AUTOMATING THE CLASSIFICATION OF URBAN ISSUE REPORTS: AN OPTIMAL STOPPING APPROACH

Yasitha Warahena Liyanage[⋆] *Daphney–Stavroula Zois*[⋆] *Charalampos Chelmis*[†] *Mengfan Yao*[†]

[⋆]Electrical and Computer Engineering Department
[†]Computer Science Department
University at Albany, SUNY, Albany, NY, USA
Emails: {yliyanage, myao, dzois, cchelmis}@albany.edu

ABSTRACT

Empowering citizens to interact directly with their local governments through civic engagement platforms has emerged as an easy way to resolve urban issues. However, for authorities to manually process reported issues is both impractical and inefficient; accurate, online and near–real–time processing methods are necessary to maintain citizens' satisfaction with their local governments. Herein, an optimal stopping framework is proposed to process urban issue requests quickly and accurately. The optimal classification and stopping rules are derived, and significant reduction in time–to–decision without sacrificing accuracy is demonstrated on a real–world dataset from SeeClickFix.

Index Terms— Civic engagement, classification, government 2.0, optimal stopping theory, quickest detection

1. INTRODUCTION

"Government 2.0" applications have recently appeared as a facet of smart cities [1, 2], with civic engagement platforms such as SeeClickFix [3] becoming indispensable in making them more effective and efficient. While such platforms provide citizens with computer–mediated urban issue (e.g. potholes or noise complaints) reporting capabilities [3–5], citizens' continuous engagement and participation cannot be guaranteed unless the issues they report are timely acknowledged and addressed by their local governments.

As the ability to "comprehend" urban issues reported in participatory platforms is at the core of citizen services, methods to bridge the intelligence gap between computer–mediated reported issues and humans responsible for reviewing them have recently been proposed. However, such methods are limited either to binary classification of reports into categories [6–8] or importance [9, 10], or require large training datasets to achieve good accuracy [9, 11]. The scalability and timeliness of such methods, although critical, have also largely been ignored. Beyond urban issue report classification, multiclass classification is far more challenging than the binary problem with two mutually–exclusive classes [12–14]; the main difficulties are the rapid degradation in classification accuracy and explosion in computational complexity as the number of classes increases. Simplistic one–versus–the–rest and pairwise classification strategies [14–17] are therefore typically used as alternatives.

Herein, we build upon the framework introduced in our prior work [8, 10] to accelerate the response of local governments to urban issue requests without additional steps from a city's staff, by addressing the challenging problem of multiclass classification. Specifically, we formulate the classification of urban issue reports as a sequential hypothesis testing problem, in which the goal is to classify each report as it becomes available by sequentially reviewing features, starting from the most informative, and stopping once it is determined that the inclusion of additional features cannot further improve the accuracy of the classification decision. As a result, our approach uses a varying number of features to classify individual reports. This is in stark contrast to popular feature selection and dimensionality reduction methods [18–21] used to identify a subset of discriminative features, common to all instances for classification. Thus, it provides a viable, realistic and timely solution for processing urban issue requests by efficiently utilizing computational resources rather than blindly relying on the same fixed set of features for all issues, as done by state–of–the–art classifiers.

2. PROBLEM DESCRIPTION

Consider a set $\mathcal{S}$ of instances, with each instance $s \in \mathcal{S}$ being associated with a vector $f(s) = \{y_1, y_2, \ldots, y_K\}$ of K features. Each instance s may belong to one of L possible hypotheses, with corresponding *a priori* probability p_i for each hypothesis $H_i, i = 1, 2, \ldots, L$. We assume for simplicity that features $y_1, y_2, \ldots, y_n$ are *independent under each hypothesis* H_i, and thus, the conditional joint probability of $\{y_1, \ldots, y_n\}$ is given as $P(y_1, \ldots, y_n|H_i) = \prod_{l=1}^{n} p(y_l|H_i)$. Even though

This material is based upon work supported by the National Science Foundation under Grant No. ECCS–1737443.

validation of this assumption is beyond the scope of this paper, we find our proposed method to work well in practice. Moreover, each coefficient $c_n, n = 1, 2, \ldots, K$, represents the cost of evaluating feature y_n, and misclassification cost $M_{i,j}$ denotes the cost of selecting hypothesis H_j when instead some other hypothesis $H_i, i \neq j$, is true.

To select between one of L possible hypotheses for each s, the proposed approach evaluates features sequentially, where at each step it has to decide between stopping and continuing based on the accumulated information thus far and the cost of evaluating the remaining features. Herein, we introduce a pair of random variables (R, D_R), where $0 \leq R \leq K$ (referred to as *stopping time* [22] in decision theory) denotes the feature at which the framework makes a classification decision at, and $1 \leq D_R \leq L$, which depends on R, denotes the possibility to select among the L hypotheses. The event $\{R = n\}$ depends only on the feature set $\{y_1, y_2, \ldots, y_n\}$, whereas the event $\{D_R = m\}$ represents choosing hypothesis m based on the information accumulated up till feature R. The goal is to select random variables R and D_R without sacrificing accuracy by solving the following optimization problem:

$$\underset{R, D_R}{\text{minimize}}\ J(R, D_R), \tag{1}$$

over cost function:

$$J(R, D_R) = \mathbb{E}\Bigg\{ \sum_{n=1}^{R} c_n + \sum_{j=1}^{L}\sum_{i=1}^{L} M_{ij} P(D_R = j, H_i) \Bigg\}, \tag{2}$$

where the first term denotes the cost of evaluating features, and the second term penalizes the misclassification cost.

3. OPTIMAL STRATEGIES

In order to solve the optimization problem defined in Eq. (1), we use a sufficient statistic of the accumulated information, the *a posteriori probability* vector $\pi_n \triangleq [\pi_n^1, \pi_n^2, \ldots, \pi_n^L]$, where the nth feature is evaluated to generate outcome y_n, and $\pi_n^i = P(H_i | y_1, \ldots, y_n)$. Note that π_n can be computed recursively as in Lemma 1.

Lemma 1. *The a posteriori probability vector π_n is given by:*

$$\pi_n = \frac{\pi_{n-1}\, \mathrm{diag}(\Delta_n(y_n))}{\pi_{n-1}\Delta_n^T(y_n)}, \tag{3}$$

where $\Delta_n(y_n) = [P(y_n|H_1), P(y_n|H_2), \ldots, P(y_n|H_L)]$, $\mathrm{diag}(A)$ denotes a diagonal matrix with diagonal elements being the elements in vector A, and $\pi_0 = [p_1, p_2, \ldots, p_L]$.

Lemma 2. *Based on the fact that $x_R = \sum_{n=0}^{K} x_n \mathbb{1}_{\{R=n\}}$ for any sequence of random variables $\{x_n\}$, where $\mathbb{1}_A$ is the indicator function for event A (i.e., $\mathbb{1}_A = 1$ when A occurs, and $\mathbb{1}_A = 0$ otherwise), the probability $P(D_R = j, H_i)$ can we written as follows:*

$$P(D_R = j, H_i) = \mathbb{E}\left\{ \pi_R^i \mathbb{1}_{\{D_R = j\}} \right\}. \tag{4}$$

Using Lemma 2, the average cost in Eq. (2) can be written compactly as:

$$J(R, D_R) = \mathbb{E}\left\{ \sum_{n=1}^{R} c_n + \sum_{j=1}^{L} \Big(\sum_{i=1}^{L} M_{ij} \pi_R^i \Big) \mathbb{1}_{\{D_R = j\}} \right\}. \tag{5}$$

Note that we can rewrite the average cost in Eq. (5) using the *a posteriori probability* vector π_n as follows:

$$J(R, D_R) = \mathbb{E}\left\{ \sum_{n=1}^{R} c_n + \sum_{j=1}^{L} \pi_R M_j^T \mathbb{1}_{\{D_R = j\}} \right\}, \tag{6}$$

where $M_j \triangleq [M_{1,j}, M_{2,j}, \ldots, M_{L,j}]$.

To obtain the optimal stopping time R, we must first obtain the optimal decision rule D_R for any given R. In the process of finding the optimal decision, we need to find a lower bound (independent of D_R) for the second term inside the expectation in Eq. (6), which is the part of the equation that depends on D_R. Theorem 3 provides such a bound.

Theorem 3. *For any classification rule D_R given stopping time R, $\sum_{j=1}^{L} \pi_R M_j^T \mathbb{1}_{\{D_R = j\}} \geqslant g(\pi_R)$, where $g(\pi_R) \triangleq \min_{1 \leqslant j \leqslant L} \left[\pi_R M_j^T\right]$. The optimal rule is defined as follows:*

$$D_R^{optimal} = \arg\min_{1 \leqslant j \leqslant L} \left[\pi_R M_j^T\right]. \tag{7}$$

From Theorem 3, we conclude that:

$$J(R, D_R) \geqslant J(R, D_R^{optimal}), \text{ where}$$
$$J(R, D_R^{optimal}) = \min_{D_R} J(R, D_R). \tag{8}$$

Thus, we can reduce the cost function in Eq. (6) to one which depends only on the stopping time R as follows:

$$\widetilde{J}(R) = \mathbb{E}\left\{ \sum_{n=1}^{R} c_n + g(\pi_R) \right\}. \tag{9}$$

To optimize the cost function in Eq. (9) with respect to R, we need to solve the following optimization problem:

$$\min_{R \geqslant 0} \widetilde{J}(R) = \min_{R \geqslant 0} \mathbb{E}\left\{ \sum_{n=1}^{R} c_n + g(\pi_R) \right\}, \tag{10}$$

which constitutes a classical problem in optimal stopping theory for Markov processes [22]. Since $R \in \{0, 1, \ldots, K\}$, the optimum strategy will consist of a maximum of $K+1$ stages, where the optimum scheme must minimize the corresponding average cost going from stages 0 to K. The solution can be obtained by using *dynamic programming* principles [23].

Theorem 4. *For $n = K-1, \ldots, 0$, the function $\bar{J}_n(\pi_n)$ is related to $\bar{J}_{n+1}(\pi_{n+1})$ through the equation:*

$$\bar{J}_n(\pi_n) = \min\Bigg[g(\pi_n), c_{n+1} + \sum_{y_{n+1}} \pi_n \Delta_{n+1}^T(y_{n+1}) \times \bar{J}_{n+1}\Bigg(\frac{\pi_n\, \mathrm{diag}\left(\Delta_{n+1}(y_{n+1})\right)}{\pi_n \Delta_{n+1}^T(y_{n+1})} \Bigg) \Bigg], \tag{11}$$

where $\bar{J}_K(\pi_K) = g(\pi_K)$.

The optimal stopping strategy derived from Eq. (11) has a very intuitive structure. The optimal stopping strategy stops at the stage n, where the cost of stopping (the first expression in the minimization) is no greater than the expected cost of continuing given all information accumulated at the current stage n (the second expression in the minimization). Specifically, at each stage n, our method faces two options given π_n: (i) stop evaluating features and selecting optimally between the L hypotheses, or (ii) continue and evaluate the next feature. The cost of stopping is $g(\pi_n)$, whereas the cost of continuing is $c_{n+1} + \sum_{y_{n+1}} \pi_n \Delta_{n+1}^T(y_{n+1}) \times \bar{J}_{n+1}\left(\frac{\pi_n \operatorname{diag}\left(\Delta_{n+1}(y_{n+1})\right)}{\pi_n \Delta_{n+1}^T(y_{n+1})}\right)$.

4. ASSESS ALGORITHM

In this section, we present ASSESS, a novel algorithm to Automatically optimally and timely claSSify rEported urban iSSues based on Lemma 1, and Theorems 3 and 4. Initially, the posterior probability vector π_0 is set to $[p_1, p_2, \ldots, p_L]$, and the two terms in Eq. (11) are compared. If the first term is not greater than the second, ASSESS classifies the instance under examination to the appropriate class, based on the optimal rule of Eq. (7). Otherwise, the first feature is evaluated. ASSESS repeats these steps until either it decides to classify the instance using $< K$ features, or the feature vector is exhausted, in which case classification is performed using all K features.

Next, we discuss some practical considerations. We use a smoothed maximum likelihood estimator to estimate $p(y_n|H_i), n = 1, \ldots, K, i = 1, \ldots, L$, from training data as follows $\hat{p}(y_n|H_i) = \frac{N_{n,i}+1}{N_i+V}$, where $N_{n,i}$ denotes the number of samples that give rise to outcome y_n and belong to hypothesis H_i, N_i denotes the total number of samples in the training dataset that belong to hypothesis H_i and V is the maximum outcome among all features. We estimate the *a priori* probabilities as $P(H_i) = \frac{N_i}{\sum_{i=1}^{L} N_i}, i = 1, \ldots, L$. Quantizing the interval $[0, 1]$ with a predefined accuracy (e.g., 0.1) for L values such that $\sum_{i=1}^{L} \pi_n^i = 1$ to generate different possible vectors π_n, enables the efficient computation of a $(K + 1) \times d$ matrix, where each row corresponds to $K + 1$ values $\bar{J}_n(\pi_n), n = 0, 1, \ldots, K$, computed using Theorem 4 for all possible d vectors of π_n. Since this computation requires only *a priori* information, it can be conducted once offline. Hence, the complexity of calculating $\bar{J}_n(\pi_n)$ is independent from the actual number of instances, which can be huge. Finally, different features can hinder or facilitate the quick identification of the hypothesis of which an instance may belong to. Consider an example of classifying urban issue reports as either 'Parking Enforcement' or 'Code Violation' using two features y_1 and y_2, where y_1 is the number of appearances of keyword 'code' in the title, and y_2 is the number of tags in an issue. In this case, intuitively, appearance of the keyword 'code' can potentially simplify the process of identifying the issue type compared to the number of tags in an issue. As a result, if feature y_2 was to be examined first, it would be very probable for feature y_1 to be examined as well to improve the chances of accurate classification. Alternatively, if y_1 was to be evaluated first, ASSESS could reach a decision using one feature only. To avoid the computational complexity of evaluating all $K!$ possible feature orderings, we sort features in increasing order of the sum of type I and II errors (considering the true class as the positive class and all the rest classes as a single negative class), scaled by the cost coefficient of the nth feature to promote low cost features that at the same time are expected to result in few errors.

5. URBAN ISSUE CLASSIFICATION

We illustrate the performance of ASSESS on a real–world dataset of $2,195$ issues, spanning a time period between Jan 5, 2010 and Feb 10, 2018, for the capital of the state of New York, collected from SeeClickFix[1]. Without loss of generality, we consider a set of four hypotheses, i.e., {Parking Enforcement, Code Violation, Traffic Signal Repair, Signs (missing, needed, or damaged)}. The goal is to assign each issue to one of the four hypotheses, using a total of $1,606$ features, directly extracted from issues' title and description by tokenizing sentences into unigrams, removing punctuation (e.g., periods, commas, and apostrophes), stopwords (e.g., "a", "the", "there"), and digits (e.g., "8th", "31st"), and stemming each word to its root (e.g., replace "parked" with "park"). A feature value corresponds to the number of appearances of a specific word in the issue report, with words being present in $\geq 95\%$ and $\leq 2\%$ of all issues excluded.

We compare ASSESS's performance to (i) a standard Bayesian detection method [24] that uses the top 1, 5, 10, 50, 100, 200, 500 features ordered using the proposed ordering technique, as well as all available features, (ii) ACTION [8], extended to multiclass classification using one–vs–the–rest (i.e., 4 classifiers are constructed such that 1 out of 4 classes is the positive class and the rest negative, and the predicted class corresponds to the maximum posterior) and one–vs–one (i.e., 6 classifiers are constructed for all pairwise combinations of the 4 classes, and the maximum posterior is used for classification) schemes [16], (iii) prior work, i.e., Support Vector Machine with feature selection (SVM–FS) [11] with linear (SVM–L) and Gaussian (SVM–G) kernels, and PCA (SVM–PCA) for dimensionality reduction, and (iv) inherently multiclass classifiers, namely Random Forest (RF) with maximum tree depths $d = 5, 10$, and XG Boosting (XG–B), which have been shown to achieve good performance while being relatively fast compared to other classification models [25,26]. In our experiments, $L = 4$ (i.e., 4 issue types), misclassification costs are set to $M_{i,j} = 1, \forall i \neq j$ and $M_{i,j} = 0, \forall i = j$,

[1] https://seeclickfix.com/albany-county

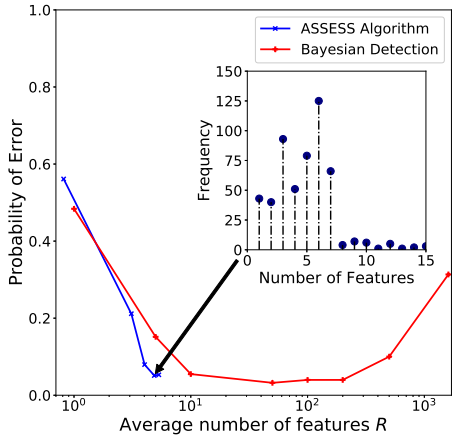

Fig. 1. Probability of error versus average number of features used. Inset shows the distribution of number of features used by ASSESS to classify each issue for an average of 5 features.

and feature costs $c_n \in \{0, 0.01, 0.10, 0.15, 0.17, 0.20\}$ are considered. Five–fold cross validation results are reported.

Fig. 1 shows the error probability achieved by both ASSESS and the standard Bayesian method as the average number of features used increases. Intuitively, with a small number of features, both ASSESS and the standard Bayesian method exhibit large error probabilities, whereas when the number of features increases, the performance improves dramatically. Observe that the performance of the standard Bayesian method is stable when the number of features used is between 10 and 200, and degrades when more than 200 features are used. This behavior can be explained as a result of the proposed feature ordering technique. Specifically, as noisy features are ranked towards the end of the list, features beyond the top 200 may introduce noise, significantly impairing classification performance. Nevertheless, ASSESS reaches the performance of the standard Bayesian method (top 10 features) using only $\sim$5 features on average; this corresponds to $\sim$50% reduction in the number of features used by the standard Bayesian method. The inset in Fig. 1 illustrates the variability in the number of features used by ASSESS to classify issues for a combined average number of $\sim$5 features. Nevertheless, the number of features used is, with few exceptions, ≤ 7.

Table 1 summarizes the ability of ASSESS to identify specific report types as compared to all baselines. Overall classification performance was examined using macro–averaged precision and recall, which is widely accepted and commonly used for multiclass classification evaluation [27]. For reference, macro–averaged precision and recall are computed independently for each class and the results are averaged over all classes with equal weight assigned to each class. We also used micro–averaged accuracy (Acc.), which uses the cumulative number of true positives, true negatives, false positives and false negatives per type [27]. Among all baselines, Bayesian detection with top 50 features achieves

Table 1. Performance comparison of ASSESS with baselines.

	Parameters	Acc.	Precision (Avg±Std)	Recall (Avg±Std)	Avg. # feat.
ASSESS	$c = 0.20$	0.44	0.32 ± 0.12	0.42 ± 0.21	0.82
	$c = 0.17$	0.58	0.50 ± 0.12	0.56 ± 0.17	1.63
	$c = 0.15$	0.79	0.77 ± 0.05	0.80 ± 0.06	3.10
	$c = 0.10$	0.92	0.91 ± 0.01	$0.92 \pm 3 \times 10^{-3}$	4.02
	$c = 0.01$	**0.95**	$\mathbf{0.94 \pm 0.01}$	$\mathbf{0.95 \pm 3 \times 10^{-3}}$	**4.87**
	$c = 0$	0.95	0.94 ± 0.01	$0.95 \pm 3 \times 10^{-3}$	5.33
Bayesian Detection	All	0.69	0.82 ± 0.07	0.74 ± 0.06	1606
	Top 500	0.90	0.91 ± 0.03	0.92 ± 0.01	500
	Top 200	0.96	0.95 ± 0.01	$0.96 \pm 2 \times 10^{-3}$	200
	Top 100	0.96	0.95 ± 0.01	$0.96 \pm 2 \times 10^{-3}$	100
	Top 50	**0.97**	$\mathbf{0.96 \pm 3 \times 10^{-3}}$	$\mathbf{0.97 \pm 1 \times 10^{-3}}$	**50**
	Top 10	0.95	0.94 ± 0.01	$0.94 \pm 2 \times 10^{-3}$	10
	Top 5	0.85	0.86 ± 0.04	0.85 ± 0.04	5
	Top 1	0.52	0.33 ± 0.13	0.45 ± 0.21	1
ACTION [8]	One–vs–rest	0.50	0.40 ± 0.15	0.49 ± 0.21	138
	One–vs–one	0.59	0.62 ± 0.11	0.60 ± 0.15	672
SVM	SVM–L (All)	**0.97**	$\mathbf{0.96 \pm 2 \times 10^{-3}}$	$\mathbf{0.97 \pm 1 \times 10^{-3}}$	**1606**
	SVM–L (Top 5)	0.85	0.86 ± 0.03	0.85 ± 0.03	5
	SVM–G (All)	**0.97**	$\mathbf{0.96 \pm 1 \times 10^{-3}}$	$\mathbf{0.97 \pm 1 \times 10^{-3}}$	**1606**
	SVM–G (Top 5)	0.85	0.86 ± 0.03	0.85 ± 0.03	5
	SVM–FS [11]	0.92	0.92 ± 0.02	0.93 ± 0.01	6
	SVM–PCA	0.96	$0.95 \pm 4 \times 10^{-3}$	$0.96 \pm 2 \times 10^{-3}$	190
RF	d=5 (All)	0.95	$0.94 \pm 3 \times 10^{-3}$	0.95 ± 0.01	1606
	d=5 (Top 5)	0.85	0.86 ± 0.03	0.85 ± 0.03	5
	d=10 (All)	0.96	$0.96 \pm 2 \times 10^{-3}$	$0.96 \pm 1 \times 10^{-3}$	1606
	d=10 (Top 5)	0.85	0.85 ± 0.03	0.85 ± 0.03	5
XG–B	All	0.96	$0.96 \pm 3 \times 10^{-3}$	$0.96 \pm 2 \times 10^{-3}$	1606
	Top 5	0.85	0.86 ± 0.03	0.85 ± 0.03	5

the highest accuracy, precision, and recall, but requires $\sim$10 times as many features as ASSESS for a mere 2.1% improvement. Solving several binary classifications (i.e., extending ACTION [8] to multi–class classification using simplistic one–versus–the–rest and one–versus–one strategies) instead of directly considering one optimization formulation as in ASSESS results in inferior classification performance, while at the same time increasing the total number of classifiers to be trained and evaluated. Last but not least, SVM–L and SVM–G that use all features achieve the same highest accuracy, highest precision and highest recall as the Bayesian detection method that uses top 50 features, but require $\sim$ 330 times more features than ASSESS for a mere 2.1% improvement in accuracy, precision and recall.

6. CONCLUSION

In this work, the problem of automatic processing of participatory urban issue reports in civic engagement platforms was addressed. An optimization problem was defined in terms of the cost of evaluating features and the Bayes risk associated with the classification rule. A near–real–time algorithm, ASSESS, was devised that implements the optimal solution. Evaluation on a real–world dataset from the SeeClickFix civic engagement platform showed that accurate muliclass classification can be performed while reducing the number of features used by up to 99.7% compared to the state–of–the–art. In future work, we plan to devise appropriate learning–to–rank approaches to dynamically order urban issues requests.

7. REFERENCES

[1] S. A. Chun, S. Shulman, R. Sandoval, and E. Hovy, "Government 2.0: Making Connections Between Citizens, Data and Government," *Info. Pol.*, vol. 15, no. 1,2, pp. 1–9, April 2010.

[2] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," 2006.

[3] Ines Mergel, "Distributed democracy: Seeclickfix. com for crowdsourced issue reporting," 2012.

[4] A. L. Kavanaugh, E. A. Fox, S. D. Sheetz, S. Yang, L. T. Li, D. J. Shoemaker, A. Natsev, and L. Xie, "Social media use by government: From the routine to the critical," *Government Information Quarterly*, vol. 29, no. 4, pp. 480–491, 2012.

[5] D. C. Brabham, "A model for leveraging online communities," *The participatory cultures handbook*, vol. 120, 2012.

[6] Y. Sano, K. Yamaguchi, and T. Mine, "Category Estimation of Complaint Reports about City Park," in *4th International Congress on Advanced Applied Informatics*, July 2015, pp. 61–66.

[7] N. Beck, "Classification of Issues in the Public Space Using Their Textual Description and Geo-Location," .

[8] D.-S. Zois, C. Yong, C. Chelmis, A. Kapodistria, and W. Lee, "Improving Monitoring of Participatory Civil Issue Requests through Optimal Online Classification," in *52nd Asilomar Conference on Signals, Systems and Computers*, October 2018.

[9] C. Masdeval and A. Veloso, "Mining citizen emotions to estimate the urgency of urban issues," *Information systems*, vol. 54, pp. 147–155, 2015.

[10] Y. W. Liyanage, Y. Mengfan, Y. Christopher, D.-S. Zois, and C. Chelmis, "What Matters the Most? Optimal Quick Classification of Urban Issue Reports by Importance," in *6th IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, November 2018.

[11] S. Hirokawa, T. Suzuki, and T. Mine, "Machine Learning is Better Than Human to Satisfy Decision by Majority," in *International Conference on Web Intelligence*. 2017, pp. 694–701, ACM.

[12] Vladimir Vapnik, *Statistical learning theory. 1998*, vol. 3, Wiley, New York, 1998.

[13] Koby Crammer and Yoram Singer, "On the learnability and design of output codes for multiclass problems," *Machine learning*, vol. 47, no. 2-3, pp. 201–233, 2002.

[14] Mohamed Aly, "Survey on multiclass classification methods," *Neural Netw*, vol. 19, pp. 1–9, 2005.

[15] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines," *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[16] Anderson Rocha and Siome Klein Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 289–302, 2014.

[17] M. Liu, D. Zhang, S. Chen, and H. Xue, "Joint binary classifier learning for ECOC-based multi-class classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2335–2341, 2016.

[18] Manoranjan Dash and Huan Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[19] Francesco Camastra, "Data dimensionality estimation methods: a survey," *Pattern recognition*, vol. 36, no. 12, pp. 2945–2954, 2003.

[20] Girish Chandrashekar and Ferat Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[21] John P Cunningham and Zoubin Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2859–2900, 2015.

[22] A. N. Shiryaev, *Optimal Stopping Rules*, vol. 8, Springer Science & Business Media, 2007.

[23] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, Athena Scientific, 2005.

[24] H. L. Van Trees, *Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory*, John Wiley & Sons, 2004.

[25] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[26] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *ACM International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[27] Marina Sokolova and Guy Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.