

Identifying Fishing Activities from AIS Data with Conditional Random Fields

Baifan Hu, Xiang Jiang

Erico N de Souza, Ronald Pelot

Faculty of Computer Science

Dalhousie University Halifax, NS, Canada B3H 1W5

Email: {baifanhu, Xiang.Jiang, erico.souza, Ronald.Pelot}@dal.ca

Stan Matwin

Faculty of Computer Science,

Dalhousie University Halifax, NS, Canada B3H1W5,

and Institute of Computer Science

Polish Academy of Sciences

Email: stan@cs.dal.ca

Abstract—Fishing activity detection is important for fishery management to maintain abundant oceans. This paper presents a novel approach to identifying fishing activities from Automatic Identification System (AIS) data using Conditional Random Fields (CRFs). CRFs are popular for solving structured prediction problems such as sequence labeling in natural language processing. To model the conditional probability distributions that can identify fishing activities of the vessel points, we treat attributes of vessel points as observed variables and the fishing and non-fishing labels as hidden variables. We present three experiments and two comparisons to demonstrate the stability and effectiveness of the resulting models.

I. INTRODUCTION

GLOBAL overfishing causes a dramatic decline in the fish population. Several major commercial fish species are endangered which threatens the ocean ecosystem. This affects millions of people who depend on fish for food and living. According to a 2014 report by the United Nations Food and Agriculture Organization¹, more than 90 percent of global fisheries are over exploited. Unfortunately, these fishing activities are often illegal, unreported and unregulated (IUU), which makes tracing them a challenge. Thus, in order to make fishing activities more transparent for practicing sustainable fisheries, fishing activity detection is urgently needed.

In 2000, International Maritime Organization (IMO) firstly introduced the Automatic Identification System (AIS) to enhance the security and safety of maritime navigation. Ships equipped with AIS can automatically broadcast information, including unique identification (Maritime Mobile Service Identity, MMSI), position, speed, course and further details of the vessel to its nearby ships and coastal authorities. AIS data are openly accessible and not encrypted. In 2008, satellites AIS technology was implemented, enabling the collection of massive and reliable information of vessels in global areas within seconds. Consequently, satellites AIS data could be used as an ideal source to monitor vessel movements and detect fishing activities around the world.

In this paper, we present a novel approach for identifying fishing activities using Conditional Random Fields and demonstrate its stability of performance using three different

evaluation experiments and two comparisons. The remainder of the paper is organized as follows: In Section 2, we describe relevant literature; in Section 3, we explain conditional random fields (CRFs) and then elaborate on how to apply them to identify fishing activities; in Section 4, we present three experiments and two comparisons and their results on historical AIS data to demonstrate the stability and effectiveness of our models; finally, in Section 5, we discuss practices and provide direction for future work.

II. BACKGROUND AND RELATED WORK

CRFs are prevalent in solving structured prediction problems [1]. It has been applied to many tasks in natural language processing (NLP) such as part-of-speech (POS) tagging [2], [3], shallow parsing [4] and name-entity recognition (NER) [5], [6]. In addition, Hierarchical CRFs [7] has been applied to extract human activities. We find similarity between these tasks and fishing activity detection from the following perspective. In POS tagging, the goal is to label words in sentences using word-category tags. The labels depend on both the word's meaning and context. This task involves two random variables, X and Y , where X is a sequence of words, and Y is a sequence of POS tags. Linear-chain conditional random fields can model the conditional probability distribution $p(y|x)$ to predict POS tags. Similarly, the task of fishing activity detection involves two random variables, X and Y , where X is the observed random variable (which represents sequences of coordinates and speeds), and Y is the hidden random variable to be predicted (Y is a sequence of fishing and non-fishing labels). Consequently, it is reasonable to test whether the linear-chain conditional random fields can model the conditional probability distribution $p(y|x)$ of fishing–non-fishing to detect fishing activities.

Most studies of fishing activity detection focus on Trawlers. For example, Mazzarella et al. identified fishing events using a clustering method [8], and Peel and Good recognized vessel activities using Hidden Markov Model [9]. It is found that trawler fishing activities are highly related with speed, meaning speed can provide useful information to aid the classification of fishing activities. However, for longliners, Souza et al. found that there is no obvious pattern to distinguish fishing activities using speed information alone [10]. Souza et al.

¹Food and Agriculture Organization of the United Nations, 2014. <http://www.fao.org>

applied Lavielle's unsupervised trajectory segmentation algorithm, inspired by animal movements, to identify the longliner fishing behavior. Jiang et al. applied a deep learning approach using autoencoders (AE) that are pretrained with restricted Boltzmann Machines [11]. To the best of our knowledge, [10] was the first paper that we can find to apply Machine Learning to the task of fishing and non-fishing detection. Here we apply a supervised machine learning method (CRFs) to detect longliner fishing activities.

III. LINEAR-CHAIN CONDITIONAL RANDOM FIELDS

A. Basic Principles of Linear-Chain Conditional Random Fields

Linear-chain conditional random fields are undirected graphical models that represent conditional probability distributions of random variables $V = X \cup Y$ that take the form

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{j=1}^n \psi_j(\mathbf{x}, \mathbf{y}), \quad (1)$$

where X is a set of observed variables, Y is a set of labels that we need to predict, and $Z(\mathbf{x})$ is a normalization or partition function

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \prod_{j=1}^n \psi_j(\mathbf{x}, \mathbf{y}'), \quad (2)$$

where $\psi_j(\mathbf{x}, \mathbf{y})$ are compatibility functions over a subset of random variables $A \subset V$, and n is the number of compatibility functions $\psi(\mathbf{x}, \mathbf{y})$ that factorize the probability distribution. Given compatibility functions in the form

$$\psi_j(\mathbf{x}, \mathbf{y}; \lambda) = \exp\left(\sum_{i=1}^m \lambda_i f_i(y_{j-1}, y_j, \mathbf{x}, j)\right), \quad (3)$$

the conditional probability distribution can be written as

$$p(\mathbf{y}|\mathbf{x}; \lambda) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{j=1}^t \sum_{i=1}^m \lambda_i f_i(y_{j-1}, y_j, \mathbf{x}, j)\right), \quad (4)$$

where m is the number of feature functions, t is the length of sequence \mathbf{y} , and λ is a set of weight parameters that help provide weighted average over these feature functions.

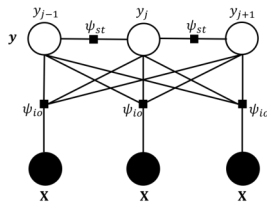


Fig. 1. Factor graph of linear-chain conditional random fields. \mathbf{x} denotes a sequence of input data, \mathbf{y} denotes a sequence of labels and ψ_{st} and ψ_{io} denote two compatibility functions.

In (3), the parameter λ_i of compatibility functions $\psi_j(\mathbf{x}, \mathbf{y}; \lambda)$ does not depend on the index j , which means the parameters are shared along the linear chain. Fig. 1 visualizes

the factor graph of linear-chain conditional random fields with two compatibility functions. In (4), the first sum runs over each position of the linear chain and the second sum runs over each feature function. The conditional probability $p(\mathbf{y}|\mathbf{x}; \lambda)$ can be represented as a mapping function from features to labels. Thus, the selection of feature functions is of great significance for the performance of a model. The parameters can be estimated using maximum-likelihood. The log-likelihood can be estimated with the Forward-Backward Algorithm. The inference of finding the most likely sequence \mathbf{y} given observations \mathbf{x} is performed using the Viterbi Algorithm [1]. The remainder of this section demonstrates how we adapt CRFs to identifying fishing activities.

B. Discretization

Compared with POS tagging where the input is a sequence of discrete words, AIS trajectory consists of real-valued features that are continuous by nature, such as longitudes and latitudes. Conditional random fields can model real-valued features, but they typically require proper normalization so that the value of the feature function is a linear function of the conditional probability $p(\mathbf{y}|\mathbf{x})$. However, since the relationships between AIS features and fishing activity labels are non-linear, we discretize the features to relax the normalization constraints and allow the conditional random fields to learn $p(\mathbf{y}|\mathbf{x})$ with a more flexible representation. We use a variant of equal interval binning discretization in our work. Each bin is associated with a set of parameters to fit the model.

C. Feature Functions

We use two sets of compatibility functions ψ_{st} and ψ_{io} to factor $p(\mathbf{y}|\mathbf{x})$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} (\psi_{st}(\mathbf{x}, \mathbf{y}) \cdot \psi_{io}(\mathbf{x}, \mathbf{y})). \quad (5)$$

The first compatibility function ψ_{st} is transition compatibility function, which models the transition probability of the labels from one state to another and takes the form

$$\psi_{st}(\mathbf{x}, \mathbf{y}; \lambda) = \exp(\lambda_{st} f_{st}(y_{j-1}, y_j, \mathbf{x}, j)), \quad (6)$$

where $f_{st}(y_{j-1}, y_j, \mathbf{x}, j)$ is transition feature function that takes the form

$$f_{st}(y_{j-1}, y_j, \mathbf{x}, j) = 1_{\{y_{j-1}=s\}} 1_{\{y_j=t\}}, \quad (7)$$

and st are all possible combinations of labels, y_{j-1} and y_j are the labels of the $(j-1)$ -th and j -th position of the linear chain and \mathbf{x} is the input sequence. More concretely, if the labels can only take two values: 0 and 1, the feature functions can be rewritten as

$$f_{00}(y_{j-1}, y_j, \mathbf{x}, j) = 1_{\{y_{j-1}=0\}} 1_{\{y_j=0\}}, \quad (8)$$

$$f_{01}(y_{j-1}, y_j, \mathbf{x}, j) = 1_{\{y_{j-1}=0\}} 1_{\{y_j=1\}}, \quad (9)$$

$$f_{10}(y_{j-1}, y_j, \mathbf{x}, j) = 1_{\{y_{j-1}=1\}} 1_{\{y_j=0\}}, \quad (10)$$

$$f_{11}(y_{j-1}, y_j, \mathbf{x}, j) = 1_{\{y_{j-1}=1\}} 1_{\{y_j=1\}}, \quad (11)$$

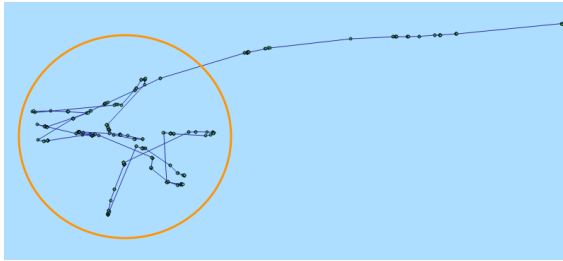


Fig. 2. Differences of fishing and non-fishing tracks. Tracks in the orange circle are fishing tracks and the rest are non-fishing tracks.

which are parameterized by $\lambda_{0,0}, \lambda_{0,1}, \lambda_{1,0}, \lambda_{1,1}$ respectively.

The second compatibility function $\psi_{\mathbf{i}o}$ is state-observation compatibility function, which models the probability distribution of the labels given a set of observations. The subscript \mathbf{i} within $\psi_{\mathbf{i}o}$ represents a set of input observations, which is different from the index i of parameter λ in (4). The state-observation compatibility function can be written as

$$\psi_{\mathbf{i}o}(\mathbf{x}, \mathbf{y}; \lambda) = \exp(\lambda_{\mathbf{i}o} f_{\mathbf{i}o}(y_{j-1}, y_j, \mathbf{x}, j)), \quad (12)$$

where $f_{\mathbf{i}o}(y_{j-1}, y_j, \mathbf{x}, j)$ is state-observation feature function that takes the form

$$f_{\mathbf{i}o}(y_{j-1}, y_j, \mathbf{x}, j) = \mathbb{1}_{\{y_j=o\}} \mathbb{1}_{\{\mathbf{x}=i\}}, \quad (13)$$

where $\mathbf{i}o$ are all possible combinations of input features and its corresponding labels. The state-observation feature functions are parameterized by a set of parameters $\lambda_{\mathbf{i}o}$.

IV. EXPERIMENTS

A. Data Preprocessing

AIS data contain attributes including MMSI, time, longitude, latitude, speed over ground (SOG) and course over ground (COG). In the experiments, we use historical AIS data from 14 longliners around the world collected from June 1st 2012 to Dec 31st 2013. The fishing and non-fishing activities of these data are labeled by a marine biology expert. Fig. 2 shows the differences between fishing and non-fishing tracks that are recovered from discrete AIS signals. The fishing tracks are in the form of a zigzag. The non-fishing tracks tend to follow smooth lines. To preprocess the data, we perform data cleansing to remove uninformative data, data conversion from absolute values to differential values (i.e. difference with to the previous point), data discretization to transform continuous value into nominal counterparts, and feature selection to fit the model with the most relevant features.

1) *Data Cleansing*: We sort the data points of each longliner in chronological order. We then remove repetitive data points as well as data points with incomplete features. We further detect and remove outliers if the speed exceeds the normal range and the location deviates from its normal trajectories. After data cleansing, we have 505893 longliners data points in total. On average, 77% of the data points are labeled as fishing. Table I shows a summary of the 14 longliners after cleansing.

TABLE I
SUMMARY OF THE 14 VESSELS DATA

Track ID	Track Size	# of Fish Points	% of Fish Activity
1	21556	17148	79.6
2	8829	6326	71.7
3	30166	24422	81.0
4	6086	4226	69.4
5	28184	22153	78.6
6	37710	32977	87.4
7	24715	16857	68.2
8	2032	1755	86.4
9	12111	8470	69.9
10	17161	14277	83.2
11	2670	1761	66.0
12	90429	78765	87.1
13	108826	73005	67.1
14	115418	86256	74.7
Mean	36135	27742	76.5

2) *Differential longitude and latitude*: In our early experiments, we trained our predictive models using absolute value of longitudes and latitudes. However, we find the resulting models overfitting on the training data and cannot generalize to different locations. In order to generalize the model into other areas, we use differential longitude, that is the difference of absolute longitudes between the current and previous data points. Thus the absolute longitudes $L = [l_1, l_2, \dots, l_n]$ are transformed to differential longitudes that takes the form $L_d = [l_2 - l_1, l_3 - l_2, \dots, l_n - l_{n-1}]$. Similarly, we transform absolute latitudes to its differential form.

3) *Discretization*: As mentioned in Section 3, we discretize the attributes of data using a variant of equal interval binning. Early experiments indicate different features require different sizes of intervals. For differential longitudes and latitudes, the interval size m takes the form

$$m = \begin{cases} 0.05 & l \in [-1, 1] \\ 20 & l > 1, l < -1 \end{cases} \quad (14)$$

where fine intervals are selected in the range $[-1, 1]$. Because most differential longitudes and latitudes are between -1 and 1, fine intervals provide a larger number of parameters to fit the model compared to coarse intervals, when l is greater than 1 or less than -1. For SOG, we set m to 0.5. For COG, we set m to 20.

4) *Feature Selection*: In early experiments, we built the model using different combinations of features, such as differential longitude, differential latitude, SOG and COG. We find the model built with differential longitude, differential latitude, and SOG performed the best. We use these features in the following three experiments of this paper, shown in Fig. 3. More precisely, we use the following feature functions in our experiments: 1) pairs of longitudes and latitudes are selected to represent the positions of data points (colored in orange); 2) pairs of neighbouring longitudes are selected to represent the changes of longitudes over time (colored in red); 3) pairs of neighbouring latitudes are selected to represent the changes of latitudes over time (colored in green); 4) speed and pairs of neighbouring speed are selected to represent speed information

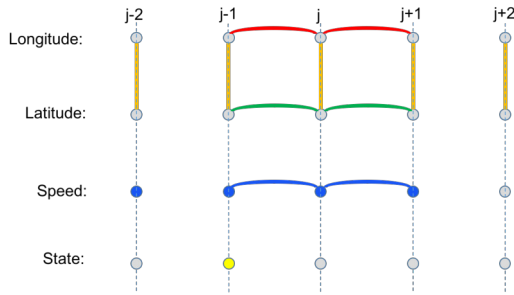


Fig. 3. Feature functions. When predicting the label of index j , the values of colored dots as well as the paired values connected by solid lines are selected as feature functions.

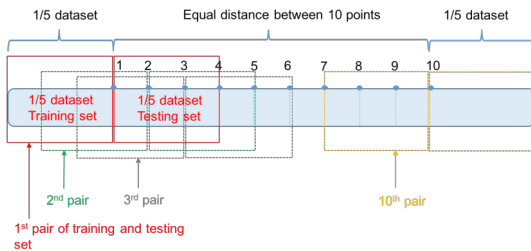


Fig. 4. Modified Monte Carlo. Choose 10 points as split point, obtain 10 pairs of training and testing sets.

(colored in blue); 5) label of the previous state is selected to help model the transition probability between states (colored in yellow). These feature functions employ spatial and temporal information to aid the classification task.

B. Results

We train CRF models using CRF++ [12]. We design three experiments to evaluate the model: Modified Monte Carlo methods, Iterative Leave One Batch Out (LOBO) and Stratified LOBO. We then compare CRFs with autoencoders and a data mining approach.

1) *Modified Monte Carlo*: Here, we concatenate all the data and apply Monte Carlo methods to get 10 pairs of training and testing data sets. The size of the training and testing set is $\frac{1}{5}$ of the total number of data points. To find the dividing points

of the ten Monte Carlo experiments, we first set aside $\frac{1}{5}$ of the whole data set in the front and back respectively, then select 10 dividing points with identical intervals that can cover the entire dataset. For each selected point, the $\frac{1}{5}$ portion of the data set to the left of the point and the $\frac{1}{5}$ portion to the right constitute one pair of training and testing set. Fig. 4 shows the way we obtain our 10 pairs of training and testing sets.

For each Monte Carlo experiment, the model is evaluated using accuracy, sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV), as shown in Table II, where the positive class is non-fishing. These measurements provide us with information about the overall performance of the model. Also, we present the mean and standard deviation (SD) of the metrics.

2) *Iterative Leave One Batch Out*: Here, we first split the 14 vessels into 2 groups, group one with 10 vessels for iterative LOBO, and group two with four vessels for stratified LOBO. To split the vessels into two groups, we first take four vessels from the 14 longliners as group two and then take the rest 10 vessels as group one. To get an accurate evaluation of the model, we need the four selected vessels to be representative of the entire data set. Therefore, we use trajectory size as a criterion to help select these vessels. Since the trajectory sizes of the 14 vessels vary from thousands to hundred thousands, as shown in Table I, we categorize the 14 trajectories into three sets based on their sizes. For the three sets, the sizes of trajectories are in the range of thousands, ten thousands and hundred thousands respectively. We then randomly select vessels that are proportional to the cardinality of each set, one vessel in set one, two vessels in set two, and one vessel in group set as the four testing vessels.

For the set of 10 vessels, in each iteration, we consider one vessel as one batch to be the test vessel and build one model on the rest of 9 vessels, and repeat this for 10 times. The results of Iterative Leave One Batch Out for the 10 test vessels are shown in Table III.

3) *Stratified Leave One Batch Out*: We use the selected four vessels from experiment 2 as independent test vessels and we train the model using the 10 vessels from experiment 2, and evaluate on the rest four vessels individually. The performance of the model on the four testing vessels are shown in Table IV. We further visualize the classification results in Fig. 5.

4) *Comparisons with autoencoders and Data Minging approach*: We reproduce the autoencoders [11] and the data mining approach [10] on the same set of data in previous ILOBO experiment. We compare the performance of CRFs with these two methods as shown in Table III. We perform paired-samples t-test to compare the classification accuracies of CRFs with autoencoders and the data mining approach. The resulting p-value from the comparison of CRFs and autoencoders is 0.057. For the comparison of CRFs and the data mining approach, the p-value is 0.032 which is less than significance level 0.05 so that we can conclude these two methods are significantly different.

TABLE II
EVALUATION USING MODIFIED MONTE CARLO METHODS .

Expt ID	Accuracy	Sensitivity	Specificity	PPV	NPV
1	0.868	0.481	0.959	0.730	0.888
2	0.872	0.460	0.947	0.610	0.906
3	0.959	0.826	0.985	0.916	0.966
4	0.973	0.876	0.991	0.953	0.976
5	0.860	0.622	0.941	0.785	0.879
6	0.765	0.821	0.739	0.598	0.897
7	0.851	0.743	0.908	0.809	0.871
8	0.944	0.885	0.965	0.900	0.960
9	0.939	0.735	0.981	0.892	0.947
10	0.946	0.835	0.983	0.940	0.947
Mean	0.898	0.728	0.940	0.813	0.924
SD	0.065	0.157	0.075	0.131	0.040

TABLE III
EVALUATION USING ITERATIVE LEAVE ONE BATCH OUT.

ID	Accuracy			Sensitivity			Specificity			PPV			NPV		
	CRF	AE	DM	CRF	AE	DM	CRF	AE	DM	CRF	AE	DM	CRF	AE	DM
1	0.86	0.83	0.65	0.74	0.60	0.45	0.90	0.93	0.91	0.75	0.77	0.85	0.90	0.85	0.57
2	0.85	0.85	0.89	0.53	0.39	0.71	0.93	0.96	0.93	0.63	0.76	0.70	0.89	0.86	0.93
3	0.83	0.86	0.46	0.92	0.58	0.35	0.78	0.94	0.93	0.65	0.76	0.95	0.96	0.88	0.25
4	0.86	0.86	0.87	0.50	0.51	0.66	0.96	0.95	0.95	0.75	0.74	0.82	0.88	0.88	0.89
5	0.96	0.86	0.89	0.74	0.44	0.54	0.99	0.94	0.98	0.88	0.57	0.83	0.96	0.90	0.90
6	0.82	0.76	0.76	0.52	0.34	0.59	0.96	0.94	0.89	0.87	0.68	0.81	0.81	0.776	0.73
7	0.92	0.80	0.54	0.53	0.00	0.20	0.98	0.93	0.94	0.80	0.00	0.81	0.93	0.85	0.5
8	0.90	0.84	0.86	0.44	0.38	0.55	0.99	0.94	0.96	0.88	0.55	0.80	0.90	0.88	0.80
9	0.83	0.86	0.88	0.55	0.62	0.89	0.97	0.94	0.86	0.92	0.76	0.93	0.81	0.89	0.80
10	0.91	0.85	0.74	0.72	0.56	0.49	0.98	0.94	0.98	0.91	0.76	0.96	0.91	0.87	0.66
Mean	0.87	0.84	0.75	0.62	0.44	0.54	0.94	0.94	0.93	0.80	0.64	0.85	0.89	0.86	0.71
SD	0.05	0.03	0.16	0.15	0.19	0.19	0.06	0.01	0.04	0.10	0.24	0.08	0.05	0.04	0.22

TABLE IV
EVALUATION USING STRATIFIED LEAVE ONE BATCH OUT.

Expt ID	Accuracy	Sensitivity	Specificity	PPV	NPV
1	0.871	0.642	0.929	0.700	0.910
2	0.818	0.576	0.922	0.759	0.835
3	0.991	0.944	0.998	0.988	0.992
4	0.888	0.824	0.919	0.833	0.914
Mean	0.892	0.747	0.942	0.820	0.913
SD	0.072	0.168	0.038	0.125	0.064

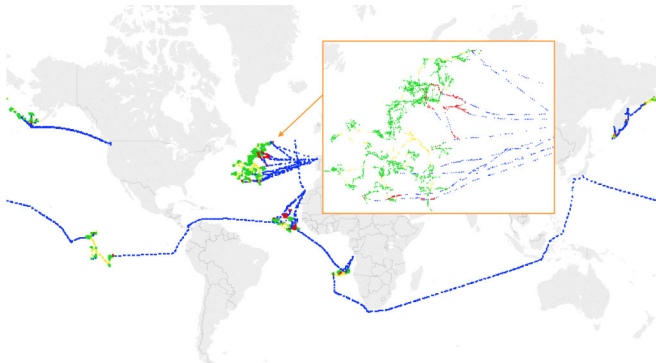


Fig. 5. The Visualization of four independent testing vessel tracks. Green points mean both the label and the prediction are fishing. Blue points mean both the label and the prediction are non-fishing. Red points mean the label is fishing while the prediction is non-fishing. Yellow points mean the label is non-fishing while the prediction is fishing. Zoomed-in region gives details of classification results in the Atlantic ocean area.

C. Discussion

The results of the three experiments are consistent. The average accuracy of the three experiments is 88.7% with 6.1% average standard deviation. By comparing the results of the second and the last experiments, we found that the models built using Iterative Leave One Batch Out perform as good as the model built in Stratified Leave One Batch Out. This further proves the stability and potential of the model in future fishing activity detection.

Through comparisons, we find CRFs can have better classification accuracy, in terms of mean and standard deviation. We also find CRFs do not suffer from imbalanced data as autoencoders: in experiment 7, autoencoders label all data

points as positive class with 0% of sensitivity, whereas CRFs have 53.4% of sensitivity. According to our t-test results, despite the fact that the CRFs and autoencoders are not systematically different on significance level 0.05, CRFs can perform as well as and sometimes better than autoencoders and the data mining approach.

V. CONCLUSIONS AND FUTURE WORK

This paper presents an approach to detecting fishing activities from historical AIS data using Conditional Random Fields. Data cleaning, discretization and transformation followed by feature selection are performed to preprocess data. We then specify proper feature functions to train CRF models. The resulting models are further evaluated in three different ways. In terms of efficiency, we find CRFs can be trained efficiently than complex models such as deep learning. In terms of effectiveness, the three evaluation experiments suggest the model can generalize well in future fishing activity detection problems.

As for future work, we will investigate better ways of developing additional features, such as density and angle. We also consider systematic approaches to incorporate additional density and angle information into feature functions to aid the development of the model.

ACKNOWLEDGMENT

The authors have been supported by the Natural Sciences and Engineering Research Council of Canada. Last author's research is also supported in part by the National Research Centre of Poland (NCN) grant DEC-2013/09/B/-ST6/01549.

REFERENCES

- [1] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Machine Learning*, vol. 4, no. 4, pp. 267–373, 2011.
- [2] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the eighteenth international conference on machine learning, ICML*, vol. 1, 2001, pp. 282–289.
- [3] A. PVS and G. Karthik, "Part-of-speech tagging and chunking using conditional random fields and transformation based learning," *Shallow Parsing for South Asian Languages*, vol. 21, 2007.

- [4] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 134–141.
- [5] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 188–191.
- [6] B. Settles, "Biomedical named entity recognition using conditional random fields and rich feature sets," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*. Association for Computational Linguistics, 2004, pp. 104–107.
- [7] L. Liao, D. Fox, and H. Kautz, "Extracting places and activities from gps traces using hierarchical conditional random fields," *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 119–134, 2007.
- [8] F. Mazzeella, M. Vespe, D. Damalas, and G. Osio, "Discovering vessel activities at sea using ais data: mapping of fishing footprints," in *Information Fusion (FUSION), 2014 17th International Conference on*. IEEE, 2014, pp. 1–7.
- [9] D. Peel, N. M. Good, and T. Quinn II, "A hidden markov model approach for determining vessel activity from vessel monitoring system data," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 68, no. 7, pp. 1252–1264, 2011.
- [10] E. N. de Souza, K. Boerder, S. Matwin, and B. Worm, "Improving fishing pattern detection from satellite ais using data mining and machine learning," *PLOS ONE*, vol. 11, no. 7, p. e0158248, 2016.
- [11] X. Jiang, D. L. Silver, B. Hu, E. N. de Souza, and S. Matwin, "Fishing activity detection from ais data using autoencoders," in *Canadian Conference on Artificial Intelligence*. Springer, 2016, pp. 33–39.
- [12] CRF++: Yet Another Toolkit . [Online]. Available: <https://taku910.github.io/crfpp/>