IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 11, 2019, accepted July 26, 2019, date of publication July 30, 2019, date of current version August 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932018

Drifted Twitter Spam Classification Using Multiscale Detection Test on K-L Divergence

XUESONG WANG[1], QI KANG[1,2], (Senior Member, IEEE), JING AN[3], (Member, IEEE), AND MENGCHU ZHOU[4], (Fellow, IEEE)

[1]Department of Control Science and Engineering, School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China
[2]Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai 201804, China
[3]School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 201418, China
[4]Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

Corresponding authors: Qi Kang (qkang@tongji.edu.cn) and Mengchu Zhou (zhou@njit.edu)

This work was supported in part by the National Natural Science Foundation of China under Grant 51775385, Grant 61703279, and Grant 71371142, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT Twitter spam classification is a tough challenge for social media platforms and cyber security companies. Twitter spam with illegal links may evolve over time in order to deceive filtering models, causing disastrous loss to both users and the whole network. We define this distributional evolution as a concept drift scenario. To build an effective model, we adopt K–L divergence to represent spam distribution and use a multiscale drift detection test (MDDT) to localize possible drifts therein. A base classifier is then retrained based on the detection result to gain performance improvement. Comprehensive experiments show that K–L divergence has highly consistent change patterns between features when a drift occurs. Also, the MDDT is proved to be effective in improving final classification result in both accuracy, recall, and f-measure.

INDEX TERMS Concept drift, drift detection test, twitter spam classification, K-L divergence.

I. INTRODUCTION

Social media is ubiquitous nowadays, evolving its functions from personal sharing with friends to communicating with strangers of similar interests [1]. Social media platforms like Twitter therefore can exploit big data techniques to describe accurate user profiles for precision marketing [2]. Many merchants have seen this opportunity and used social media to help boost sales, among which some provide, unfortunately, bad services. They publish spam that could possibly link to unauthorized downloads and illegal commodities or even virus websites [3]. Users are unaware to click the link and suffer from information leak and financial deception. Moreover, the virus may fail the whole network and bring disastrous loss to the social media companies [4], [5].

Since social media spam can inflict catastrophic harm to the network environments, network safety corporations as well as social media platforms have dedicated themselves to identifying spam to assure user safety. The major solutions are black list systems and data-driven classification models [6]. Companies establish a black list filtering system based on manual inspection and user reports. Once a target link exists in the list, the browser automatically cuts off the connection and thus prevents further loss. The advantage of this method is stableness due to low false alarms by human verification. However, the cost to build such a system is fairly high compared to that of reproducing a new spam link. Also, when there is a report claimed from the user, the damage is unavoidable. Therefore, more and more companies turned to data-driven models aided by labor inspection to judge whether a tweet is spam.

Data-driven models use classification algorithms or anomaly detection methods to find spam among normal tweets. They benefit from low-labor costs. They can also help discover new latent features of twitter spam [7], [34]. Nevertheless, illegal merchants are building spam generating models too. They flood the filtering system with tons of spam to detect decision boundaries of normal and abnormal data. Once a bug is found, the next generation of spam can be much stealthier. This is why a twitter spam filtering model relying only on historical data would fail in the future: the twitter spam itself is evolving or as we define, has concept drifts.

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

Concepts are defined as the joint probability of data x as well as label y [8]. Concept drift means that current probability of data is different from the past. In our case, the decision boundaries of normal tweets and spam can change over time. If we use historical classifiers to predict new tweets, we will make horrible mistakes in the future since spam "knows" how to trick the models. In order to build an evolvable classification model, we need to trace spam changes and update our model accordingly so as to improve classification performance [9]. In this work, we mainly focus on tracing change, which will be introduced by two modules: concept extraction and drift detection.

Our main contribution of this work is to build a framework for detecting abrupt shifts in twitter spam series. We adopt K-L divergence to represent spam distribution and initiatively observe correlated drift patterns among twitter features including account age shift, the numbers of followers and followings. Also, we innovatively use a multiscale detection test to localize drifted time on three out of ten days and improve final classification accuracy to reach 98.86%. The rest of this paper is organized as follows. Section II reviews related work. Section III presents the proposed methods including: a detection framework, concept extraction, concept drift detection, and classification/update. Sections IV provides the experimental results. The paper is concluded in Section V.

II. RELATED WORK

A. CONCEPT EXTRACTION METHOD

The aim of concept extraction is to represent data distribution. The main extraction methods use raw features, statistical features and neural networks-based features. When raw input features are distinguishable enough, they can be directly applied to monitor a concept change. Statistical methods characterize data information through testing a proper hypothesis, e.g., some given data follow a normal distribution [10]. Neural networks can extract semantic features through layer structures without hypothesizing distribution [31], [32], but they need training processes and big data to fit parameters, which cannot be satisfied in some scenarios. Therefore, we further introduce several statistical methods.

Feature Extraction for Explicit Concept Drift Detection (FEDD) [11] computes 6 linear and 2 nonlinear statistical features to describe concepts. The linear ones include autocorrelation, variance, skewness, and kurtosis. The nonlinear ones are bicorrelation and mutual information. These 8 features are computed along each input dimension and obtain a concept vector. Then cosine or Person distances are compared among vectors at different time steps. Other distribution distance measurements involve total variation distances [29] and streaming hashing histograms [30]. Nevertheless, the concept vectors of FEDD suffer from high computational cost. Therefore, Kullback-Leibler divergence (K-L divergence), also known as relative entropy, is proposed to measure distance with lower complexity and has been widely used in anomaly detection scenarios [12], [13]. Its advantage lies in high consistency among extracted features. Hence, we adopt K-L divergence as a target extraction method.

TABLE 1. Notations and descriptions.

Notation	Description
D_t	twitter data at time t
F	norm/ spam classification model
W	time window
P, H	distribution of present and history twitter data
T, S, S_{sub}	test window, stationary and sub window
n	cardinality of the test window
T_1, T_2	Further split on the test window
r	correlation coefficient between concept features
R	average of absolute correlation coefficients on all features

B. CONCEPT DRIFT DETECTION METHOD

Detection methods are designed to find shift points in concept series. Afterwards, a classifier model can use data after the shift points to adapt itself [14]. An active approach refers to the strategy that a model is only updated when a detection method finds a drift [15]. Most of the detection algorithms are based on hypothesis tests, i.e., given $h_0$that current data has the similar distribution as the historical one, a test method validate whether h_0 holds true. Based on different $h_{0,}$ several detection algorithms are proposed [16], [37].

Page-Hinkley test (PH-test) presumes that mean values of current concepts should be close to historical ones [17]. It cumulates difference between the observed values and historical ones. If the minimum of such difference exceeds a threshold, current moment is claimed as drift time. Cumulative Sum (CUSUM) hypothesizes that stationary concepts should fluctuate within a small range [18]. A cumulative sum variable is built. It should be near zero when there is no drift since negative and positive small values offset each other. A drift is found when such variable explodes to reach a predefined bound.

Based on a resampling scheme and a paired student t-test, we have proposed a multiscale drift detection test (MDDT) that localizes abrupt drift points when a concept changes [19]. It applies a detection procedure on two different scales. Initially, the detection is performed on a broad scale to check if recently gathered drift indicators remain stationary. If a drift is claimed, a narrow scale detection is performed to trace the refined change time. This multiscale structure reduces massive time of constant checking and filters noises successfully. Hence, we use MDDT as a final drift detector in this work. Such application is never seen to the best knowledge of the authors.

III. DRIFTED TWITTER SPAM CLASSIFICATION

In this section, we present a drifted twitter spam classification method based on multiscale drift detection test (MDDT) [19]. The main idea is to detect distributional change and use drifted data to update the classification model. The notations frequently used in this paper are summarized in Table 1.

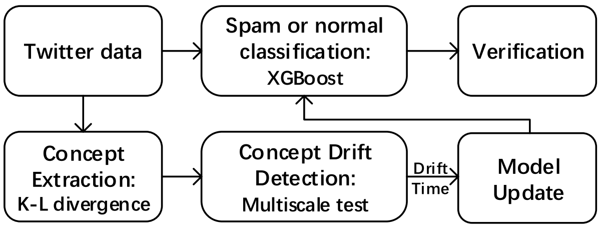


FIGURE 1. Framework for drifted twitter spam classification.

The concrete steps of the method are detailed in the following sections.

A. PROPOSED METHOD

1) FRAMEWORK

The framework of our method is given in Algorithm 1 and Fig. 1. First, we train a binary classification model on tweets to decide whether they are spam or normal. Meanwhile, a concept extractor is computed using K-L divergence which measures distributional distance among different samples. The purpose is to describe the difference between the present data distribution and historical one and leave the adaptation task to a base classifier. Then, MDDT is adopted to check whether current data concepts differ from historical ones and if so, claims the drift time. Afterwards, drifted data after that time are utilized to update the model to enhance robustness. Finally, further data are input to verify performance improvement.

Algorithm 1 Framework of Drifted Twitter Spam Classification

Input: Twitter data in time: $D = \{D_1, D_2, \ldots, D_t\}$
Output: classification model F

1. Initialization: train a classifier F on D_1, time window $W = \emptyset$
2. For $t = 2, 3, \ldots$
3. Compute K-L divergence between D_t and D_1:$D_{KL}(D_t||D_1)$ and add it to W
4. Multiscale drift detection test on W to see if there are drifts
5. **If** True
6. Retrain F with data after the drift point, $W = \emptyset$, $D_1 = D_t$
7. **End If**
8. Verify F with D_t
9. End For

2) CONCEPT EXTRACTION: K-L DIVERGENCE

Concept extraction is aiming at representing data distribution. When drift occurs, it changes correspondingly such that drift detection algorithms can easily find outliers therein. In our case, K-L divergence is chosen as a measure for similarity or asymmetry between two distributions. It is calculated as

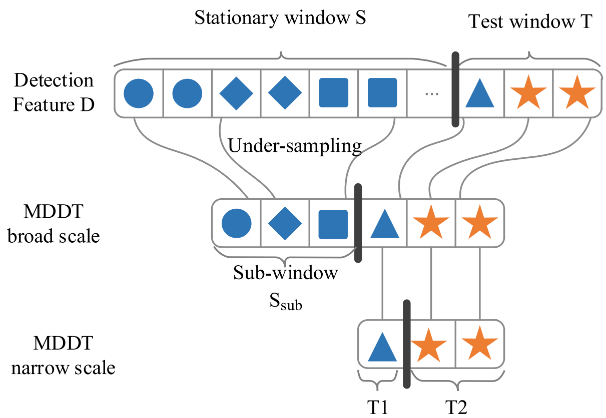


FIGURE 2. Framework of MDDT.

$$D_{KL}(P \parallel H) = -\sum\nolimits_{i=1}^{K} P_i ln\frac{H_i}{P_i} = \sum\nolimits_{i=1}^{K} P_i ln\frac{P_i}{H_i} \quad (1)$$

where P and H represent two 1-D distributions of categorical variables, $P_i = P(x|x = i)$ and K is the set of all possible outcomes. In our case, P and H are present and historical twitter data distribution. If they are identical, their divergence should be small since $ln\,(P_i/H_i) \approx 0$. In case $H_i = 0$ when i only occurs in P, we revise P_i and H_i by P_i' and H_i' as suggested in [20]

$$P_i^{'} = 0.66\,(P_i + 0.5)\,, \quad H_i^{'} = 0.66(H_i + 0.5) \quad (2)$$

Equation (2) is always applied for consistency. For numerical variables, K-L divergence can be approximated by splitting inputs into categories. For multi-dimensional variables we compute it along each dimension and use cosine distance to aggregate total difference, i.e.,

$$d_{cos}\,(P, H) = 1 - \frac{< P, H >}{\|P\|\,\|H\|} \quad (3)$$

where $< P, H >$ is the inner product of two vectors and $\|\cdot\|$ is the l_2-norm of a vector.

Now we are able to evaluate difference of the present spam and historical one. If the result is huge, then there is a drift in the time window W. However, to evaluate difference with ‘‘huge’’ or ‘‘small’’ is sometimes blurry and can claim false alarm drifts. Therefore, we need an accurate checking method to find reliable drift points as to be discussed next.

3) CONCEPT DRIFT DETECTION: MULTISCALE TEST

We utilize Multiscale Drift Detection Test (MDDT) [19] to localize drift points in a time window W. It is described in Algorithm 2 and Fig. 2.

Suppose that a stationary environment changes at a certain point t $*$ (unknown in advance). Then the latest detection features in a test window T shall be significantly different from a sub-window S_{sub} picking features from the past. More specifically, first we want to check out whether current features are drifted. If so, can they be further purified to leave only drifted features? Our main contribution is that we do not need to examine each and every feature in T. Instead we apply a further drift detection on the split of T, i.e., T_1 and T_2. If they are significantly different, then the split point between T_1 and T_2 is supposed to be the drift point.

The reason why this split works is that we build a t-test statistic for T_1 and T_2 when they are significantly different. Then based on relationship that $T = T_1 + T_2$, we find a condition to satisfy a new statistic representing significant difference between T and stationary window S, i.e., $|T_1| = \frac{n}{1+\left(\frac{t_\alpha(n-2)}{t_\alpha(2n-2)}\right)^2 \frac{n-4}{n-2}}$ in Algorithm 2.

$|T|$ is the cardinality of a window T, $t_\alpha(n)$ is the α quantile of t-distribution with n degrees of freedom. MDDT tries to select the latest samples to formulate a test window T (step 1) and check if they are significantly different from the past (steps 2-3). We adopt paired t-test (**Theorem 1**) to evaluate difference significance. If positive, can T be further split so as to find an accurate segment between drifted spam and historical one (steps 5-6)? If so, MDDT claims a drift point $t*$.

Algorithm 2 [19] Multiscale Drift Detection Test (MDDT)

Input: time window W
Output: drift time $t*$

1.Split W into stationary window S and test window T, $n = |T|$, $|S| >> |T|$
2.Undersampling S to get sub-window S_{sub}, $|S_{sub}| = |T|$
3. t-test on S_{sub} and T to see if they are significantly different
4. **If** True
5. Further split T into T_1 and T_2, $|T_1| = \frac{n}{1+\left(\frac{t_\alpha(n-2)}{t_\alpha(2n-2)}\right)^2 \frac{n-4}{n-2}}$,
6. t-test on T_1 and T_2 to see if they are significantly different
7. **If** True
8. $t* =$ time at T_1& T_2's boundary
9. **End If**
10. **End If**

The central limit theorem (CLT) establishes that, when independent random variables are averaged, the distribution of the mean is closely approximated by a normal distribution, even if the original variables themselves are not normally distributed. Hence, we can use the mean values of independent KL divergence for a paired t-test.

Theorem 1 (paired t-test): *Let $S_1, S_2, \ldots,$ and S_{n1}, and $T_1, T_2, \ldots,$ and T_{n2} be two independent samples satisfying $S \sim N(\mu_1, \sigma^2)$ and $T \sim N(\mu_2, \sigma^2)$. $\bar{S}$ and $\bar{T}$ denote their sample means and* σ_S^2 and σ_T^2 *are sample variances. Given hypothesis $H_0 : \mu_1 - \mu_2 \le \delta$ and a confidence level α, the statistic t obeys a student distribution:*

$$t = \frac{\bar{S} - \bar{T} - (\mu_1 - \mu_2 - \delta)}{\sigma_w \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t(n_1 + n_2 - 2) \quad (4)$$

$$\sigma_w^2 = \frac{(n_1 - 1)\sigma_S^2 + (n_2 - 1)\sigma_T^2}{n_1 + n_2 - 2} \quad (5)$$

When t lies within the rejected region, i.e., $t \ge t_\alpha(n_1 + n_2 - 2)$, we accept $\mu_1 - \mu_2 > \delta$ and assert significant difference between S and T.

The final output of MDDT is the drift time. The next section introduces how to adapt a classification model to improve performance.

4) CLASSIFICATION MODEL AND UPDATING

After comparing with KNN and SVM models, we choose random forest (RF) as a spam/normal classification model. A random forest is an ensemble of sub decision tree classifiers. Training data are split for building different sub trees. A sub tree calculates the Gini coefficient of a subset of all features and recursively builds a binary classifier on the feature with the smallest coefficient [33]. The ensemble using data and feature split not only increases diversity on a data level, but also on a feature level, which balances well between bias and variance. It turns out that twitter spam data has high intra-class variance. Hence, the mechanism of random forests to use sub-features can learn different sub tress for intra-class samples and is therefore suitable for our case. A revised version of a forest called XGBoost in order is adopted to further improve performance. Later experimental results show that random forest outperforms other base learners like SVM and KNN [21]–[24]. As for model updating, we simply retrain a new classifier with data after the claimed drift time from MDDT.

IV. EXPERIMENTS AND RESULTS

In this section, experiments are detailed to test the proposed method on an open source drifted twitter spam dataset. Several criteria are used to evaluate concept extraction, claimed drift points and classification performance. Experiments are performed on 2.60 GHz Core i5-3230M machines with 12 GB of memory. The simulation environment includes Python 2.7. All base classifiers are built by using open-source scikit-learn package.

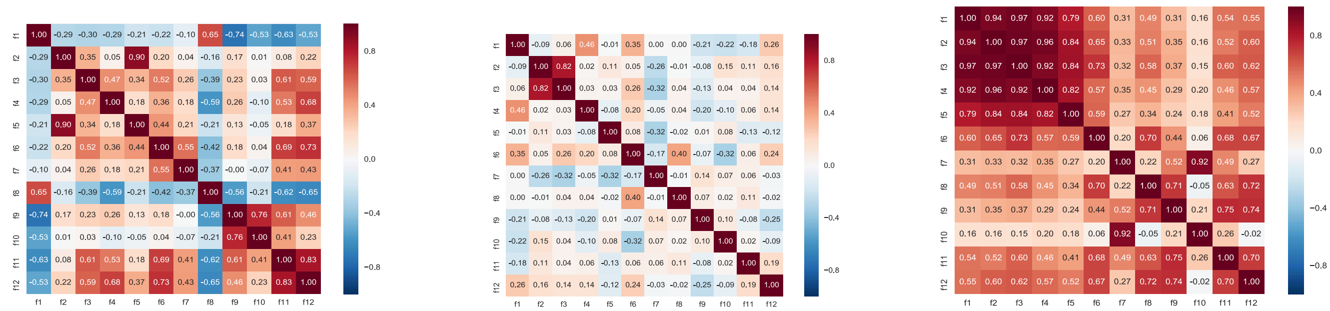
A. DATASET: DRIFTED TWITTER SPAM

We use the public dataset from [6]. It collects 12 features that are directly accessible through Twitter API (Table 2). Only tweets with URL are selected, whether they are spam or normal data are verified by Web Reputation Technology from Trend Micro. According to AV Comparatives' testing report, the protection rate of the WRT system is 100%. 10,000 per day of total 10 day records are used. The spam rate is set to be 5% to mimic real world scenarios. More details of the dataset can be seen in [36].

B. COMPARING METRICS

1) CONCEPT EXTRACTION

Raw input, FEDD and KL-divergence are chosen as concept extractors to be compared. Each extractor calculates a



(a) raw input (b) FEDD (c) K-L divergence

FIGURE 3. Correlation coefficients of three concept extraction methods.

TABLE 2. Drifted twitter spam dataset (no means number).

feature index	description	feature index	description
f1	account age	f7	no_retweets
f2	no_follower	f8	no_hashtag
f3	no_following	f9	no_usermention
f4	no_userfavourate	f10	no_urls
f5	no_lists	f11	no_char
f6	no_tweets	f12	no_digits

concept vector with dimension equal to 12 (raw dimension). Correlation coefficients are computed among 12 features:

$$r_{i,j} = \frac{\sum_{day=2}^{10}\left(f_i^{(day)} - \bar{f}_I\right)\left(f_j^{(day)} - \bar{f}_J\right)}{\sqrt{\sum_{day=2}^{10}\left(f_i^{(day)} - \bar{f}_I\right)^2}\sqrt{\sum_{day=2}^{10}\left(f_j^{(day)} - \bar{f}_J\right)^2}} \tag{6}$$

$$R = \frac{1}{12 \times 12}\sum_{i=1}^{12}\sum_{j=1}^{12}|r_{i,j}| \tag{7}$$

High R values means that this extraction method obtains consistent concept features, which is good because if a drift occurs, every feature value is expected to fluctuate accordingly. Otherwise, if some features shift while others not, we cannot decide whether it is a real shift or just noise on certain features.

2) MODEL UPDATE PERFORMANCE

We evaluate classification performance on different methods. They are categorized as: RF/KNN/SVM/XGB- based methods. In each set five methods are compared

[a] X ∈ {RF, KNN, SVM, XGB}
[b] $X^{\#}$
[c] MDDT + X
[d] CUSUM + X
[e] PH + X

where X can be a base learner, e.g., KNN (nearest neighbor k = 5), RF, and SVM (penalty coefficient C = 1.5, kernel = RBF with balanced reweighting for each class). The tolerance factor δ of the PH test is set to be 0.005. The change detection threshold λ of PH test is set to be 50. The max depth of an XGB tree is 50. We use cross-validation to select the appropriate parameters of the above methods. X is trained only once on the first day. $X^{\#}$ represents an X classifier that is constantly retrained based on the last-day data. Methods c, d and e mean that X is retrained only after detectors claim a drift point.

Experiment (a) is to find the optimal one among tested base classifiers for the problem. Experiment (b) is to test whether constant update can enhance performance. Experiments (c) -(d) are used to compare different drift detection methods. Besides accuracy, other metrics like recall and F-measure for imbalanced classification are used to evaluate performance. The confusion matrix is defined as follows:

	Spam	Normal
Predicted Spam	TP (True Positive)	FP (False Positive)
Predicted Normal	FN (False Negative)	TN (True Negative)

Then,

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

C. RESULTS AND ANALYSIS

1) CONCEPT EXTRACTION

The heat map plots of correlation coefficients are displayed in Fig. 3 and the results of absolute average over coefficients are given in Table 3.

K-L divergence extraction achieves the highest score of 0.55 and has the most correlated features in heat maps. In Fig. 3, after K-L representation, f1-f3 are found to be

TABLE 3. Average over absolute correlation coefficients.

Extraction method	Raw input	FEDD	KL-divergence
R	0.41	0.20	**0.55**

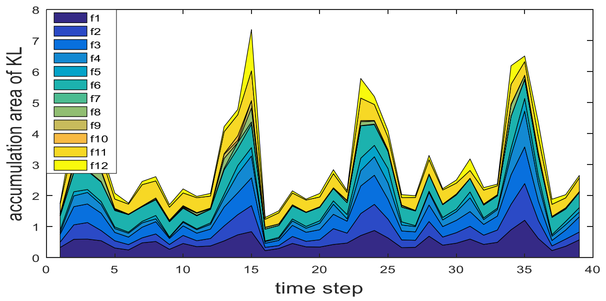


FIGURE 4. Accumulation of K-L divergence for twelve features.

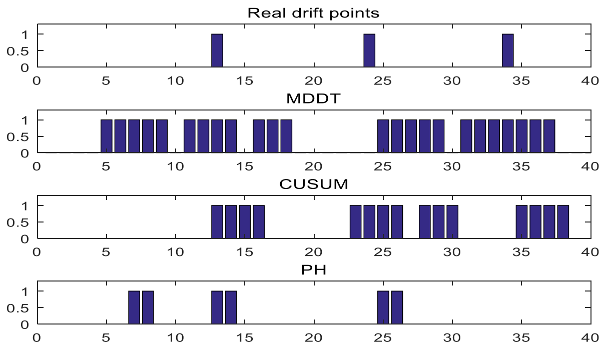


FIGURE 5. Claimed drift intervals of detection methods.

most correlated with each other, which means that when the account age shift, the numbers of followers and followings are very likely to be drifted together. Also, f7 is highly consistent with f10, indicating that fluctuations in the number of URL attached can directly affect total retweets. Overall, when there is a drift, every feature exhibits various degree of change. Hence, we choose K-L divergence as a concept extractor.

The accumulation of K-L divergence for 12 features are illustrated in Fig. 4. The total time step count is 40 instead of 10 since we split everyday data into 4 even parts. This generates more concept vectors that help better display distributional shifts. Most of features have a similar trend and the overall trend peaks at time step 13, 24 and 34, i.e., day 4, 6 and 9. Therefore, we use 13, 24 and 34 as real drift points to evaluate concept drift detection algorithms. The detected intervals are plotted in Fig. 5: MDDT and CUSUM catch all drift points, PH has one missing point at time step 34. MDDT has 1-time-step latency on position 24 but still catches it. The time step MDDT discovers a drift is the 29$^{\text{th}}$ one, the time step it localizes the drift is the 25$^{\text{th}}$ one. Hence, MDDT successfully catches all drift points. MDDT claims one more false alarm point than CUSUM. It regards a small

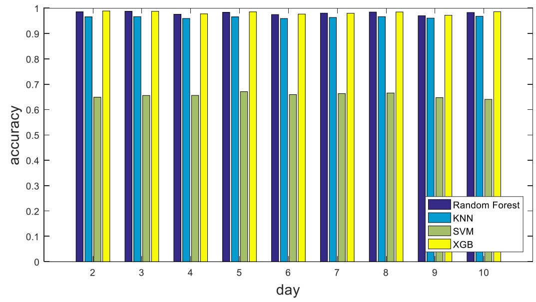


FIGURE 6. Accuracy of three methods on nine days.

fluctuation on the interval [5], [10] as a drift. This implies that CUSUM is more suitable for detecting severe abrupt drifts whereas MDDT can also detect non-severe drifts. A drift detection method belongs to data preprocessing and needs to be computationally efficient so as to save time for classifier training. Hence, we do not aggregate the results of the three methods. Later classification comparisons show that more sensitive adaption is necessary and helpful for dealing with drifts in this case.

2) MODEL UPDATE PERFORMANCE

The results of experiment (a) are illustrated in Fig. 6. We use data from day 1 as a training set and predict spam labels for the next 9 days. XGBoost (XGB) achieves the highest accuracy compared to RF, KNN and SVM. The average value is 98.19% whereas the same metrics for RF, KNN and SVM are 98.03% 96.36%, and 65.59%. Low performance of SVM is attributed to a balanced reweighting process, which is aimed to successfully classify more spam data but leads to more errors in normal data. Hence, the recall score of SVM (0.87) is higher than that of KNN (0.51), RF (0.63) and XGB (0.69). As mentioned earlier, spam data has high intra-class variations. A spam data with no similar neighbors can still be a valid point yet SVM might ignore it to avoid overfitting. Hence, algorithms that tend to overfit data like XGB, RF can predict spams well in such scenario.

In order to explore whether continuous retraining can outperform never-adapting models, experiment (b) is added. Also, we compare concept drift detection methods including MDDT, CUSUM and PH-test to validate whether they can keep track of drifts and adapt models accordingly via Experiments (c)-(e). Accuracy, recall and F-measure results are displayed in Tables 4-6. Boxplot performances of overall algorithms are given in Figs. 7-10.

From Tables 4-7 column (b) we can conclude that $X^{\#}$ competes against all other tested methods on or close to drift days (4, 6 and 9). This means that compared to non-adaptive classifiers, constantly adapting ones can respond more quickly and gain improvement right after the drifts. However, their average metrics are lower than MDDT-based methods' (Figs. 7-10), indicating that improper updating can possibly lead to unstable performance.

TABLE 4. Classification performances of five RF - based algorithms on nine days.

Day	Accuracy(%)					Recall					F-measure				
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
2	98.5	98.6	**98.7**	98.6	98.6	0.75	0.74	0.75	0.75	**0.77**	0.83	0.84	0.85	0.84	**0.85**
3	98.7	98.5	**99.5**	98.6	98.2	0.76	0.71	**0.90**	0.75	0.66	0.86	0.83	**0.95**	0.84	0.78
4	97.6	**97.9**	97.4	97.6	96.7	0.54	**0.59**	0.50	0.56	0.37	0.69	**0.74**	0.66	0.70	0.53
5	98.3	**98.7**	98.6	98.6	95.3	0.71	**0.77**	0.73	0.76	0.07	0.81	**0.86**	0.84	0.85	0.13
6	97.5	98.1	**98.5**	97.7	95.5	0.54	0.64	**0.70**	0.58	0.13	0.68	0.77	**0.82**	0.72	0.22
7	98.0	**98.1**	97.7	95.4	94.9	0.63	**0.65**	0.57	0.13	0.04	0.76	**0.78**	0.71	0.22	0.08
8	98.4	98.5	**99.5**	98.0	98.2	0.71	0.72	**0.91**	0.63	0.67	0.82	0.83	**0.95**	0.76	0.79
9	97.0	**97.3**	96.8	96.7	96.8	0.42	**0.48**	0.37	0.35	0.37	0.58	**0.64**	0.54	0.51	0.54
10	98.3	98.8	99.2	**99.3**	98.2	0.68	0.78	0.84	**0.86**	0.66	0.80	0.87	0.91	**0.92**	0.79

* Notation: a. RF, b. RF#, c. MDDT+RF, d. CUSUM + RF, e. PH +RF

TABLE 5. Classification performances of five KNN - based algorithms on nine days.

Day	Accuracy(%)					Recall					F-measure				
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
2	96.5	96.5	96.5	96.5	96.5	0.56	0.56	0.56	0.56	0.56	0.62	0.62	0.62	0.62	0.62
3	96.6	96.6	**97.4**	96.6	96.2	0.56	0.59	**0.68**	0.56	0.55	0.62	0.63	**0.72**	0.62	0.59
4	**95.9**	95.8	95.5	95.9	94.8	0.39	**0.43**	0.41	0.39	0.27	0.49	**0.51**	0.48	0.49	0.34
5	96.5	96.0	**96.9**	96.0	93.6	0.58	0.54	**0.69**	0.54	0.06	0.63	0.57	**0.69**	0.57	0.09
6	95.9	96.2	**96.8**	95.4	93.5	0.44	0.52	**0.60**	0.41	0.09	0.51	0.58	**0.65**	0.47	0.12
7	96.3	**96.8**	96.4	94.7	92.1	0.49	**0.63**	0.55	0.12	0.06	0.57	**0.66**	0.60	0.19	0.07
8	96.6	96.9	**97.7**	96.3	96.3	0.58	0.72	**0.77**	0.59	0.59	0.63	0.70	**0.77**	0.62	0.62
9	96.1	**96.9**	96.4	96.0	96.1	0.43	**0.63**	0.56	0.47	0.47	0.52	**0.67**	0.61	0.54	0.55
10	96.8	96.4	97.5	**97.5**	96.6	0.60	0.59	0.72	**0.73**	0.61	0.65	0.62	0.74	**0.75**	0.64

* Notation: a. KNN, b. KNN#, c. MDDT+KNN, d. CUSUM + KNN, e. PH +KNN

TABLE 6. Classification performances of five SVM - based algorithms on nine days.

Day	Accuracy(%)					Recall					F-measure				
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
2	64.8	64.9	64.8	64.8	64.8	0.89	0.89	0.89	0.89	0.89	0.20	0.20	0.20	0.20	0.20
3	65.5	65.2	**68.6**	65.5	66.9	0.81	0.81	**0.82**	0.81	0.75	0.19	0.19	**0.21**	0.19	0.19
4	65.6	**74.6**	68.6	65.6	66.6	0.74	0.72	**0.77**	0.74	0.60	0.18	**0.22**	0.20	0.18	0.15
5	67.0	63.9	**75.3**	61.3	48.8	0.94	0.95	0.85	**0.96**	0.60	0.22	0.21	**0.26**	0.20	0.11
6	65.9	**70.4**	67.8	59.8	48.4	0.85	0.82	0.83	**0.91**	0.54	0.2	**0.22**	0.21	0.18	0.09
7	66.3	64.8	**67.8**	63.6	48.8	0.91	**0.94**	0.89	0.91	0.57	0.21	0.21	**0.22**	0.20	0.10
8	66.5	65.8	**71.1**	60.7	60.7	0.93	**0.96**	0.91	0.94	0.94	0.22	0.22	**0.24**	0.19	0.19
9	64.7	**71.5**	70.0	68.1	59.0	0.86	0.85	0.85	0.82	**0.89**	0.20	**0.23**	0.22	0.21	0.18
10	64.0	60.7	**69.4**	66.7	56.6	0.94	0.96	0.90	0.94	**0.97**	0.21	0.20	**0.23**	0.22	0.18

* Notation: a. SVM, b. SVM#, c. MDDT+SVM, d. CUSUM + SVM, e. PH +SVM

From Figs. 7-10 we can see that MDDT is the winner among all the tested algorithms. Its best average values of accuracy, recall and F-measure on an XGB-based classifier are 98.86%, 0.80 and 0.87 respectively, with recall and F-measure being 0.14 higher than CUSUM's and 0.3 higher than PH-test's. The outlier in Fig 8(b) is on day 7, which is much lower than the rest of days. From the K-L divergence (Fig. 4) we know that there are two fluctuations in the interval [25], [30] (day 7). Hence the model of CUSUM updates too early on the first distribution and fails to fit the later one. Also, the performance of MDDT is more stable than those of CUSUM and PH-test with only one outlier (Red

TABLE 7. Classification performances of five XGBoost - based algorithms on nine days.

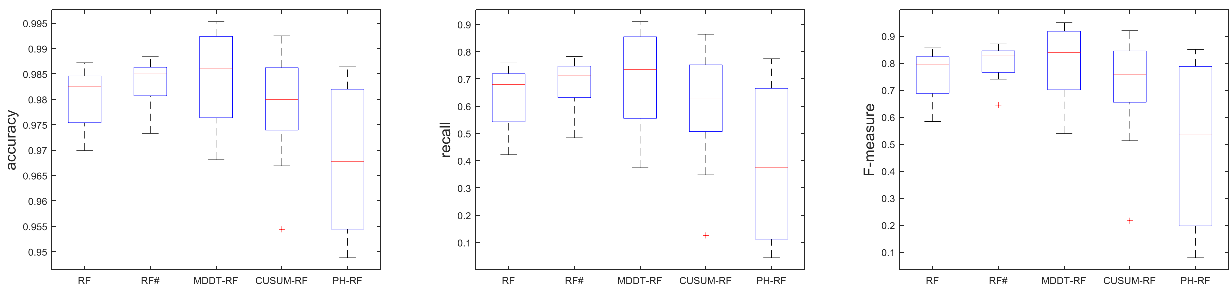
Day	Accuracy(%)					Recall					F-measure				
	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
2	98.9	98.9	98.9	98.9	98.9	0.81	0.81	0.81	0.81	0.81	0.88	0.88	0.88	0.88	0.88
3	98.7	98.8	**99.4**	98.7	98.6	0.80	0.78	**0.90**	0.80	0.75	0.86	0.86	**0.94**	0.86	0.84
4	97.7	98.1	**99.0**	97.7	97.2	0.59	0.65	**0.82**	0.59	0.49	0.72	0.77	**0.89**	0.72	0.64
5	98.5	**98.9**	98.8	98.8	95.2	0.76	**0.84**	0.78	0.79	0.10	0.84	**0.89**	0.86	0.87	0.17
6	97.6	**98.3**	97.5	98.0	95.6	0.60	**0.69**	0.56	0.64	0.19	0.72	**0.80**	0.69	0.76	0.30
7	98.0	**98.3**	97.8	95.5	95.0	0.64	**0.69**	0.60	0.17	0.06	0.76	**0.80**	0.73	0.28	0.11
8	98.5	98.8	**99.0**	98.4	98.4	0.75	0.80	**0.82**	0.74	0.74	0.83	0.87	**0.89**	0.82	0.82
9	97.2	98.2	**100**	97.0	97.2	0.48	0.67	**1.00**	0.41	0.49	0.63	0.79	**1.00**	0.58	0.64
10	98.6	99.1	99.3	**99.4**	98.5	0.77	0.85	0.89	**0.89**	0.74	0.84	0.90	0.93	**0.93**	0.83

* Notation: a. XGB, b. XGB#, c. MDDT+XGB, d. CUSUM + XGB, e. PH +XGB

TABLE 8. Wilcoxon test results of five comparing methods.

Classifier	Metric \ Method	MDDT-X vs X	MDDT-X vs $X^{\#}$	MDDT-X vs CUSUM-X	MDDT-X vs PH-X
X=RF	accuracy	0.096	0.633	0.091	**0.012**
	recall	0.207	0.767	0.183	**0.017**
	F-measure	0.192	0.767	0.108	**0.018**
X=KNN	accuracy	**0.042**	0.092	**0.034**	**0.012**
	recall	**0.012**	0.183	**0.017**	**0.012**
	F-measure	**0.017**	0.092	**0.035**	**0.012**
X=SVM	accuracy	**0.012**	0.26	**0.012**	**0.012**
	recall	0.139	0.233	0.205	0.107
	F-measure	**0.01**	0.203	**0.011**	**0.011**
X=XGB	accuracy	**0.035**	0.326	0.063	**0.012**
	recall	0.069	0.484	0.123	**0.012**
	F-measure	**0.036**	0.327	0.093	**0.012**

* Notation: p-values smaller than 0.05 indicates MDDT-X is significantly better.

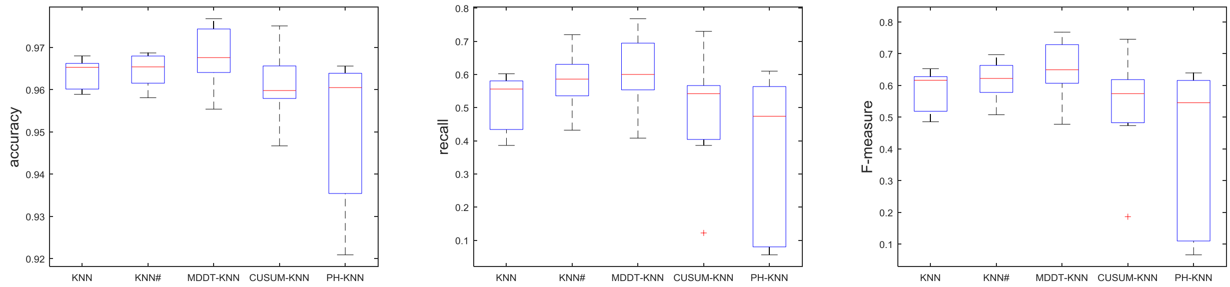


(a) accuracy (b) recall (c) f-measure

FIGURE 7. Boxplot classification performance of five RF-based algorithms.

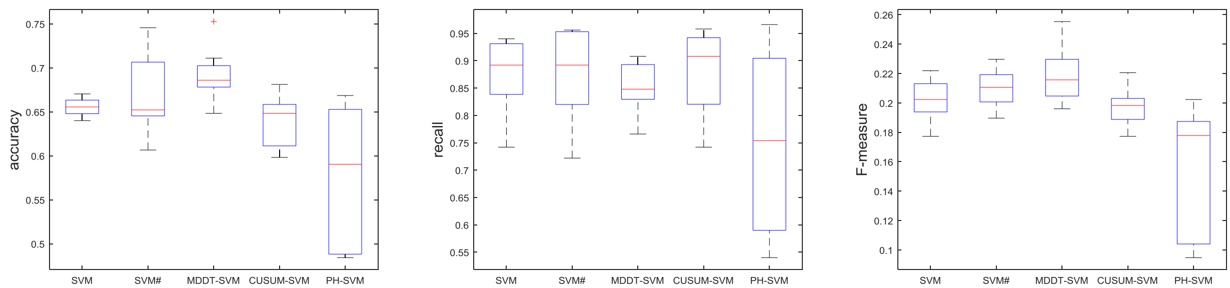
cross in Fig. 9(a)) and lower variations (box range in the figures), which satisfies the need of practical use of twitter spam classification.

We have applied Wilcoxon's test on all the methods. The results are given in Table 8. Among different drift detection algorithms, MDDT are significantly better than PH on almost every metric. Also, MDDT outperforms X and CUSUM-X on KNN-based methods. MDDT is not significantly better than most of RF-based methods because the original performance of RF is already high. $X^{\#}$ is designed to become the upper bound for all drift detection methods like MDDT since intuitively a model that keeps updating is supposed to fit better on



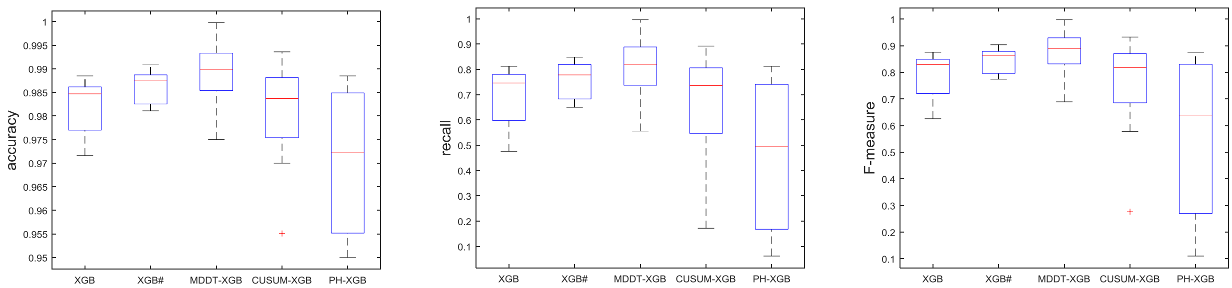
(a) accuracy (b) recall (c) f-measure

FIGURE 8. Boxplot classification performance of five KNN-based algorithms.



(a) accuracy (b) recall (c) f-measure

FIGURE 9. Boxplot classification performance of five SVM-based algorithms.



(a) accuracy (b) recall (c) f-measure

FIGURE 10. Boxplot classification performance of five XGB-based algorithms.

new data distribution. In practice, however, if the model only updates on the drift time like MDDT, the average accuracy could be slightly higher than X#. Although such strength is not obvious, considering that MDDT does not lose accuracy, it has lower time cost on model retraining and is therefore more competitive than other drift detection methods.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a drifted twitter spam classification method by using multiscale drift detection test (MDDT) on K-L divergence. K-L divergence is used as a concept extractor to represent spam distributional change, while MDDT localizes shift points in the divergence sequence. Once a drift is found, a base classifier using XGBoost is called. The results reveal that K-L divergence has highly consistent change patterns among features when a drift occurs. Also, MDDT improves final classification accuracy to achieve 98.86% and well outperforms state-of-art drift detection algorithms, which is significant in this field.

In the future, we plan to exploit artificial neural networks [25]–[28], [38] and imbalanced classification methods [39], [40] to blend concept extraction and model adaptation, which may enable us to explore concept drift in a coupled feature space [35]. Also, we plan to build sub-trees for new concepts in RF or XGB to have lower cost of model retraining and knowledge forgetting.

ACKNOWLEDGMENT

The authors would like to thank the help of Mr. Zibo Yu, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215123, China, in typesetting this paper.

REFERENCES

[1] A. Kumar, R. Bezawada, R. Rishika, R. Janakiraman, and P. K. Kannan, ''From social to sale: The effects of firm-generated content in social media on customer behavior,'' *J. Marketing*, vol. 80, no. 1, pp. 7–25, 2016.

[2] J. A. Aloysius, H. Hoehle, S. Goodarzi, and V. Venkatesh, ''Big data initiatives in retail environments: Linking service process perceptions to shopping outcomes,'' *Ann. Oper. Res.*, vol. 270, nos. 1–2, pp. 25–51, Nov. 2018.

[3] S. Sedhai and A. Sun, ''Semi-supervised spam detection in Twitter stream,'' *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 1, pp. 169–175, Mar. 2018.

[4] B. Liu, Z. Ni, J. Luo, J. Cao, X. Ni, B. Liu, and X. Fu, ''Analysis of and defense against crowd-retweeting based spam in social networks,'' *World Wide Web*, vol. 21, pp. 1–23, 2018. doi: 10.1007/s11280-018-0613-y.

[5] M. Stamp, *Introduction to Machine Learning With Applications in Information Security*. London, U.K.: Chapman & Hall, 2017.

[6] C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, and G. Min, ''Statistical features-based real-time detection of drifted Twitter spam,'' *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 914–925, Apr. 2017.

[7] S. Wang, Z. Ding, and Y. Fu, ''Feature selection guided auto-encoder,'' in *Proc. 31st AAAI Conf. Artif.*, San Francisco, CA, USA, Feb. 2017, pp. 2725–2731.

[8] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, ''A survey on concept drift adaptation,'' *ACM Comput. Surv.*, vol. 46, no. 4, pp. 44:1–44:37, Apr. 2014.

[9] C. Alippi, G. Boracchi, and M. Roveri, ''Hierarchical change-detection tests,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 246–258, Feb. 2016.

[10] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, ''Learning with drift detection,'' in *Proc. Brazilian Symp. Artif. Intell.*, 2004, pp. 286–295.

[11] R. C. Cavalcante, L. L. Minku, and A. L. I. Oliveira, ''FEDD: Feature extraction for explicit concept drift detection in time series,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 740–747.

[12] S. Yu, X. Wang, and J. C. Principe, ''Request-and-reverify: Hierarchical hypothesis testing for concept drift detection with expensive labels,'' in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, Jul. 2018, pp. 3033–3039.

[13] S. Schmidt and P. S. Heyns, ''Localised gear anomaly detection without historical data for reference density estimation,'' *Mech. Syst. Signal Process.*, vol. 121, pp. 615–635, Apr. 2019.

[14] J. Zhang, Z. Wei, Z. Yan, M. Zhou, and A. Pani, ''Online change-point detection in sparse time series with application to online advertising,'' *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 6, pp. 1141–1151, Jun. 2019. doi: 10.1109/TSMC.2017.2738151.

[15] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, ''Learning in nonstationary environments: A survey,'' *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 12–25, Apr. 2015.

[16] I. Frías-Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, and Y. Caballero-Mota, ''Online and non-parametric drift detection methods based on Hoeffding's bounds,'' *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 810–823, Mar. 2015.

[17] J. Gama, R. Sebastião, and P. P. Rodrigues, ''On evaluating stream learning algorithms,'' *Mach. Learn.*, vol. 90, no. 3, pp. 317–346, 2013.

[18] O. A. Grigg, V. T. Farewell, and D. J. Spiegelhalter, ''Use of risk-adjusted CUSUM and RSPRTcharts for monitoring in medical contexts,'' *Stat. Methods Med. Res.*, vol. 12, no. 2, pp. 147–170, 2003.

[19] X. Wang, Q. Kang, M. C. Zhou, and S. Yao, ''A multiscale concept drift detection method for learning from data streams,'' in *Proc. IEEE 14th Int. Conf. Automat. Sci. Eng.*, Munich, Germany, Aug. 2018, pp. 786–790.

[20] R. Krichevsky and V. Trofimov, ''The performance of universal encoding,'' *IEEE Trans. Inf. Theory*, vol. 27, no. 2, pp. 199–207, Mar. 1981.

[21] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, ''A comparison of machine learning techniques for phishing detection,'' in *Proc. Anti-Phishing Working Groups 2nd Annu. eCrime Res. Summit*, Oct. 2007, pp. 60–69.

[22] C. Vens and F. Costa, ''Random forest based feature induction,'' in *Proc. IEEE 11th Int. Conf. Data Mining*, Vancouver, BC, Canada, Dec. 2011, pp. 744–753.

[23] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse, ''An empirical study of learning from imbalanced data using random forest,'' in *Proc. 19th IEEE Int. Conf. Tools Artif. Intell.*, Oct. 2007, pp. 310–317.

[24] R. Khan, A. Hanbury, and J. Stoettinger, ''Skin detection: A random forest approach,'' in *Proc. IEEE Int. Conf. Image Process.*, Hong Kong, Sep. 2010, pp. 4613–4616.

[25] D. Yu and J. Li, ''Recent progresses in deep learning based acoustic models,'' *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 3, pp. 396–409, Jul. 2017.

[26] Q. Kang, B. Huang, and M. Zhou, ''Dynamic behavior of artificial Hodgkin–Huxley neuron model subject to additive noise,'' *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2083–2093, Sep. 2016.

[27] C. Li, L. Wang, G. Zhang, H. Wang, and F. Shang, ''Functional-type single-input-rule-modules connected neural fuzzy system for wind speed prediction,'' *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 751–762, Apr. 2017.

[28] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, ''Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.

[29] G. I. Webb, L. K. Lee, F. Petitjean, and B. Goethals, ''Understanding concept drift,'' Apr. 2017, *arXiv:1704.00362*. [Online]. Available: https://arxiv.org/abs/1704.00362

[30] D. Yang, B. Li, L. Rettig, and P. Cudré-Mauroux, ''Histosketch: Fast similarity-preserving sketching of streaming histograms with concept drift,'' in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, New Orleans, LA, USA, Nov. 2017, pp. 545–554.

[31] C. Lee, Y.-B. Kim, D. Lee, and H. Lim, ''Character-level feature extraction with densely connected networks,'' 2018, *arXiv:1806.09089*. [Online]. Available: https://arxiv.org/abs/1806.09089

[32] M. Jaworski, P. Duda, and L. Rutkowski, ''Concept drift detection in streams of labelled data using the restricted Boltzmann machine,'' in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–7.

[33] H. Liu, M. Zhou, and Q. Liu, ''An embedded feature selection method for imbalanced data classification,'' *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 703–715, May 2019.

[34] M. Alrubaian, M. Al-Qurishi, A. Alamri, M. Al-Rakhami, M. M. Hassan, and G. Fortino, ''Credibility in online social networks: A survey,'' *IEEE Access*, vol. 7, pp. 2828–2855, 2019.

[35] T. Wu, S. Wen, S. Liu, J. Zhang, Y. Xiang, M. Alrubaian, and M. M. Hassan, ''Detecting spamming activities in twitter based on deep-learning technique,'' *Concurrency Comput., Pract. Exper.*, vol. 29, no. 19, p. e4209, 2017.

[36] M. Alrubaian, M. Al-Qurishi, M. Al-Rakhami, M. M. Hassan, and A. Alamri, ''Reputation-based credibility analysis of Twitter social network users,'' *Concurrency Comput., Pract. Exper.*, vol. 29, no. 7, p. e3873, 2017.

[37] G. Fenza, M. Gallo, and V. Loia, ''Drift-aware methodology for anomaly detection in smart grid,'' *IEEE Access*, vol. 7, pp. 9645–9657, 2019.

[38] Z. Ren, K. Qian, Y. Wang, Z. Zhang, V. Pandit, A. Baird, and B. Schuller, ''Deep scalogram representations for acoustic scene classification,'' *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 3, pp. 662–669, May 2018.

[39] Q. Kang, X. Chen, S. Li, and M. C. Zhou, ''A noise-filtered under-sampling scheme for imbalanced classification,'' *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4263–4274, 2017

[40] Q. Kang, L. Shi, M. C. Zhou, X. Wang, Q. Wu, and Z. Wei, ''A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4152–4165, 2018

XUESONG WANG received the B.S. degree in mechanical engineering and automation from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016, and the M.S. degree in control science and engineering from Tongji University, Shanghai, China, in 2019. He is currently pursuing the Ph.D. degree in computer science and engineering from the University of New South Wales, Australia. His research interests include concept drift detection and streaming data learning.

QI KANG (S'04–M'09–SM'15) received the B.S. degree in automatic control, the M.S. degree in control theory and control engineering, and the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2002, 2005, and 2009, respectively.

From 2007 to 2008, he was a Research Associate with the University of Illinois, Chicago, IL, USA. From 2014 to 2015, he was a Visiting Scholar with the New Jersey Institute of Technology, NJ, USA. He is currently a Professor with the Department of Control Science and Engineering, Tongji University, Shanghai, China. His research interests include computational intelligence, machine learning, intelligent control, and engineering optimization in transportation, energy, and water systems.

JING AN (M'14) received the B.S. degree in automatic control, the M.S. degree in traffic information engineering and control, and the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2002, 2005, and 2013, respectively.

She is currently an Associate Professor with the School of Electrical and Electronic Engineering, Shanghai Institute of Technology. Her research interests include computational intelligence and intelligent information processing.

MENGCHU ZHOU (S'88–M'90–SM'93–F'03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor of electrical and computer engineering. His research interests include Petri nets, sensor networks, web services, big data, semiconductor manufacturing, and transportation and energy systems. He has over 800 publications including 12 books, 600+ journal papers (380+ in IEEE Transactions), and 29 book-chapters. He is a Life Member of the Chinese Association for Science and Technology, USA, and he has served as its President, in 1999. He is a Fellow of the International Federation of Automatic Control (IFAC), the American Association for the Advancement of Science (AAAS), and the Chinese Association of Automation (CAA). He was a recipient of the Perlis Research Award and the Fenster Innovation in Engineering Education Award from NJIT, the Humboldt Research Award for U.S. Senior Scientists, the Leadership Award and Academic Achievement Award from the Chinese Association for Science and Technology, USA, and the Outstanding Contributions Award, the Distinguished Lecturership, the Franklin V. Taylor Memorial Award, and the Norbert Wiener Award from the IEEE SMC Society, and the Distinguished Service Award from the IEEE Robotics and Automation Society. He is currently the Vice-President of Conferences and Meetings of the IEEE Systems, Man, and Cybernetics Society. He is the Founding Editor of the IEEE Press Series on Systems Science and Engineering and the Editor-in-Chief of the IEEE/CAA Journal of Automatica Sinica.