Event Classification in Foreign Language Aviation Reports

Anil Yelundur, Chris Giannella, Karine Megerdoomian, Craig Pfeifer
The MITRE Corporation
7515 Colshire Dr., McLean VA 22102 USA
{ayelundur,cgiannella,karine,cpfeifer}@mitre.org

Abstract: When adverse aviation events occur, narrative reports describing the events and their associated flights provide a valuable record for improving safety. Manual examination of large collections of such reports is challenging. Tools for automated event classification can help to mitigate this challenge. Event classification involves the assignment of type labels to individual reports indicating all the types of events that are described in the report. While several studies have developed and systematically empirically evaluated event classification tools on *English* aviation narratives, we are not aware of any that have done the same on foreign language narratives. We developed and implemented an approach for event classification based on Bayesian logistic regression and a novel feature selection technique. For comparison purposes, we also implemented an approach described in the literature. We collected and annotated a corpus of Japanese aviation incident reports, as well as, a corpus of French incident reports. We carried out a series of experiments comparing the accuracy of our approach and the other approach. On the Japanese dataset, our approach exhibited greater accuracy for all event types considered. On the French dataset, our approach exhibited greater accuracy for two of four event types and worse for the other two.

1. Introduction

Safety is of paramount importance in the commercial airline industry. When adverse events occur during a flight, narrative reports describing the events and their associated flights provide a valuable record for improving safety. By examining large collections of these reports, analysts can better understand the causes of the events. For example, analysts can characterize the factors that contribute to specific types of events, e.g. factors contributing to loss of control of the aircraft. However, the large number of reports motivates the need for automated text analysis capabilities to assist the analyst. One such capability is automated event classification – the assignment of type labels to individual reports indicating all the types of events that are described in the report. This capability can help an analyst to focus attention on a subset of reports relevant to the events of interest [1]. While several studies have developed event classification tools and carried out a detailed and systematic empirical evaluation of them on *English* aviation narratives, we are not aware of any that have done the same on foreign language narratives. Since many airlines and governments generate foreign language aviation narratives, the development and empirical evaluation of event classification tools on foreign language narratives is warranted.

1.1 The Event Types

We developed the two-level event type hierarchy depicted in Figure 1 based on the main top categories proposed by the International Civil Aviation Organization / Commercial Aviation Safety Team (ICAO/CAST)[1]. Due to the modest sizes of our datasets, we use the five top level types, hereafter referred to as "event types". A detailed discussion of the event type hierarchy can be found in the Appendix.

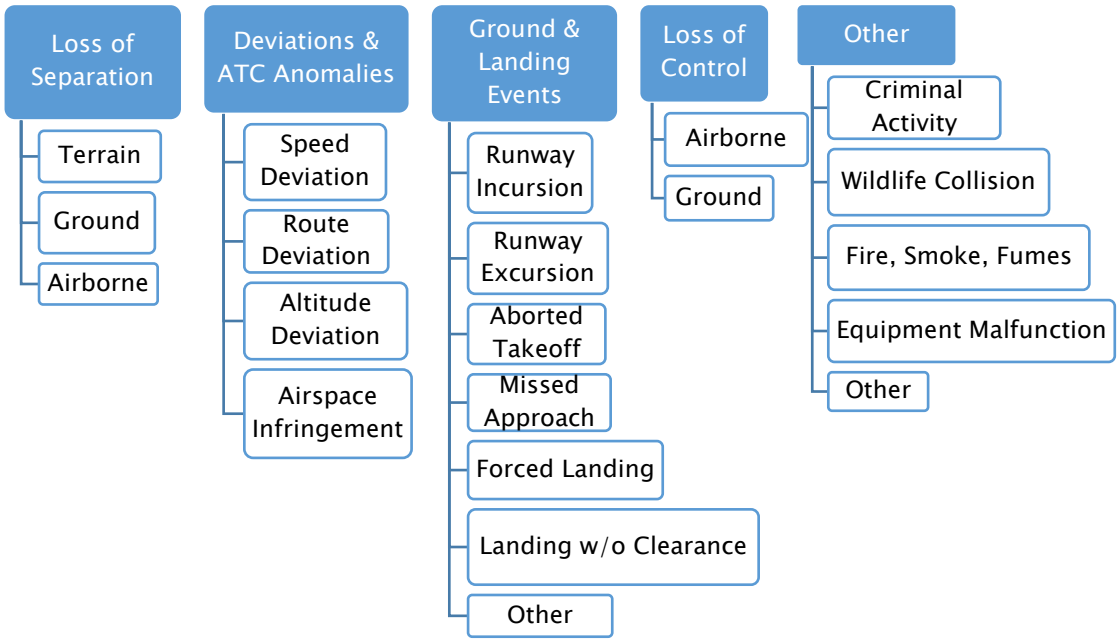

FIGURE 1: Event Type Hierarchy

1.2 The Problem, Our Approach and Contributions

Let Φ denote a subset of the five event types and $\{(D1,T1), \ldots,(Dn,Tn)\}$ denote a training dataset. Di is a foreign language aviation report and Ti is a subset of Φ such that, for each event type t in Ti, the narrative in Di implies that an event of type t occurred. The problem is to build a classifier that maps new reports D to sets of event types $T \subseteq \Phi$ such that for each event type t in T, the narrative in D implies that an event of type t occurred. We refer to this as the *aviation event multi-label classification* problem – "multi-label" because reports can be mapped to zero, one, or more event types.

The aviation event multi-label classification problem is an example of a multi-label learning problem, a topic that has been addressed in the Machine Learning literature. One simple approach to multi-label learning, called one-versus-all, is to build one binary classifier for each event type, then given a new report, assign each type whose classifier returns true. This approach is referred to as a first-order strategy since the assignment of the types are made

[1] http://www.icao.int/APAC/Meetings/2012_APRAST1/OccurrenceCategoryDefinitions.pdf

independently. Other kinds of first-order strategies, as well as, higher-order strategies have been developed. The reader is referred to [2] for an extensive survey of multi-label learning approaches.

We employ a simple one-versus-all approach to address the aviation event multi-label classification problem. We build a multi-label classifier from the training data *{(D1,T1),…,(Dn,Tn)}* as follows. For each event type *t* that appears in the training data:

1. A binary training dataset *{(D1,[t±]1), …,(Dn, [t±]n)}* is created where *[t±]i* is *+1* if *t* is in *Ti*, else *[t±]i* is *−1*, i.e., "*+1*" denotes the positive label indicating that *t* is in *Ti*.
2. From the binary training dataset, a binary report classifier is trained – see Section 2 for details. Let *Ct* denote the classifier.

A new report, *D*, is mapped to the following set of types: {*t* in **Φ**: *t* appears in the training dataset and *Ct[D] = +1*} where *Ct[D]* denotes the label assigned to *D* by *Ct*.

We develop a novel Bayesian logistic regression algorithm for binary text classification. The primary novelty of our approach is the feature selection procedure designed to mitigate the deleterious effects of imbalanced training data. We utilize this binary text classification algorithm, in one-versus-all fashion, to address the aviation event multi-label classification problem. We collect and annotate Japanese and French aviation reports and perform experiments to evaluate the accuracy of our approach as compared with an approach described in [3]. Ours is the first study in which an aviation event multi-label classifier is developed and systematically empirically evaluated on foreign language aviation incident reports.

2. Binary Report Classification

Here we describe our approach to building a classifier from a binary training dataset *{(D1,[t±]1), …,(Dn,[t±]n)}* – an extensively studied problem [4].

Let *V* denote a list of all unique terms (token unigrams and bigrams) that appear in any of the reports *D1,…,Dn*. *V* is the universe of all possible features. Each report *Di* is mapped to a Boolean vector, *Xi*, of length */V/* such that *Xi[k] = 1* if the k^{th} term in *V* appears in *Di*, else *Xi[k] = 0*. In effect, *Di* is represented as a bag of terms.

The first subsection below describes a version of Bayesian logistic regression that is the corner-stone of our classification approach. The next subsection describes the feature selection procedure we employ. The final subsection describes our entire classification approach – both training and application.

2.1 Adaptive, Laplacian Prior, Bayesian Logistic Regression

The standard logistic regression model with */V/* Boolean dependent variables maps a label *y* in {*+1,−1*} and a length */V/* Boolean vector, *Z*, to a probability

$$P_{\beta,b}(y|Z) \stackrel{\text{def}}{=} \frac{1}{1+\exp\left(-yb - y\beta^T Z\right)}$$

where *b* denotes the bias coefficient and β denotes the term coefficients. A two-level Bayesian approach is employed to set coefficient *b* and set coefficient vector β from a training dataset with *N* labeled vectors: *{(Z1,y1),…,(ZN,yN)}* with each *yi* either *+1* or *-1*. The aim is to maximize the posterior distribution with the following prior on the coefficients and the following hyper-prior on the hyper-parameter λ:

$$L\left[0,2\lambda^2\right](b)\prod_{k=1}^{|V|} L\left[0,2\,\lambda^2/wk^2\right](\beta k) \text{ and } G[0.001,0.001]\left(\lambda^2\right)$$

where *L[0,2a²](x)* denotes the value at *x* of the Laplace distribution with mean zero and variance a^2 [5]; βk denotes the k^{th} component of β; *wk* denotes a fixed weight; and, *G[0.001,0.001](x)* denotes the value at *x* of the Gamma distribution with shape and rate parameters 0.001. The weight parameters will be instrumental in the feature selection procedure (described later), but are treated as constants here. The weights allow shrinkage effects to be individually adjusted for the coefficients in β – hence the use of the term "adaptive". Details regarding how the weights are fixed can be found at the end of this subsection. The aim can be equivalently stated as jointly maximizing the following expression for *b*, β and λ.

$$\sum_{i=1}^{N}\log_{10}\left(P_{\beta,b}(yi|Zi)\right)+\log_{10}\left(L[0,2\lambda^2](b)\right)+\sum_{k=1}^{|V|}\log_{10}\left(L[0,2\,\lambda^2/wk^2](\beta k)\right)+\log_{10}\left(G[0.001,0.001](\lambda^2)\right).$$

To avoid the Gamma hyper-prior overwhelming the log-likelihood (the first summation), λ is bounded below by a fixed constant $\alpha > 0$. The coefficient *b* and vector of coefficients β are set by solving:

$$\max_{b,\beta,\lambda}\left\{\sum_{i=1}^{N}\log_{10}\left(P_{\beta,b}(yi|Zi)\right)+\log_{10}\left(L[0,2\lambda^2](b)\right)+\sum_{k=1}^{|V|}\log_{10}\left(L[0,2\,\lambda^2/wk^2](\beta k)\right)+\log_{10}\left(G[0.001,0.001](\lambda^2)\right)\right\}$$

$$\text{subject to: } \lambda > \alpha.$$

This constrained optimization problem is solved using the R function *nloptr_ld_mma* which implements a variant of the Method of Moving Asymptotes [6].

Fixing the weights: As argued in [7], the use of the Laplacian prior promotes feature selection. The coefficients on irrelevant variables tend to be driven to zero during optimization (coefficient shrinkage). The weights enhance this effect – making *wk* smaller tends to dampen the shrinkage on βk. By default, the weights are set to one. Some variables have their weights heuristically reduced in order to dampen shrinkage on those variables' coefficients. These are the variables whose entry is one only in positively labeled vectors or is one only in negatively labeled vectors. The weight decrease for each of these variables depends on the number of vectors whose entry is one. Precisely stated, the weight *wk* assigned to the k^{th} variable is:

$$c_+(k) \stackrel{def}{=} |\{Zi: 1 \le i \le n \text{ and } yi = +1 \text{ and } Zi[k] = 1\}|$$

$$c_-(k) \stackrel{def}{=} |\{Zi: 1 \le i \le n \text{ and } yi = -1 \text{ and } Zi[k] = 1\}|$$

$$wk \stackrel{def}{=} \begin{cases} 1/\min\{\log_{10}[c_+(k)+9], \log_{10} 400\} & \text{if } c_+(k)>0 \text{ and } c_-(k)=0 \\ 1/\min\{\log_{10}[c_-(k)+9], \log_{10} 400\} & \text{if } c_-(k)>0 \text{ and } c_+(k)=0 \\ 1 & \text{otherwise.} \end{cases}$$

2.2 Feature Selection

Since aviation adverse events are uncommon, the training dataset is typically imbalanced: it has more negatively labeled vectors than positively labeled ones. Training a classifier from an imbalanced dataset can be problematic. One such problem stems from discriminative terms that are prevalent in the negatively labeled vectors close to the decision boundary. These terms are important in classifying the "difficult" negatively labeled vectors. However, the discriminative effect of these terms might be dampened by the larger number of discriminative terms that are only prevalent in the negatively labeled vectors not close to the decision boundary. To avoid dampening the effect of the negative discriminative terms close to the boundary, a subset of negatively labeled vectors are selected – those deemed close to the boundary. These vectors, along with all of the positively labeled ones are used to select features through coefficient shrinkage.

Example: Suppose the type *t* is runway excursion and the term "runway excursion" is positively significant – the term occurs more in the positively labeled vectors than the negatively labeled ones. Further suppose a negatively labeled report contains the phrase "a runway excursion did not occur". Selecting the vector corresponding to this report can help the training process to key in on important negatively discriminating terms such as "not occur" which helps to produce a classifier that will avoid mistakenly labeling this vector as positive despite the presence of the term "runway excursion".

The feature selection algorithm: Given the bag-of-terms training dataset for type *t*, $\{(X1,[t\pm]1),\ldots,(Xn,[t\pm]n)\}$, a subset of terms Ω_t is selected from *V*. First, the weights are computed as described at the end of the previous subsection but with $\{(Z1,y1),\ldots,(ZN,yN)\} = \{(X1,[t\pm]1),\ldots,(Xn,[t\pm]n)\}$.

Next, a subset of negatively labeled vectors is selected. Namely, those negatively labeled vectors *Xi* such that: there exists $1 \le k \le |V|$ such that $Xi[k]=1$ and $c_+(k) > c_-(k)$. Let $\{(Xi_1,[t\pm]i_1),\ldots,(Xi_s,[t\pm]i_s)\}$ denote all of the positively labeled vectors and the selected negatively labeled ones (*s* vectors in total).

Next, the training dataset, $\{(X1,[t\pm]1),\ldots,(Xn,[t\pm]n)\}$, is modified to remove all terms that are in a selected negatively labeled vector and are not positively significant with respect to $\{(Xi_1,[t\pm]i_1),\ldots,(Xi_s,[t\pm]i_s)\}$. Specifically, for each $1 \le k \le |V|$, if

$$\hat{c}_-(k) > 0 \text{ and } c_+(k) \le \hat{c}_-(k), \text{ where:}$$

$$\hat{c}_{-}(k) \stackrel{\text{def}}{=} \left|\{Xi_j : 1 \le j \le s \text{ and } [t\pm]i_j = -1 \text{ and } Xi_j[k] = 1\}\right|,$$

then *Xi[k]* is set to *0* for all $1 \le i \le n$.

Next, the term coefficients β are set using the procedure from Subsection 2.1 with $\{(Z1,y1),\ldots,(ZN,yN)\} = \{(Xi_1,[t\pm]i_1),\ldots,(Xi_s,[t\pm]i_s)\}$ and α fixed as:

$$\alpha \stackrel{\text{def}}{=} \begin{cases} \sqrt{\max\{1/\log_{10}|V|,\ 1/\log_{10} s\}} & \text{if } \left|\{i_j : 1 \le j \le s,\ yi_j = +1\}\right| < \left|\{i_j : 1 \le j \le s,\ yi_j = -1\}\right| \text{ and } |V|,s > 10 \\ 1 & \text{otherwise.} \end{cases}$$

Finally, for each $1 \le k \le |V|$, the k^{th} term in V is added to Ω_t if $|\beta k| > 0.0001$. Intuitively, Ω_t contains only those terms whose coefficient is sufficiently far away from zero.

2.3 Entire Classification Algorithm

Application: Given β and b, a report D is classified with respect to type t as follows.

1. D is mapped to X, a length $|V|$ Boolean vector as described earlier.[2]
2. If $P_{\beta,b}(+1|X) > 0.5$, then D is assigned label $+1$, else -1.

Training: To train a binary report classifier from the binary training dataset for type t $\{(D1,[t\pm]1),\ldots,(Dn,[t\pm]n)\}$, the bias coefficient b and term coefficients β must be set. The following algorithm is carried out to do so. Steps 1–3 were discussed in the previous two subsections. Step 4, a bias adjustment step, takes affect when the number of negatively labeled reports is larger than the number of positively labeled ones. In this case, the separating boundary tends to be pushed towards the positive set resulting in reduced recall. The bias adjustment aims to improve recall by sacrificing a small amount of precision.

1. The list of unique terms V is created and each Di is mapped to a length $|V|$ Boolean vector Xi as described at the beginning of Section 2. The result is a bag-of-terms training dataset for type t, $\{(X1,[t\pm]1),\ldots,(Xn,[t\pm]n)\}$.
2. A subset of terms Ω_t is selected from V as described in Subsection 2.2 and all the other terms are removed from the training dataset. Specifically, for all $1 \le k \le |V|$, if the k^{th} term in V does not appear in Ω_t, then $Xi[k]$ is set to 0 for all $1 \le i \le n$.
3. The bias coefficient b and $|V|$ term coefficients β are set as described in Subsection 2.2 with $\alpha = 1$ and $\{(Z1,y1),\ldots,(ZN,yN)\} = \{(X1,[t\pm]1),\ldots,(Xn,[t\pm]n)\}$. If the k^{th} term in V does not appear in Ω_t, then βk will be set to zero during optimization.
4. If $ratio = |\{i : 1 \le i \le n,\ [t\pm]i = -1\}| / |\{i : 1 \le i \le n,\ [t\pm]i = +1\}| > 1$, then the bias coefficient b is adjusted as follows.
    a. Set $\hat{b}$ to 0.
    b. Carry out $\lceil 100 \log_{10} ratio \rceil + 1$ iterations:

[2] Terms in D that do not appear in V are ignored.

1st) Set *TP*, *FP*, *FN* to 0. For $i=1$ to n: if $P_{\beta,(b+\hat{b})}(+1/X_i) > 0.5$ and $[t\pm]i=+1$, increment *TP*; if $P_{\beta,(b+\hat{b})}(+1/X_i) > 0.5$ and $[t\pm]i=-1$, increment *FP*; if $P_{\beta,(b+\hat{b})}(+1/X_i) \leq 0.5$ and $[t\pm]i=+1$, increment *FN*. Compute the *F* score.

2nd) Set $\hat{b}$ to $min\{\hat{b}+0.05, log_{10} ratio\}$.

c. Add to b the $\hat{b}$ that produced the highest *F* score.

3. Experiments

3.1 Data

We downloaded Japanese and French aviation reports from two sources.

- The Japanese Transport Safety Board (JTSB), aircraft accident and incident reports.[3]
- The French Office of Investigations and Analysis for Civil Aviation Safety.[4]

The reports were all pdf files and we were able to extract the text from only a subset of them. The Japanese reports each had an accompanying English translation (manually produced by the JTSB). Using these translations, a MITRE colleague (Vanesa Jurica) annotated all of the Japanese reports using the five event types (recall a report can be assigned more than one event type). One of the authors (Megerdoomian) reads French and annotated a subset of the French reports.

For the purposes of classification we only used part of the reports. From the Japanese reports, we only used the text in the "Summary of Serious Incident" section for serious incident reports, or only the text in the "History of the Flight" section for all other reports. From the French reports, we only used the text in the "Circonstances" Section.

In all reports, we replaced all digits (0-9) with the character d. For the Japanese reports, we used the tokenizer provided by Lucene 5.0.0 analyzers-kuromoji (http://lucene.apache.org/core/5_0_0/analyzers-kuromoji/index.html). For the French reports, we used the tokenizer provided in the Stanford University core NLP open source library version 3.3.1 (http://nlp.stanford.edu/software/corenlp.shtml), we lowercased all tokens, and dropped all tokens that appeared in a stop-word[5] list. Figure 2 provides statistics regarding the datasets.

	JAPANESE	FRENCH
Total number of reports	110	206
Avg. (stan. dev.) report size in tokens	262 (292)	437 (204)
Number of reports with 1 assigned types	91	121
Number of reports with 2 assigned types	19	67
Number of reports with 3 assigned types	0	15
Number of reports with 4 assigned types	0	3

[3] http://www.mlit.go.jp/jtsb/airrep.html
[4] http://www.bea.aero/en/index.php
[5] The stop-word list contained 463 words.

Number of reports with assigned type		
Loss of Separation	24	46
Deviation & ATC Anomalies	3	13
Ground & Landing Events	61	164
Loss of Control	4	45
Other	37	44

FIGURE 2: Japanese & French Datasets

3.2 Baseline Approach

For comparison purposes, we implemented an approach very similar to the inductive Support Vector Machine (SVM) approach described in Sections 5.2.2 and 6.4.1 of [3]. In one-versus-all fashion, this approach builds a binary report classifier for each event type t. Given a binary training dataset $\{(D1,[t\pm]1), \ldots,(Dn, [t\pm]n)\}$, the universe of all terms, V, is computed. As discussed in Section 2, each report Di is mapped to a Boolean vector, Xi, of length $|V|$ resulting in a bag-of-terms training dataset $\{(X1,[t\pm]1),\ldots,(Xn,[t\pm]n)\}$.

1. The bag-of-terms training dataset is randomly split into pure training and development parts.
    a. 70 percent of the positively labeled vectors and 70 percent of the negatively labeled ones are selected and put into the pure training part.
    b. The remainder of the positively and negatively labeled vectors are put into the development part.
2. Four parameters are chosen based on the pure training and development split: C (controlling the SVM error vs. margin trade-off), γ (controlling the radial basis function kernel), PT (the percentage of terms selected), and θ (the classification probability threshold).
    a. For each pair (C, γ) in $\{2^{-5},2^{-4},\ldots,2^{10},2^{11}\}x\{2^{-5},2^{-4},\ldots,2^{10},2^{11}\}$, an SVM model is built from the pure training part using all of the terms; the F score of this model (using classification probability threshold 0.5) is computed[6] over the development part. The pair producing the highest F score is chosen (with ties broken by smaller parameters). Let C_0 and γ_0 denote the chosen parameters.
    b. For each pair (PT,θ) in {10,20..,100}x{0.05,0.1,…,0.95}, an SVM model is built from the pure training part using (C_0,γ_0) and the top PT percent of the terms with respect to their information gain; the F score of this model (using classification threshold θ) is computed over the development part. The pair producing the highest F score is chosen (with ties broken by larger parameters). Let PT_0 and θ_0 denote the chosen parameters.

[6] The SVM model is applied to each vector in the development part producing a probability for the label +1. If this probability is greater than the classification probability threshold, then the vector is classified as +1, else -1.

3. An SVM model is built from the training dataset using (C_0, γ_0) and using the top PT_0 percent of the terms with respect to information gain. When using this model to classify new vectors, θ_0 is used as the classification probability threshold.

In all cases, the Sequential Minimal Optimization (SMO) algorithm [8] is used to build SVM models. We used the SMO class in WEKA 3.6.10 [9] to build SVM models and to produce positive label probabilities when applying the models to reports.

The SVM approach above to building binary report classifiers is similar, but not identical to the approach in [3]. We chose the optimal setting of the SVM model parameters, C and γ, by dividing the training data as pure train/development sets instead of applying cross-validation. Besides Ul Abedin, Ng, & Khan, 2010 do not discuss the ranges over which C and γ were varied. We classified new vectors by setting a positive threshold on the classification probabilities instead of setting a threshold on the raw scores (distance from the hyper-plane) from the SVM model.

3.3 Methodology

For each dataset (Japanese and French), we applied leave one out cross-fold validation on each event type, t, that occurred in at least 30 reports. Let Δ denote the full binary dataset for type t. Let Di and $[t\pm]i$ denote the i^{th} report and binary label; let $\Delta[-i]$ denote Δ with (Di, $[t\pm]i$) removed.

1. Set *TP[BLR]*, *FP[BLR]*, *FN[BLR]*, *TP[SVM]*, *FP[SVM]*, and *FN[SVM]* all to 0.
2. For $i=1$ to n, do
    a. A binary classifier is built from $\Delta[-i]$ and applied to Di using the procedures described in Section 2 (our Bayesian Logistic Regression approach).
        i. If the classifier predicts label $+1$ and $[t\pm]i$ is $+1$, then *TP[BLR]* is incremented. If the classifier predicts label $+1$ and $[t\pm]i$ is -1, then *FP[BLR]* is incremented. If the classifier predicts label -1 and $[t\pm]i$ is $+1$, then *FN[BLR]* is incremented.
    b. A binary classifier is built from $\Delta[-i]$ and applied to Di using the procedures described in Section 3.2 (the SVM approach).
        i. If the classifier predicts label $+1$ and $[t\pm]i$ is $+1$, then *TP[SVM]* is incremented. If the classifier predicts label $+1$ and $[t\pm]i$ is -1, then *FP[SVM]* is incremented. If the classifier predicts label -1 and $[t\pm]i$ is $+1$, then *FN[SVM]* is incremented.
3. The precision recall and F scores are computed from *{TP[BLR], FP[BLR], FN[BLR]}* and from *{TP[SVM], FP[SVM], FN[SVM]}*.

4. Results

In figures 3-6, BLR (Bayesian Logistic Regression) denotes the results of our approach and SVM denotes the results of the baseline approach. As seen in figure 3, the accuracy comparison between BLR and SVM yielded mixed results. On the Japanese dataset, BLR exhibited greater

accuracy (*F* score) for all event types considered. On the French dataset, BLR exhibited greater accuracy for two of four event types considered and worse for the other two.

As seen in figures 4–6, BLR tends to exhibit more balance between precision and recall than SVM. This is by design and is usually preferred. Classifiers are built to maximize the area under the ROC curve. In BLR, this is achieved via adjusting the bias coefficient *b* such that we maximize the *F* score. In unbalanced data sets, the linear separating boundary tends to be pushed towards the smaller class (usually the positively labeled class) resulting in a very high precision and poor recall. Moving the separating boundary away from the smaller class results in trading a little bit of precision but gaining a much higher recall i.e., maximizing the *F* score. In some circumstances, recall may be weighted higher than precision, in which case the *F* score that weighs the recall higher should be used as the metric that is maximized. This is easily achieved in BLR by replacing the *F* score that balances precision and recall with an *F* score that weighs recall higher than precision.

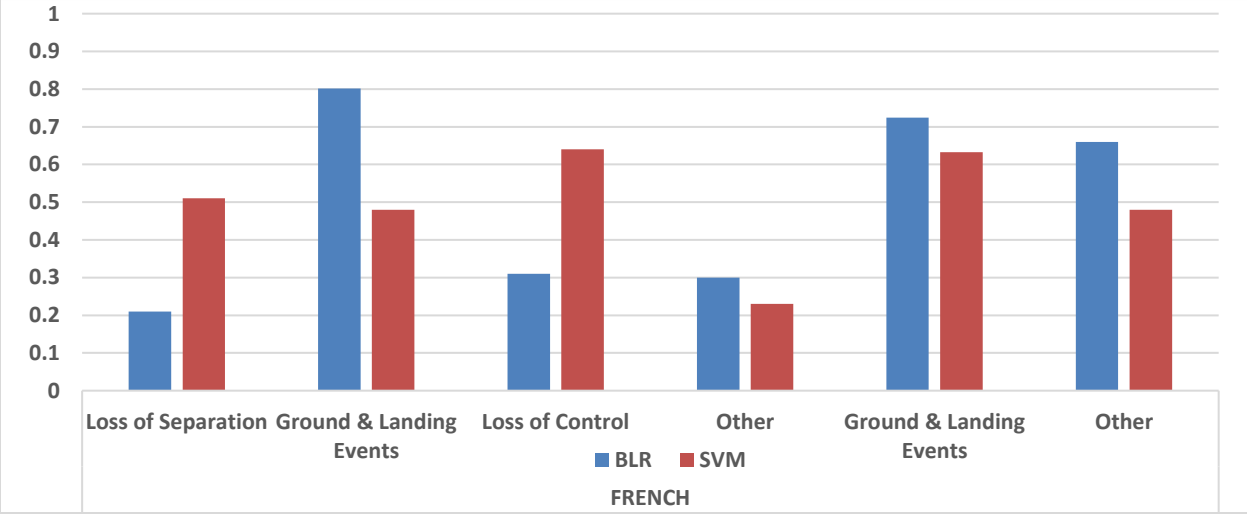

FIGURE 3: F Scores for All Event Types

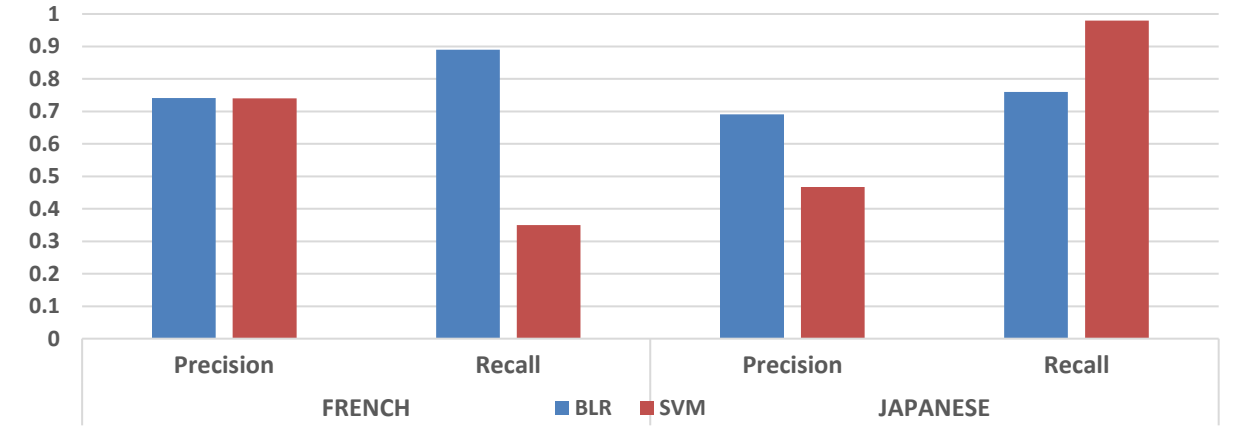

FIGURE 4: Precision and Recall for the Ground & Landing Event Type

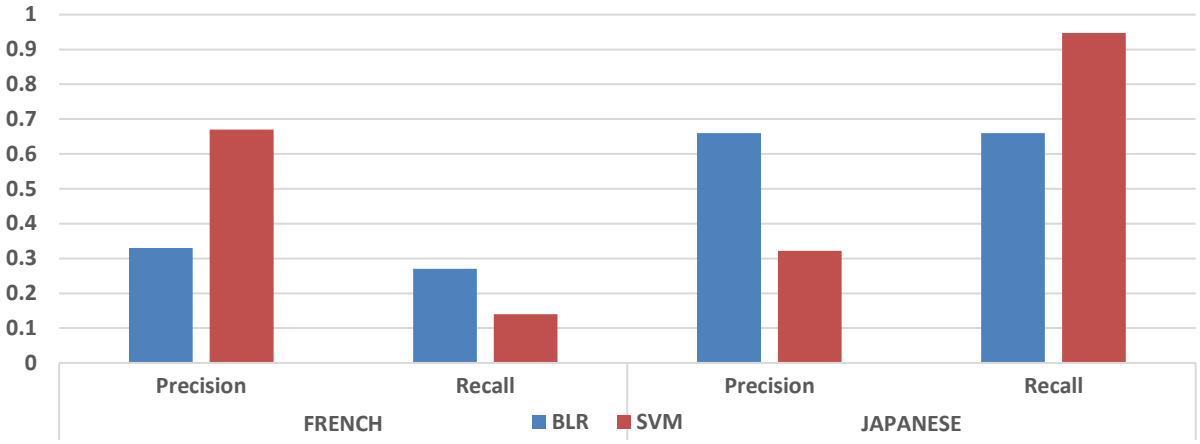

FIGURE 5: Precision and Recall for the Other Event Type

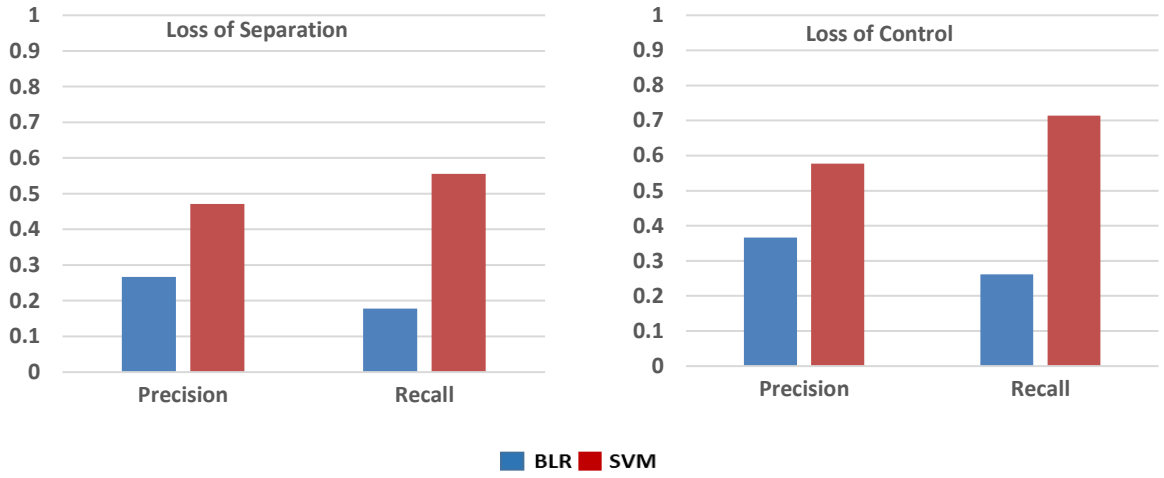

FIGURE 6: Precision and Recall for the Loss of Separation and Loss of Control Event Types

5. Related Work

5.1 Aviation Report Classification

This work is a contribution to addressing the multi-label classification problem for aviation events. Literature discussing the analysis of aviation incidents based on non-narrative data, for example, [10], is beyond our scope.

In [11] a one-versus-all semi-supervised approach automatically augments the labeled training reports using a bootstrapping method applied to a set of unlabeled reports. Pre-labeled reports determine terms that are highly indicative of an event type, then unlabeled reports with enough indicative terms are assigned an event-type label. These reports are added to the set of already labeled reports and the process is iterated. The final set of labeled reports is used to train classifiers in a standard way.

In [3] several different approaches are developed. One approach applies a bootstrapping method to the labeled training data to automatically induce sets of indicative terms for each event type. Then, in one-versus-all fashion, a simple heuristic is used to label new documents based only on the indicative terms. Two other approaches use two different multi-label learning strategies beyond one-versus-all (on top of statistical classification, rather than simple keyword heuristics). As a base classifier in these approaches, several different kinds of SVMs are considered (including a transductive SVM, which can be considered semi-supervised).

In [12] several one-versus-all approaches are developed to utilize annotator rationales in addition to the event type labels on the reports. These authors assume that the manual annotation of the training data assigns event types to the reports, as well as, identifies snippets of text in the reports that the annotator deems as rationales for assigning types. The idea being that the annotators highlight parts of the reports that were particularly influential in the assignment of event types. This information, in addition to the event type labels themselves, is utilized in a variety of ways to train classifiers.

In [1] several one-versus-all approaches are developed which modify simple word gram features by utilizing several kinds of automatically induced, simple semantic information. For example, the authors identify word collocations, pairs of words that tend to appear next to each other, and replace those with the concatenation of the constituent words. Also, the authors use part of speech information to add disambiguation information to words, e.g. pilot-NOUN vs. pilot-VERB. The authors use several other kinds of feature modifications of this nature.

In [13] two one-versus-all approaches are developed. The first is a standard SVM but using simulated annealing to set the hyper-parameters. The second is based on non-negative matrix factorization (NMF). Through experiments they observe the SVM approach to outperform (by a small margin) the NMF-based approach.

In [14] a semi-supervised clustering approach is developed.[7] This approach creates a vocabulary by computing the information gain of all words in the labeled training reports and drops all but the top 1000 words. Each training report (labeled or not) is then mapped to a length 1000 Boolean vector whose i^{th} entry is 1 if the report text contains the i^{th} word, else the i^{th} entry is 0. A soft clustering (with a fixed number of clusters) is computed over the labeled and unlabeled training reports. Associated with each cluster is a set of weights on the event types (the weights need not sum to one). This completes the training procedure (the event type classifier model is the soft clustering). To assign events to a new report (test report), the

[7] In multi-label learning terminology, this approach is considered a higher-order strategy since event type classifications are not made independently.

nearest cluster is found. The event type weights associated with this cluster are assigned to the new report. These weights are used to assign event types to the new report.[8]

In [15] approaches are discussed for automatically classifying French aviation reports and finding reports that describe similar incidents. Regarding the classification approach[9], the process of building a classifier from a manually annotated French aviation report corpus proceeds as follows. Phrases (e.g. "pilote automatique") are identified using occurrence and parse information. Words and phrases are grouped into term categories using ontologies and expert knowledge (e.g. "PA" and "pilote automatique" are grouped together). For each event type and each term category, the conditional probability is estimated that a report contains a word or phrase in the term category given that the report was manually assigned the event type. Moreover, weights are assigned to each term category and a classification threshold is fixed for each event type. The conditional probability estimates, the weights, and the thresholds form the classifier. A new report is automatically assigned an event type if, for all term categories with a word or phrase contained in the report, the weighted sum of the conditional probabilities associated with the event type exceeds a fixed threshold. Pimm et al. discuss preliminary results of an empirical evaluation of the classification accuracy.

5.2 Classification: Bayesian Logistic Regression and Class Imbalance

Bayesian logistic regression: A number of studies have been published describing the application of Bayesian logistic regression to classify high-dimensional datasets (e.g. collections of text reports). A few of these studies are discussed here. The reader is referred to Section 5.1 of [16] for more background.

In [17] and [18], a Bayesian logistic regression approach is developed for classifying high dimensional datasets emphasizing the use of a Laplace prior to address over-fitting (a problem made particularly acute by high-dimensional data). The hyper-parameter λ is set through cross-validation.

In [19], a fully Bayesian logistic regression approach is developed for classifying high dimensional datasets. A Laplace prior is used and the hyper-parameter λ is integrated out analytically making use of an improper Jeffrey's hyper-prior. Results from [20] are used to carry out the integration.

In [21] adaptive, Laplacian prior, Bayesian linear regression is studied with extensions to logistic regression. That study provides a theoretical basis for the setting of the coefficient weights w, hence the use of the term "adaptive".

[8] Ahmed et al. do not go into much detail here, so we are unclear as to how the nearest cluster is defined or how the event type weights on the new report are mapped to the assignment of event types (yes or no) to the report.
[9] Pimm et al. do not go into much detail so we are unclear as to the specifics of the classification approach and are speculating somewhat in our description.

Class imbalance: In situations where one class is significantly less common than the rest, classifiers can be prone to under-emphasize the rare class. This problem has been studied fairly extensively. The reader is referred to [22] for a survey of approaches in machine learning to account for class imbalance, as well as, [23] and [24] for recent work in the Statistics community on modified Bayesian logistic regression to account for class imbalance.

In [25] an under-sampling approach is developed for binary classification based on SVMs. The key idea is that vectors from the common class that are closest to the decision boundary are more informative. For each positively labeled vector, the η nearest (in terms of weighted Euclidean distance) negatively labeled vectors are selected where η is a fixed constant (care is taken to avoid repetitive selection).

5.3 Summary of Differences and Similarities Between Our Approach and the Literature

Like the literature discussed in subsection 5.1, we address the aviation event multi-label classification problem. We employ a simple one-versus-all approach, unlike [14] and [3] which employ higher-order approaches. Unlike all the literature in subsection 5.1, except [15], we empirically evaluated our approach on foreign language aviation incident reports. Pimm et al., however, provide only sketchy empirical results do not report the outcome of a detailed and systematic study, as we do.

Like the literature discussed in subsection 5.2, we develop a binary text classification approach. Like [17], we utilize Bayesian logistic regression with a Laplace prior. Unlike that paper, we set the hyper-parameter λ through joint maximization and a Gamma hyper-prior (rather than setting λ through cross-validation). Unlike [19], our approach is not fully Bayesian. In that paper, λ is analytically integrated out, thus no parameters need be set manually or through cross-validation. We place a Gamma hyper-prior on λ which, in turn, has hyper-parameters (shape and rate) that we manually set to 0.001. Like [25], we under-sample near the boundary to address class imbalance.

Utilizing the theoretical results in [21] would be an interesting direction for future work. Specifically, develop and test an algorithm for setting the coefficient weights w in theoretically principled way, rather than heuristically as we do now.

6. Conclusion

We developed and implemented an approach, based on a Bayesian logistic regression algorithm for binary text classification, to address the aviation event multi-label classification problem. The primary novelty in our approach is the feature selection procedure. For comparison purposes, we implemented an approach similar to the SVM based approach described in Sections 5.2.2 and 6.4.1 of [3]. We collected and annotated French and Japanese aviation incident reports. We carried out a series of experiments comparing the accuracy of the two approaches. The results were mixed. On the Japanese dataset, our approach exhibited greater accuracy (F score) for all event types considered. On the French dataset, our approach exhibited

greater accuracy for two of four event types and worse for the other two. Our approach tends to exhibit more balance between precision and recall than the SVM based approach.

Two avenues for future work seem promising, one algorithmic, the other empirical. The first would improve the binary classification algorithm by utilizing the results in [21] for setting coefficient weights in a theoretically principled way, rather than heuristically as we do now. The second is to improve experimental comparison by collecting and annotating larger foreign language aviation datasets.

7. Appendix – Event Type Hierarchy Details

Loss of Separation

A loss of separation occurred whenever specified separation minima were breached. Minimum separation standards for airspace are specified by air traffic service authorities, based on ICAO standards.

Airborne: This refers to a loss of radar separation involving instrument flight rules (IFR) aircraft, loss of separation involving visual flight rules (VFR) aircraft in airspace where minimum separation standards are prescribed, a suspected loss of separation involving formation flights, and a loss of separation involving non-radar standards.

Ground: This refers to an aerodrome surface loss of separation, including any ground surveillance alert between two aircraft, any ground surveillance alert between an aircraft and a vehicle, any suspected loss of runway/airport surface separation between two aircraft, any suspected loss of runway/airport surface separation between an aircraft and a vehicle, or any suspected loss of runway/airport surface separation between an aircraft and a pedestrian.

Terrain: This refers to a loss of separation between an IFR aircraft and terrain or obstacles (for example, operations below minimum vectoring altitude), or an incident involving a VFR aircraft in proximity to terrain or obstructions that the employee providing air traffic services determines affected the safety of flight. This category includes controlled flight into or toward terrain, namely, instances when an airworthy aircraft under the complete control of the pilot was (inadvertently) flown into or toward terrain, water, or an obstacle.

Loss of Control

A loss of control occurred whenever control of the aircraft was lost either terminally or transitorily.

Airborne: This refers to loss of control while the aircraft was in flight -- a major cause of fatal aircraft accidents.

Ground: This refers to loss of control while the aircraft was on the ground.

Deviations and ATC Anomalies

Examples of deviations or air traffic control (ATC) anomalies include: an aircraft entered airspace on other than the expected or intended altitude; an aircraft operated at an altitude, routing, or airspeed that the employee providing air traffic services determines affected the safety of flight; or an aircraft entered special use airspace (e.g., a warning area, military operations are, or ATC-assigned airspace) without coordination and/or authorization. This category does not include deviations or anomalies in which a loss of separation occurred.

Airspace infringement: This refers to an aircraft entering notified airspace without previously requesting and obtaining clearance from the controlling authority of that airspace, or entering the airspace under conditions that were not contained in the clearance.

Altitude deviation: This refers to an aircraft failing to fly at the level to which it has been cleared, also known as level bust.

Route/course deviation: This refers to an aircraft deviating from its cleared flight path.

Speed deviation: This refers to an aircraft deviating from its cleared speed.

Ground and Landing Events

These are events that occurred at the approach and landing phase of the flight that did not involve a loss of separation, but may have affected the safety of operations. These are also events in the aerodrome environment that did not involve a loss of separation, but may have affected the safety of operations.

Missed approach / Go-around: This refers an aborted landing initiated by either a flight crew or ATC involving turbojet aircraft within one-half of a mile of the arrival threshold not involving practice approaches.

Forced landing: This refers to a landing by an aircraft under factors outside the pilot's control, such as the failure of engines, systems, components or weather which makes continued flight impossible.

Landing without Clearance: This refers to an aircraft landing without obtaining proper clearance prior to landing. This includes instances in which an aircraft unexpectedly landed or departed, or attempted to land or depart, on a runway or surface.

Runway incursion: This refers to an occurrence of an aerodrome involving the incorrect presence of an aircraft, vehicle, or person on the protected area of a surface designated for the landing and takeoff of aircraft.

Runway excursion: This refers to an aircraft unintentionally maneuvering off the runway or taxiway. This also includes overrun on takeoff (departing aircraft failing to become airborne before reaching the end of the runway), overrun on landing (landing aircraft failing to stop before reaching the end of the runway), undershoot on landing (landing aircraft touching down in the undershoot area of the designated runway).

Aborted takeoff: This refers to instances in which any part of the aircraft crossing over the runway hold-short line and the controller canceling the takeoff or the flight crew aborting the takeoff.

Other landing/takeoff issues: Examples include instances in which there was an unstable approach to runway during landing; cases where landing took place in the wrong airport or runway; instances in which an aircraft landed or flew an unrestricted low approach to a closed runway (or portion thereof).

Other Events

Fire / smoke / fumes: This refers to the presence of fire or smoke is in or on the aircraft while in flight or on the ground (which is not the result of aircraft crashing).

Equipment malfunction: This refers to a failure or malfunction occurring involving aircraft or ATC equipment, including engine-related problems, errors in software systems, and parts separating from an aircraft.

Wildlife collision: This refers to the aircraft colliding with or taking evasive action to avoid colliding with wildlife (particularly birds) on the movement area of an aerodrome. Includes instances where evasive action was taken by the flight crew that led to a collision off the movement area of the aerodrome or to consequences other than a collision (e.g., gear collapsing).

Criminal activity: This refers to a criminal or security related act occurring which resulted in an accident or incident. These include hijacking and/or aircraft theft, flight control interference, sabotage, suicide, and acts of war.

Other: This refers to occurrences that cannot be classified as any of the event types listed in this taxonomy.

Acknowledgements

We are thankful for the assistance provided by our MITRE colleagues: Armen Gomtsyan, Michelle Harper, Vanesa Jurica, Shaun Michel, Danijela Nardelli, and Evelyne Tzoukermann.

Bibliography

[1] S. Wolfe, "Wordplay: An Examination of Semantic Approaches to Classify Safety Reports," in *Proceedings of the American Institute of Aeronautics and Astronautics Conference*, 2007.

[2] M.-L. Zhang and Z.-H. Zhou, "A Review on Multi-Label Learning Algorithms," *IEEE Transactions on Knowledge and Data Engineering,* vol. 28, no. 8, pp. 1819-1837, 2014.

[3] M. A. Ul Abedin, V. Ng and L. Khan, "Cause Identification from Aviation Safety Incident Reports via Weakly Supervised Semantic Lexicon Construction," *Journal of Artificial Intelligence Research,* vol. 38, pp. 569-631, 2010.

[4] C. Aggarwal and C. Zhai, "A Survey of Text Classification Algorithms," in *Mining Text Data*, C. Aggarwal and C. Zhai, Eds., Springer US, 2012, pp. 77-128.

[5] C. Forbes, M. Evans, N. Hastings and B. Peacock, Statistical Distributions, 4th, Ed., Hoboken, New Jersey: John Wiley & Sons, Inc., 2011.

[6] K. Svanberg, "A Class of Globally Convergent Otimization Methods Based on Conservative Convex Separable Approximations," *SIAM Journal on Optimization,* vol. 12, no. 2, pp. 555-573, 2002.

[7] A. Ng, "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance," in *International Conference on Machine Learning (ICML)*, 2004.

[8] J. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges and A. Smola, Eds., Cambridge, MA: MIT Press, 1999, pp. 185-208.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and W. Ian, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations,* vol. 11, no. 1, pp. 10-18, 2009.

[10] N. Maille, "On the Use of Data-Mining Algorithms to Improve FOQA Tools for Airlines," in *Proceedings of the IEEE Aerospace Conference*, 2013.

[11] I. Persing and V. Ng, "Semi-Supervised Cause Identification from Aviation Safety Reports," in *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2009.

[12] M. A. Ul Abedin, V. Ng and L. R. Khan, "Learning Cause Identifiers from Annotator Rationales," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[13] N. Oza, P. Castle and J. Stutz, "Classification of Aeronautics System Health and Safety Documents," *IEEE Transactions on Systems, Man, and Cybernetics -- Part C: Applications and Reviews,* vol. 39, no. 6, pp. 670-680, 2009.

[14] M. S. Ahmed, L. Khan, N. Oza and M. Rajeswari, "Multi-Label ASRS Dataset Classification Using Semi-Supervised Subspace Clustering," in *Proceedings of the Conference on Intelligent Data Understanding (CIDU)*, 2010.

[15] C. Pimm, C. Raynal, N. Tulechki, E. Hermann, G. Caudy and L. Tanguy, "Natural Language Processing (NLP) Tools for the Analysis of Incident and Accident Reports," in *Proceedings of the International Conference on Human-Computer Interaction in Aerospace (HCI-Aero)*, 2011.

[16] D. Vidaurre, C. Bielza and P. Larrañaga, "A Survey of L1 Regression," *International Statistical Review,* vol. 81, no. 3, p. 361–387, 2013.

[17] A. Genkin, D. Lewis and D. Madigan, "Large-Scale Bayesian Logistic Regression for Text Categorization," *Technometrics,* vol. 49, no. 3, pp. 291- 304, 2007.

[18] S. K. Shevade and S. S. Keerthi, "A Simple and Efficient Algorithm for Gene Selection Using Sparse Logistic Regression," *Bioinformatics,* vol. 19, no. 17, pp. 2246-2253, 2003.

[19] G. Cawley, N. Talbot and M. Girolami, "Sparse Multinomial Logistic Regression via Bayesian L1 Regularisation," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 19, J. C. Platt, B. Schölkopf and T. Hoffman, Eds., Neural Information Processing Systems Foundation, Inc., 2006, pp. 209-216.

[20] P. Williams, "Bayesian Regularization and Pruning Using a Laplace Prior," *Neural Computation,* vol. 7, pp. 117-143, 1994.

[21] H. Zou, "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association,* vol. 101, no. 476, pp. 1418-1429, 2006.

[22] H. He and Y. Ma, Eds., Imbalanced Learning: Foundations, Algorithms, and Applications, Hoboken, New Jersey: John Wiley & Sons, Inc., 2013.

[23] M. Maalouf and T. B. Trafalis, "Robust Weighted Kernel Logistic Regression in Imbalanced and Rare Events Data," *Computational Statistics and Data Analysis,* vol. 55, pp. 168-183, 2011.

[24] M. Maalouf and M. Siddiqi, "Weighted Logistic Regression for Large-Scale Imbalanced and Rare Events Data," *Knowledge-Based Systems,* vol. 59, pp. 142-148, 2014.

[25] A. Anand, G. Pugalenthi , G. Fogel and P. N. Suganthan, "An Approach for Classification of Highly Imbalanced Data Using Weighting and Undersampling," *Amino Acids,* vol. 39, no. 5, pp. 1385-1391, 2010.