dimensional label

vectors $\mathbf{d}_f \in \mathbb{R}^L$ are set to be binary, i.e., $\mathbf{d}_f^{(i)} \in \{-\beta, +\beta\}$, where only the correct class has positive polarity, with fixed amplitude $\beta \leq 1$. For a trained network, output unit with the largest positive value indicates the class prediction.

2.1. Stochastic Gradient Descent

The exact error gradient in the RKHS is computed at the end of each input sequence, using the Gaussian kernel $\mathcal{K}_a(\mathbf{u}, \mathbf{u}') = \exp(-a\|\mathbf{u} - \mathbf{u}'\|^2)$, where $a > 0$ is the kernel parameter. The joint inner products are computed using $\mathcal{K}_{a_s}(\mathbf{s}, \mathbf{s}')$ and $\mathcal{K}_{a_u}(\mathbf{u}, \mathbf{u}')$, respectively. The error gradient with respect to the weights in the RKHS at time i is

$$\frac{\partial \varepsilon_i}{\partial \Omega_i} = -\mathbf{e}_i^T \frac{\partial \mathbf{y}_i}{\partial \Omega_i} = -\mathbf{e}_i^T \frac{\partial \mathbf{y}_i}{\partial \mathbf{s}_i} \frac{\partial \mathbf{s}_i}{\partial \Omega_i} \quad (1)$$

where $\mathbf{e}_i = \mathbf{d}_i - \mathbf{y}_i \in \mathbb{R}^{n_y \times 1}$ is the error vector, $\frac{\partial \mathbf{y}_i}{\partial \mathbf{s}_i} = \mathbb{I}$, and the partial derivative $\frac{\partial \mathbf{s}_i}{\partial \Omega_i}$ consists of n_s state terms, $\frac{\partial \mathbf{s}_i}{\partial \Omega_i^{(1)}}, \frac{\partial \mathbf{s}_i}{\partial \Omega_i^{(2)}}, \dots, \frac{\partial \mathbf{s}_i}{\partial \Omega_i^{(n_s)}}$. For the k -th state component of Ω_i , the gradient can be expanded using the product rule as

$$\begin{aligned} \frac{\partial \mathbf{s}_i}{\partial \Omega_i^{(k)}} &= \frac{\partial \Omega_i^T \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i)}{\partial \Omega_i^{(k)}} \\ &= \Omega_i^T \frac{\partial \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i)}{\partial \Omega_i^{(k)}} + \mathbf{I}_{n_s}^{(k)} (\varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i))^T \end{aligned} \quad (2)$$

where $\mathbf{I}_{n_s}^{(k)} \in \mathbb{R}^{n_s \times n_s}$ is the $n_s \times n_s$ identity matrix's k -th column.

Using the representer theorem, Ω_i can be written as a linear combination of prior features $\Omega_i = \Psi_i \mathbf{A}_i$ where $\Psi_i \triangleq [\varphi(\mathbf{s}_{-1}) \otimes \phi(\mathbf{u}_0), \dots, \varphi(\mathbf{s}_{m-2}) \otimes \phi(\mathbf{u}_{m-1})] \in \mathbb{R}^{n_\psi \times m}$ is a collection of the m past tensor-product features with potentially infinite dimension n_ψ , and $\mathbf{A}_i \triangleq [\alpha_{i,1}, \dots, \alpha_{i,n_s}] \in \mathbb{R}^{m \times n_s}$ is the set of corresponding coefficients. The k -th component ($1 \leq k \leq n_s$) becomes $\Omega_i^{(k)} = \Psi_i \mathbf{A}_i^{(k)} = \Psi_i \alpha_{i,k}$. Substituting the expression for Ω_i into the feedback gradient on the right-hand side of (2) and applying the chain rule gives

$$\begin{aligned} \Omega_i^T \frac{\partial \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i)}{\partial \Omega_i^{(k)}} &= \mathbf{A}_i^T \frac{\partial \Psi_i^T \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i)}{\partial \mathbf{s}_{i-1}} \frac{\partial \mathbf{s}_{i-1}}{\partial \Omega_i^{(k)}} \\ &= \underbrace{2a_s \mathbf{A}_i^T \mathbf{K}_i \mathbf{D}_i^T}_{\Lambda_i} \frac{\partial \mathbf{s}_{i-1}}{\partial \Omega_i^{(k)}} \end{aligned} \quad (3)$$

where $\mathbf{K}_i \triangleq \text{diag}(\Psi_i^T \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i))$ is a diagonal matrix with eigenvalues $\mathbf{K}_i^{(j,j)} = \mathcal{K}_{a_s}(\mathbf{s}_j, \mathbf{s}_{i-1}) \cdot \mathcal{K}_{a_u}(\mathbf{u}_j, \mathbf{u}_i)$ and $\mathbf{D}_i \triangleq [(\mathbf{s}_{-1} - \mathbf{s}_{i-1}), \dots, (\mathbf{s}_{m-2} - \mathbf{s}_{i-1})]$ is the difference matrix between state centers of the filter and the current input state \mathbf{s}_{i-1} . Define the state-transition gradient matrix as $\Lambda_i \triangleq \frac{\partial \mathbf{s}_i}{\partial \mathbf{s}_{i-1}} = 2a_s \mathbf{A}_i^T \mathbf{K}_i \mathbf{D}_i^T$. Substituting (3) into (2) yields

$$\frac{\partial \mathbf{s}_i}{\partial \Omega_i^{(k)}} = \Lambda_i \frac{\partial \mathbf{s}_{i-1}}{\partial \Omega_i^{(k)}} + \mathbf{I}_{n_s}^{(k)} (\varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i))^T. \quad (4)$$

The state gradient (4) is independent of any teacher signal, i.e., error \mathbf{e}_i , we can forward propagate the state gradients in

the recursion. The initial state is user-defined and functionally independent of the filter weights. By setting $\frac{\partial s_0}{\partial \Omega_i^{(k)}} = \mathbf{0}$, we can factor out the basis functions and express the recursion as

$$\begin{aligned} \frac{\partial \mathbf{s}_i}{\partial \Omega_i^{(k)}} &= \Lambda_i \mathbf{V}_{i-1}^{(k)} \Psi'_{i-1}^T + \mathbf{I}_{n_s}^{(k)} (\varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i))^T \\ &= [\Lambda_i \mathbf{V}_{i-1}^{(k)}, \mathbf{I}_{n_s}^{(k)}] [\Psi'_{i-1}, \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i)]^T \\ &= \mathbf{V}_i^{(k)} \Psi'_{i-1}^T \end{aligned} \quad (5)$$

where $\Psi'_{i-1} \triangleq [\Psi'_{i-1}, \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i)] \in \mathbb{R}^{n_\psi \times i}$ are centers generated by the input sequence and forward-propagated states from a fixed filter weight Ω_i , and $\mathbf{V}_i^{(k)} \triangleq [\Lambda_i \mathbf{V}_{i-1}^{(k)}, \mathbf{I}_{n_s}^{(k)}] \in \mathbb{R}^{n_s \times i}$ is the updated state-transition gradient, with initializations $\mathbf{V}_1^{(k)} = \mathbf{I}_{n_s}^{(k)}$ and $\Psi'_1 = [\varphi(\mathbf{s}_0) \otimes \phi(\mathbf{u}_1)]$.

Updating the weights in the negative direction yields

$$\begin{aligned} \Omega_{i+1}^{(k)} &= \Omega_i^{(k)} + \eta \Psi'_{i-1} \left(\mathbf{I} \mathbf{V}_i^{(k)} \right)^T \mathbf{e}_i \\ &= [\Psi_i, \Psi'_{i-1}] \begin{bmatrix} \mathbf{A}_i^{(k)} \\ \eta \left(\mathbf{I} \mathbf{V}_i^{(k)} \right)^T \mathbf{e}_i \end{bmatrix} \\ &\triangleq \Psi_{i+1} \mathbf{A}_{i+1}^{(k)} \end{aligned} \quad (6)$$

where η is the learning rate. Since the weights are updated online, to reduce redundancy and improve stability, we evaluate each new center from the feature update Ψ' with existing ones in Ψ , using the vector quantization method outlined in [20, Algorithm 2], controlled by a threshold factor q .

3. SIMULATION RESULTS

Table 1: Summary of the Flying Insect Dataset.

Species	Train	Test	All
Yellow Fever Mosquito (<i>Aedes aegypti</i>)	800	104	904
Common Housefly (<i>Musca domestica</i>)	800	117	917
Common Fruit Fly (<i>Drosophila melanogaster</i>)	800	154	954
Southern House Mosquito (<i>Culex quinquefasciatus</i>)	800	485	1285
Western Encephalitis Mosquito (<i>Culex tarsalis</i>)	800	465	1265
Total	4000	1325	5325

We used the same dataset (Table 1) evaluated in [16] and [17], consisting of 5325 passages collected over six days, in controlled laboratory conditions. The mosquito species are vectors of diseases: *Aedes aegypti* (dengue, West Nile virus, Zika), *Culex tarsalis* (St. Louis encephalitis, west Equine encephalitis, West Nile virus), and *Culex quinquefasciatus* (avian malaria, lymphatic filariasis, West Nile virus).

Recordings were sampled at 16 kHz. Insect passages of varying duration are centered and zero-padded to generate signals of uniform length, e.g., 1 s segments, please refer to [16] for detailed data preparation. Because KAARMA supports spatiotemporal signals of different length, we removed the zero-padding using the threshold-detection method

Table 2: Performance Comparison of Automatic Insect Recognition (AIR) Algorithms.

#Mel	Species	Aedes aegypti	Musca domestica	Drosophila melanogaster	Culex quinquefasciatus	Culex tarsalis	Overall
Method							
Static Pattern (100)	MFCC+SVM	90.0% \pm 2.8%	90.5% \pm 2.2%	82.7% \pm 2.2%	91.0% \pm 1.3%	83.9% \pm 0.8%	87.4% \pm 0.5%
	Mel+SVM	88.4% \pm 2.8%	93.6% \pm 1.6%	84.9% \pm 4.0%	91.5% \pm 1.7%	91.4% \pm 1.3%	90.7% \pm 0.9%
	Mel+KNN	82.4% \pm 4.7%	98.7% \pm 1.0%	80.5% \pm 1.6%	90.1% \pm 1.9%	89.9% \pm 1.1%	89.1% \pm 0.8%
	R-SAE+SVM(50)	89.6% \pm 3.7%	95.2% \pm 1.6%	90.9% \pm 2.2%	92.2% \pm 1.3%	92.2% \pm 1.6%	92.1% \pm 0.7%
	R-SAE+SVM(10)	88.9% \pm 2.5%	94.3% \pm 1.7%	88.2% \pm 2.5%	91.4% \pm 0.9%	92.3% \pm 0.8%	91.4% \pm 0.5%
	R-SAE+Softmax(50)	88.6% \pm 2.9%	93.4% \pm 3.0%	88.4% \pm 2.6%	91.4% \pm 1.4%	92.2% \pm 1.9%	91.3% \pm 0.9%
	R-SAE+Softmax(10)	87.3% \pm 2.7%	91.9% \pm 1.9%	88.3% \pm 2.9%	91.5% \pm 1.8%	92.0% \pm 1.3%	91.0% \pm 0.6%
(40)	KAARMA	91.0% \pm 2.4%	97.1% \pm 1.8%	88.3% \pm 3.4%	93.4% \pm v1.1%	94.3% \pm 1.3%	93.2% \pm 0.7%
(40)	KAARMA	89.42%	95.73%	96.75%	94.23%	96.77%	95.17%
(20)	KAARMA	88.46%	96.58%	91.56%	95.46%	94.41%	94.19%
(12)	KAARMA	96.15%	97.44%	85.71%	94.02%	90.11%	92.15%
Static, Batch (50)	SVM, PUK kernel with $\omega, \sigma = 1$						93.93%
	Random Forest, Trees = 15						92.09%
	KNN, Neighbors = 1, Euclidean dist.						92.08%
	GMM, Gaussians = 10						91.56%
	RBF Network, Clusters = 5						89.92%

in [17]. This significantly reduced the data, since most are less than 200 ms. We segmented each recording into 20 ms frames, at 100 fps rate. Signals are bandpass-filtered from 10-4000 Hz, with 40 MFCCs extracted per frame using a bank of 60 filters (pre-emphasis coefficient 0.97 and cepstral sine lifter parameter 22). We use binary vector labels (magnitude $\beta = 0.25$) to train the 40-dimensional MFCC sequences (input kernel parameter $a_u = 2$, hidden state dimension $n_x = 3$, and state kernel parameter $a_s = 1$) for 25 epochs, with learning rate $\eta = 0.05$ and quantization factor $q = 0.45$.

The top half of Table 2 shows averaged test accuracies over 10 independent trials with \pm one standard deviation. The best performance in each column is highlighted in bold. Note, individual accuracies are less indicative of the performance for multiclass task; inferior classifiers often overfit to certain training classes. First seven rows are results published in [17], with state-of-the-art AIR algorithm being a robust stacked autoencoder (R-SAE) coupled with a SVM. The R-SAE takes 100 Mel-scale features (KAARMA uses 40) and outputs a 50-d vector (KAARMA outputs 8: 3 hidden states plus 5 label states). The R-SAE was trained using the maximum correntropy criterion, compared with the simple mean squared error (MSE) criterion used in KAARMA. For the reported accuracies, a separate SVM classifier with radial basis function (RBF), or Gaussian, kernel was trained on the R-SAE outputs. Despite using a more parsimonious architecture, multiclass KAARMA network performed the best with an average overall accuracy of 93.2%. The bottom half of Table 2 pits the best multiclass KAARMA classifiers, using 40, 20, and 12 MFCCs, against the five best batch-learned classifiers (50 MFCCs) in [16]. The batch results serve as upper reference performances for static pattern learning. Our online adaptive classifier easily beats the batch classifiers, with

the best overall performance of 95.2%. When the number of MFCCs were reduced to 20, KAARMA still outperformed the batch methods. Even with only 12 MFCCs, KAARMA maintained a competitive accuracy of 92.2%. Clearly, more information lie in the flight dynamics than in the number of cepstral features, past a certain threshold. Dynamic modeling using KAARMA fully exploits the transitional and contextual information within the signals to achieve the best results.

4. CONCLUSION

We presented a novel approach to identifying flying insects using optically recorded flight information. Since flight behavior is both nonlinear and nonstationary in nature, dynamic modeling using KAARMA provides a general framework that fully exploits the transitional and contextual information. Results demonstrate the proposed multiclass KAARMA classifier outperforms the state-of-the-art AIR methods involving SVM and deep learning autoencoders, while using significantly more data-efficient representation. KAARMA leverages fewer features per frame using transitional information from multiple frames in each recording to achieve an even better performance than batch learning using static patterns. Our novel approach opens the door to many solutions in computational entomology and can be applied to other problem domains involving short-term dynamics.

Acknowledgment

The authors would like to thank Prof. Eamonn Keogh, University of California, Riverside, and Prof. Gustavo E. A. P. A. Batista, University of Sao Paulo, for sharing the dataset.

5. REFERENCES

- [1] S. G. Potts, V. Imperatriz-Fonseca, H. T. Ngo, J. C. Biesmeijer, T. D. Breeze, LynnV.Dicks, L. A. Garibaldi, R. Hill, J. Settele, and A. J. Vanbergen, “The assessment report on pollinators, pollination and food production: summary for policymakers,” *Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services (IPBES)*, 2016.
- [2] P. G. Kevan, “Pollinators as bioindicators of the state of the environment: species, activity and diversity,” *Agriculture, Ecosystems & Environment*, vol. 74, no. 1-3, pp. 373–393, 1999.
- [3] D. Pimentel, “Environmental and economic costs of the application of pesticides primarily in the United States,” in *Integrated Pest Management: Innovation-Development Process: Volume 1*, R. Peshin and A. K. Dhawan, Eds. Springer Netherlands, 2009, pp. 89–111.
- [4] A. Grube, D. Donaldson, T. Kiely, and L. Wu, “Pesticides industry sales and usage 2006 and 2007 market estimates,” *U.S. Environmental Protection Agency*, 2011.
- [5] E. Petersen, M. E. Wilson, S. Touch, B. McCloskey, P. Mwaba, M. Bates, O. Dar, F. Mattes, M. Kidd, G. Ippolito, E. I. Azhar, and A. Zumla, “Rapid spread of zika virus in the americas - implications for public health preparedness for mass gatherings at the 2016 brazil olympic games,” *International Journal of Infectious Diseases*, vol. 44, pp. 11–15, 2016.
- [6] WHO, “A global brief on vector-borne diseases,” World Health Organization, Geneva, Switzerland, Tech. Rep. WHO/DCO/WHD/2014.1, Mar. 2014.
- [7] C. Barcellos, D. R. Xavier, A. L. Pavao, C. S. Boccolini, M. F. Pina, M. Pedroso, D. Romero, and A. R. Romao, “Increased hospitalizations for neuropathies as indicators of zika virus infection, according to health information system data, Brazil,” *Emerg Infect Dis.*, vol. 22, no. 11, Nov. 2016.
- [8] R. J. Gill, O. Ramos-Rodriguez, and N. E. Raine, “Combined pesticide exposure severely affects individual- and colony-level traits in bees,” *Nature*, vol. 491, pp. 105–108, 2012.
- [9] S. Sathantripop, P. Paeporn, and K. Supaphathom, “Detection of insecticides resistance status in culex quinquefasciatus and aedes aegypti to four major groups of insecticides,” *Tropical Biomedicine*, vol. 23, pp. 97–101, 2006.
- [10] D. Silva, V. Souza, and E. K. a. E. G.E.A.P.A. Batista, “SIGKDD Demo: Sensors and software to allow computational entomology, an emerging application of data mining,” in *Proc. 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, CA, USA, 2011, pp. 761–764.
- [11] R. J. Gill, O. Ramos-Rodriguez, and N. E. Raine, “Exploring low cost laser sensors to identify flying insect,” *Journal of Intelligent and Robotic Systems*, vol. 80, no. Suppl 1, pp. S313–S330, 2015.
- [12] S. Reed, C. Williams, and L. Chadwick, “Frequency of wing-beat as a character for separating species races and geographic varieties of drosophila,” *Genetics*, vol. 27, pp. 349–361, 1942.
- [13] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980.
- [14] B. S. Atal and S. L. Hanauer, “Speech analysis and synthesis by linear prediction of the speech wave,” *J. Acoust. Soc. Am.*, vol. 50, p. 637, 1971.
- [15] D. Silva, V. Souza, and E. K. a. E. G.E.A.P.A. Batista, “Applying machine learning and audio analysis techniques to insect recognition in intelligent traps,” in *Proc. ICMLA*, Miami, Florida, USA, 2013, pp. 99–104.
- [16] V. M. A. D. Souza, D. F. Silva, and G. E. A. P. A. Batista, “Classification of data streams applied to insect recognition: Initial resultss,” in *Proc. BRACIS*, Fortaleza, Ceara, Brazil, 2013, pp. 76–81.
- [17] Y. Qi, G. T. Cinar, V. M. A. Souza, G. E. A. P. A. Batista, Y. Wang, and J. C. Principe, “Effective insect recognition using a stacked autoencoder with maximum correntropy criterion,” in *Proc. IJCNN*, Killarney, Ireland, 2015, pp. 1–7.
- [18] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proc. ICML*, Helsinki, Finland, 2008, pp. 1096–1103.
- [19] W. Liu, P. Pokharel, and J. Principe, “Correntropy: Properties and applications in non-gaussian signal processing,” vol. 55, no. 11, pp. 5286–5298, Nov. 2007.
- [20] K. Li and J. C. Príncipe, “The kernel adaptive autoregressive-moving-average algorithm,” *IEEE Trans. Neural Netw. Learn. Syst.*, 2015.