Active Estimation of Motivational Spots for Modeling Dynamic Interactions

Juan Sebastian Olier[1, 2], Damian Andres Campo[1], Lucio Marcenaro[1], Emilia Barakova[2], Matthias Rauterberg[2], and Carlo Regazzoni[1, 3]

[1] Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture, University of Genoa, Genoa, Italy

[2] Department of Industrial Design, Eindhoven University of Technology, Eindhoven, The Netherlands

[3]Intelligent Systems Lab, University Carlos III de Madrid, Madrid, Leganés, Spain

J.S.Olier.Jauregui@tue.nl, carlo.regazzoni@unige.it

Abstract

To understand the behavior of moving entities in a given environment, one should be capable of predicting their motion, that is, to model their dynamics. In a setting where different behaviors can arise, one can assume that each of them corresponds to different motivational states of observed entities. Here, those motivations are understood as goal positions or spots where entities seek to arrive. To build prediction models based on that idea, we present an unsupervised method to estimate motivational spots actively. Additionally, we use the output of such process to refine an adaptive system modeling the dynamics of inferred hidden causes of observed data. The whole method uses deep variational methods, and particularly, the network estimating motivations is trained through dynamic programming. Results show that modeling the dynamics of entities can be better achieved by integrating information about motivational spots. Notably, a network modeling the dynamics converges faster through the incorporation of information about motivations.

1. Introduction

Understanding and interpreting how different regions in a monitored environment relate to observed behaviors is an important task to improve the prediction and awareness of surveillance systems[1, 17]. To that aim, here we assume causal relations between the behavior of moving entities and the elements or regions in a given scene. Furthermore, we assume that, when observing crowds, the dynamics of moving entities are governed by interactive cognitive behaviors [4, 5]. Thus, the effects exerted on entities by a particular spot, or area in the environment can be estimated through the observed motion of entities. Such environmental spots could be, for example, bodies being avoided, or open paths attracting the entities.

Moreover, some works [2, 7, 14] have demonstrated the possibility of modeling movements of entities as a result of forces acting on them, which facilitates to deduce properties of unobserved entities. Particularly, it has been shown how force models can accurately describe motions of pedestrians [3, 6, 15]. Such models individuate general types of effects on the dynamics of moving entities by hypothesizing that external factors exert forces on them, thus causing observed behaviors.

Thereby, in this work, observed motions are assumed to be produced by external causes associated with changing motivational spots in the environment. Accordingly, observing the movement of various entities can reveal how the causes evolve throughout a typical interaction in a given environment.

Moreover, spots here are defined actively along the development of the interaction between a particular entity and the environment, since the spots conducting the dynamics of the moving entity can change over time. For example, an agent may be attracted by a particular area in the scene, but at a given time, the area of interest can change to different spots throughout the interaction.

In general, a point in the environment attracting an entity at a given time is understood as a motivational spot. Thus, locating such motivators can provide useful information for training models that predict more accurately the dynamics of new observed entities. Moreover, that can be used for detecting potential abnormalities when groups of new entities start following dynamics that deviate from the regular motions and motivations.

The work in [3] has demonstrated the possibility of iden-

978-1-5386-2939-0/17/$31.00

tifying the position of possible attractive points and quantizing their common effects by observing the dynamics of moving entities. In that case, motivations are assumed to be discrete and fixed; however, motivations could also be described more dynamically, evolving together with the observed motion.

Moreover, in [3], angular information is used as criteria to obtain trajectory segments that point in similar directions, i.e., that describe a quasi-constant angle, which is in turn used to estimate corresponding motivations. In this work, no preprocessing of such kind is assumed; but observed sequences are used to train a neural network (NN) model through an algorithm based on dynamic programming.

Thus, the primary goal of the method proposed here is to estimate motivational spots actively and to evaluate their influence in the capability of modeling the dynamics of moving entities.

With that aim, in the method proposed, a first NN actively estimates motivations from the observation of single data points. Such results are then used to improve the training and accuracy of a second network, which encodes the dynamics of observed entities and thus is capable of dynamically making predictions in a kind of probabilistic filtering process. That is achieved without imposing on the system any particular previous knowledge about the data.

The proposed methodology is focused on the movement of groups of entities following similar dynamics. Our method is evaluated on simulated data. Found results demonstrate the capabilities of the approach at dynamically predicting destination points and modeling the observed behaviors. Such features, we argue can be useful, for example, in abnormality detection tasks.

The novelty of this work relies on the usage of deep learning techniques for dynamically estimating motivational spots that govern the motion of cognitive entities. Notably, that is achieved through a training based on dynamic programming, which extracts patterns from observing groups of entities.

The rest of the paper is organized as follows: The proposed methodology is presented in section 2, implementation details are shown in section 3, the dataset is described in section 4, results are exposed in section 5 and conclusions are provided in section 6.

2. Method

There are two main objectives for the proposed method, first to dynamically estimate the motivational points of observed entities, and secondly to describe their dynamics to predict future states and filter observations. The model achieves those goals by encoding two kinds of information into different representational levels learned directly from the input data. The inputs correspond to streams of tracking points of entities interacting with a given environment.

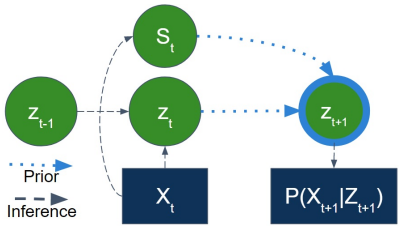

Figure 1. Schema of the proposed model. Causes of observed data are denoted by Z and motivations by S. Input data is denoted by X. Light blue represents prior calculations, while dashed lines depict inference or recognition processes.

Each data point includes the position and the velocity at each time step.

The two kinds of representations are understood as causes of observed data and behaviors. Firstly, a kind of representation encodes the motivations, which denote the causes for the motion dynamics of an observed entity over time. The second kind of representation is related to the causes of the observed data at a particular time. The motivations are denominated as S, while the hidden states generating the observations are denominated as Z (see figure 1).

Z is expressed in terms of Gaussian distributions that are used to make predictions about the following time step by a kind of Bayesian filtering. Such process implies modeling the dynamics of the observed behavior, which is assumed to be actively affected by the motivations.

Similarly, a motivation is expressed as the expected position of an entity in the future (more that a single time step) given its current position and velocity. Each of the parameters of such prediction is to be defined by a Gaussian distribution, where the variances encode the uncertainty of predicted positions. Since motivations S are affecting the transition model for Z, they could be interpreted as a switching variable in a probabilistic graphical model. In particular, that is achieved by including S in the calculation of priors as explained below (see figure 1).

The model is thought as a three step process. i) Causes and motivations are estimated. ii) Inferred Z_t and S_t are used to predict Z_{t+1} (prior). iii) Based on the predicted Z_{t+1}, the expected observation X_{t+1} is estimated.

The proposed method is based on variational deep generative models. In such models, continuous variational representations can be constructed directly from data in an end-to-end fashion [9, 13]. Similarly, in other works [10, 8], Bayesian filters have been modeled with NNs. In the proposed approach we take into account relevant issues from both the generative models and filtering approaches.

The model here proposed is inspired by the ideas in [12], where the definition of concepts is elaborated in favor of more dynamic accounts. In that work concepts are understood as evolving phenomena encoding environment-action

relationships, in opposition to views on concepts as categories or symbols. In relation to the problem at hand, motivations are not understood as static entities that can be classified, but as dynamic phenomena evolving with the interaction between the moving entities and the environment.

Moreover, the here proposed method is also partially based on the one in [12]. However, the transition model here is based only on the immediately previous states instead of using LSTMs for encoding sequences. Equally, the motivations have been added to dynamically modify the transition model.

2.1. Description

The system and its training are both separated in two stages. First, the network estimating motivations (S) is trained; such network is called motivations-NN. Then, based on the motivations predicted, and the input data, another NN is trained to estimate the internal states (Z) and model the observed dynamics. Such network is named dynamics-NN.

The training of both NNs is understood as unsupervised since, for the motivations-NN, the target position is never given explicitly, but it has to be inferred. In fact the target position is unknown. Similarly, for the dynamics-NN, the target representations are not given, nor any information different to the data itself.

2.2. Motivations estimation

The motivations-NN is to predict the goal position from a given measurement. To that aim, we propose an algorithm based on dynamic programming and inspired by deep reinforcement learning[11].

To apply dynamic programming we elaborated the following reasoning. If one assumes that a change in the input from X_t to X_{t+1} is linear and given by the difference between them, i.e., ΔX_t; then, after τ time steps, the final state can be expressed as $X_{t+\tau} - X_t = \sum_{i=0}^{\tau}(\Delta X_{t+i})$, or equivalently, $\Delta X_\tau = \Delta X_t + \sum_{i=1}^{\tau}(\Delta X_{t+i})$

Due to the lack of evidence related to future estimations, every term in the sum is exponentially weighted by a discount factor $\gamma < 1$, which relates to such uncertainty. Thus:

$$\Delta X_\tau = \Delta X_t + \sum_{i=1}^{\tau}(\gamma^i \Delta X_{t+i})$$

As explained in [11], the higher γ is, the further in time the states are being predicted.

Given that τ is not explicitly defined, the training is not constrained to any specific sequence. In particular, one can assume τ to be big in relation to the time frame of the interaction, or in general approaching to infinity. Then, under that assumption and by letting $\phi^S(X_t) = \Delta X_\tau$, one gets:

$$\phi^S(X_t) = \Delta X_t + \gamma \phi^S(X_{t+1}) \quad (1)$$

where ϕ^S would be modeled by the motivations-NN. The recursion shown in (1) facilitates the training of ϕ^S with dynamic programming as proposed in [11].

However, to also encode uncertainty, we consider instead $S_t \sim \mathcal{N}\left(\mu_{S_t}, diag(\sigma^2_{S_t})\right)$, where $[\mu_{S_t}, \sigma_{S_t}] = \varphi^s(X_t)$, with μ_{S_t} approximating $\phi^S(X_t)$, and σ_{S_t} encoding variability in the data. $\varphi^s(X_t)$ is the motivations-NN.

2.2.1 Training

The training of φ^s is performed by means of two different networks (φ^s and $\varphi^{s'}$) to optimize $\varphi^s(X_t) = \Delta Z_t + \gamma\varphi^{s'}(X_{t+1})$. The weights of $\varphi^{s'}$ are kept constant for k time steps (in this implementation $k = 10$), during which the weights of φ^s are updated to maximize $log(P(S_t|X_t)$; that is, the likelihood of the distribution S_t generating $\Delta Z_t + \gamma * \varphi^{s'}(X_{t+1})$ given X_t. After the k steps, the weights from φ^s are copied to $\varphi^{s'}$. Thereby, the training is performed through stochastic gradient descent (SGD), maximizing $log(P(S_t|X_t)$, and using as training data points (X_t, X_{t+1}) pairs.

2.3. Modeling of motion dynamics

The dynamics-NN is in charge of estimating dynamical causes, i.e., Z values. It is based on the principles of variational auto-encoders (VAE) [9]. In a VAE, an observation X is used to infer latent variables Z by capturing variations in X. An inference model approximates $P(Z|X)$, and is assumed as Gaussian: $Z \sim \mathcal{N}\left(\mu_z, diag(\sigma^2_z)\right)$, with $[\mu_z, \sigma_z] = \varphi^Z(X)$ approximated by a NN. The prior $P(Z)$ and $P(X|Z)$ are assumed to be Gaussian distributions. $P(Z|X)$ is sampled as $z = \mu_z + \sigma_z\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$. To train such model the Kullback-Leibler (KL) divergence between $P(Z|X)$ and $P(Z)$ is minimized, and $log(P(X|Z))$ maximized.

In our architecture, we include time dependencies between consecutive states in a sequence. Thus, the generative model instead of regenerating the input directly from Z_t, it learns a transition dynamics from which the prior $P(Z_{t+1}|Z_t, S_t)$ is estimated, and in turn used to predict the input X_{t+1}. Such prior is calculated by:

$$Z_{t+1} \sim \mathcal{N}\left(\mu_{o_t}, diag(\sigma^2_{o_t})\right)$$

where $[\mu_{o_t}, \sigma_{o,t}] = \varphi^{prior}(Z_t, S_t)$.

Since motivations are understood as the approximation of future changes in position and are related to how state transitions occur, then φ^{prior} takes S_t as an argument. That way, the motivations are used for modeling the dynamics and to improve estimations. A NN approximates φ^{prior}.

From the prior, $P(X_{t+1}|Z_{t+1})$ is estimated. As in the VAE, the generated prediction is assumed to be Gaussian:

$$X_{t+1} \sim \mathcal{N}\left(\mu_{x_{t+1}}, diag(\sigma^2_{x_{t+1}})\right)$$

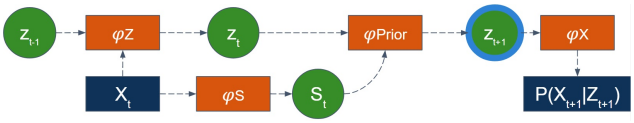

Figure 2. Implementation schematic. Orange boxes are NNs, green circles representational states, and dark blue boxes correspond to inputs, either real or predicted.

where $\left[\mu_{x_{t+1}}, \sigma_{x_{t+1}}\right] = \varphi^X(Z_{t+1})$ is approximated by a NN.

The recognition model $P(Z_t|X_t, Z_{t-1})$ is given by $[\mu_{z_t}, \sigma_{z_t}] = \varphi^Z(X_t, Z_{t-1})$ and approximated by a NN.

2.3.1 Training

There are two objectives optimized through SGD, the likelihoods of prediction generation, and the KL divergence between priors and recognition.

The training for this network then maximizes the following expression for each sample sequence:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}[-KL\left(P(Z_t|X_t, Z_{t-1})||P(Z_t|Z_{t-1}, S_t)\right) + log(P\left(X_{t+1}|Z_{t+1}\right))$$

The training is performed over sequences of $T = 10$ time steps. It is important to note that given $P(Z_t|X_t, Z_{t-1})$, the gradient goes through the transition back to the recognition model directly and not only by the KL divergence. The advantages of that have been shown in [8].

3. Implementation details

As mentioned, the architecture is implemented using NNs (see figure 2):

φ^Z is encoded through what for simplicity we call a GaussCod, which is composed of two fully connected layers (FC) with n units, followed by two separated FC of size Z_t, one encoding the means and other the variances. φ^{prior} is as well a GaussCod, with input size given by the sum of sizes of Z_t and S_t.

For φ^X a GaussCod is used, with output of the size of X_t, and input of the size of Z_t. $\varphi^s(X_t)$ is built with 3 FC of n units each, followed by a GaussCod with output equals to the size of S_t.

The input to both networks (X_t) is a stack of the position and the velocity of an entity at a given time. The size n is 512, and the size of Z_t and S_t is 2. All the training is performed using the Theano library [16].

4. Dataset

A simulated dataset is considered, where moving entities cross a scene to reach a common objective while avoiding an obstacle. In order to model the motion of entities, a couple of velocity fields related to the effect that attractive and repulsive objects exert on moving entities are considered. The attractive velocity field is defined as:

$$\vec{v}_{attract} = \left(\beta_0 + e^{\frac{(||X-X_0||_2)^2}{\alpha_0}}\right)\hat{r}_0$$

where X is the agent position, β_0 represents the velocity with which entities arrive to the final destination (X_0). α_0 defines how entities change their velocities as they approach to the attractive point, and $\hat{r}_0$ indicates a radial symmetry in the direction of X_0. Similarly, the repulsive velocity force is defined as:

$$\vec{v}_{repulse} = -\left(\beta_1 + e^{\frac{(||X-X_1||_2)^2}{\alpha_1}}\right)\hat{r}_1$$

where β_1 represents the minimum velocity with which an entities is repelled from X_1. α_1 defines how entities are deviated as they approach the repulsive object and $\hat{r}_1$ indicates a radial symmetry in the direction of X_1.

In the tested scenario, parameters are set as follows: $\beta_0 = 0.4$, $\beta_1 = 1.6$, $X_0 = (0, 20)$, $X_1 = (0, 0)$, $\alpha_0 = 400$, $\alpha_1 = 1000$. A layout of the scene is shown in figure 3, where the trajectories of sample entities are depicted.

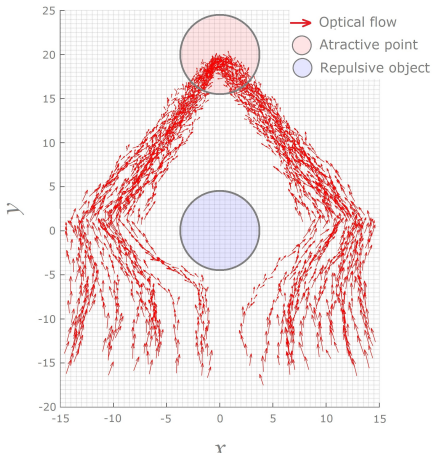

Figure 3. Layout of the produced training data. Entities move from bottom to top in the scene while avoiding an obstacle in the middle.

The final training data is generated by repeating this process twice with different starting points and objectives. One as depicted in figure 3, and the second with $X_0 = (-15, 0)$, $X_1 = (0, 0)$, that is, the same configuration rotated 90 degrees. In that way, there are areas where different entities have similar velocity, even when moving towards diverse final points.

5. Results

5.1. Motivation estimations

Figure 4 illustrates an example of how motivations are estimated from measurements. Each green point represents

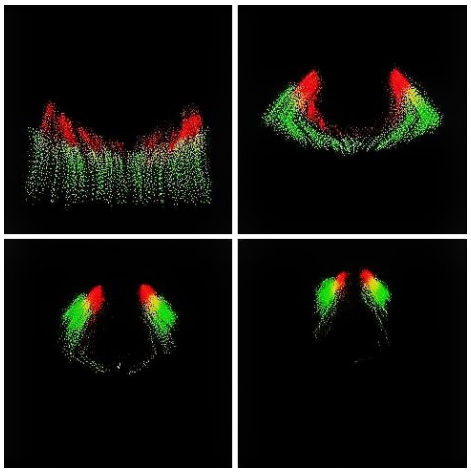

Figure 4. Anticipation example showing the evolution of estimated motivations over time. Green corresponds to current position of entities, and red to the estimated motivation. Time increases from left to right, and top to bottom.

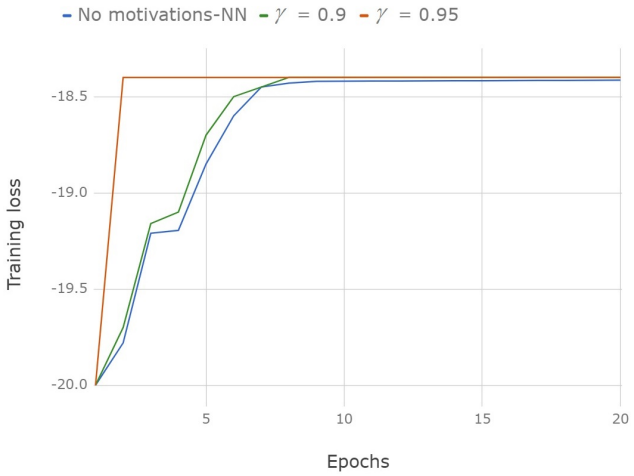

Figure 5. Comparison of the evolution of loss over 20 training epochs for the dynamics-NN. Three cases are considered, when no motivations-NN is used, and when it is used and trained with $\gamma = 0.9$ and $\gamma = 0.95$.

a single entity in a specific time step, and each of them has its corresponding motivational spot in the same image (red points). The depicted example shows a situation in which entities follow the scenario described in section 4. During the trajectory, entities go from bottom to top avoiding an obstacle in the middle of the scene (see figure 3).

The trajectories of all entities were aligned on time; figure 4 shows only four different time steps. At each of those steps, all entities are depicted as they appear in the scene at that given time instance on the alignment. Thus, from left to right and top to bottom, the four frames of figure 4 can be read as the temporal evolution of all the entities moving together along with their motivational points. That can be interpreted as a dynamic clustering of entities based on their position and velocity at a given time. The motivations-NN was trained with a $\gamma = 0.9$ for that example.

5.2. The effect of motivation on the dynamic model

The effect of motivations on the dynamics-NN is evaluated by training such network with and without S_t as an input to $\varphi^p rior$. Two different values of γ are used. The results are shown in figure 5, where it can be seen that the dynamics-NN converges faster when using motivations. This also suggests that the NN uses motivations to make better predictions, as well as to estimate the causes of observed data, which is exactly the intended purpose. The difference is greater for $\gamma = 0.95$, suggesting that the further in time the motivations are estimated (higher γ), the more useful they become to estimate the dynamics of an entity.

6. Discussion

In this work, a method to estimate motivations of moving entities whose motions depend on dynamic goal positions has been developed. Results from such estimations have been used as an input to train a second network that models the dynamics of entities and infers causes of their states over time. All of this is achieved in an unsupervised manner. Thus, the method provides a way to model an environment directly from observing the movement of entities in it. Moreover, it encodes the relations between motivations and the motion dynamics of moving entities.

Results show that the model adapts the dynamic models describing the motion of an entity in relation to estimated motivations while the observed interaction evolves. Such capability facilitates a complete analysis of data without the need of clustering or separating it in terms of, for example, trajectory classes or activities. That can be particularly useful for abnormality detection since unexpected changes in motivations over time could be easily interpreted as anomalies. However, that would require an additional analysis on the evolution of motivations.

In future works, this method is to be evaluated on more complex data coming from actual surveillance videos.

Acknowledgments

-This work was partially supported by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA n 2010-0012.

-Carlo Regazzoni has contributed to produce this work partially under the program UC3M-Santander Chairs of Excellence.

References

[1] M. Bhatt and H. Guesgen. *Situational Awareness for Assistive Technologies - Volume 14 Ambient Intelligence and Smart Environments*. IOS Press, Amsterdam, The Netherlands, 2012.

[2] D. Campo, V. Bastani, L. Marcenaro, and C. Regazzoni. Incremental learning of environment interactive structures from trajectories of individuals. volume 19, pages 589–596, 2016.

[3] D. Campo, A. Betancourt, L. Marcenaro, and C. Regazzoni. Static force field representation of environments based on agents' nonlinear motions. *EURASIP Journal on Advances in Signal Processing*, 2017(1):13, 2017.

[4] S. Haykin. Cognitive dynamic systems. *Proceedings of the IEEE*, 94(11):1910–1911, Nov 2006.

[5] S. Haykin. *Cognitive Dynamic Systems: Perception-action Cycle, Radar and Radio*. Cambridge University Press, New York, NY, USA, 2012.

[6] D. Helbing and A. Johansson. Pedestrian, crowd, and evacuation dynamics. *Encyclopedia of Complexity and Systems Science*, 16:1–28, 2009.

[7] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995.

[8] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[9] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[10] R. G. Krishnan, U. Shalit, and D. Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[12] J. S. Olier, E. Barakova, C. Regazzoni, and M. Rauterberg. Re-framing the characteristics of concepts and their relation to learning and cognition in artificial agents. *Cognitive Systems Research*, 2017.

[13] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1278–1286, 2014.

[14] C. Rudloff, T. Matyus, S. Seer, and D. Bauer. Can Walking Behavior Be Predicted? *Transportation Research Record: Journal of the Transportation Research Board*, 2264(1):101–109, 2011.

[15] S. Seer, C. Rudloff, T. Matyus, and N. Brndle. Validating social force based models with comprehensive real world motion data. *Transportation Research Procedia*, 2:724 – 732, 2014.

[16] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[17] R. Zhen, Y. Jin, Q. Hu, Z. Shao, and N. Nikitakos. Maritime anomaly detection within coastal waters based on vessel trajectory clustering and nave bayes classifier. *Journal of Navigation*, 70(3):648–670, 2017. cited By 0.